Interactive Oracle Proofs of Proximity to Algebraic Geometry Codes

Sarah Bordage, Mathieu Lhotel, Jade Nardi, Hugues Randriam

Interactive Oracle Proofs of Proximity to Algebraic Geometry Codes

Sarah Bordage* Mathieu Lhotel † Jade Nardi ‡ Hugues Randriam§

February 14, 2022

Abstract

In this work, we initiate the study of proximity testing to Algebraic Geometry (AG) codes. An AG code $C = C(\mathcal{X}, \mathcal{P}, D)$ is a vector space associated to evaluations on $\mathcal{P}$ of functions in the Riemann-Roch space $L_{\mathcal{X}}(D)$. 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 [BBHR18a]. We identify suitable requirements for designing efficient IOPP systems for AG codes. Our approach relies on a neat decomposition of 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. We thus provide a framework in which a proximity test to C can 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. As concrete instantiations, we study AG codes on Kummer curves and curves in the Hermitian tower. The latter can be defined over polylogarithmic-size alphabet. We specialize the generic AG-IOPP construction to reach linear prover running time and logarithmic verification on Kummer curves, and quasilinear prover time with polylogarithmic verification on the Hermitian tower.

*LIX, CNRS UMR 7161, Ecole Polytechnique, Institut Polytechnique de Paris & Inria, Palaiseau, France
sarah.bordage@lix.polytechnique.fr

†Laboratoire de Mathématiques de Besançon, UMR 6623 CNRS Université de Bourgogne Franche-Comté, France
mathieu.lhotel@univ-fcomte.fr

‡Univ Rennes, CNRS, IRMAR - UMR 6625, F-35000 Rennes, France
jade.nardi@univ-rennes1.fr

§ANSSI, Paris, France & Institut Polytechnique de Paris, Télécom Paris, Palaiseau, France
randriam@telecom-paris.fr

Contents

1 Introduction

Let $C \subset \Sigma^S$ be an evaluation code with evaluation domain S of size n and alphabet Σ. For $u \in \Sigma^S$, if $\Delta(u, C) > \delta$, we say that u is δ-far from C and δ-close otherwise. 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. An algebraic geometry (AG) code $C = C(\mathcal{X}, \mathcal{P}, D)$ is a vector space formed by evaluations on a set $\mathcal{P} \subset \mathcal{X}$ of functions in the Riemann-Roch space $L_{\mathcal{X}}(D)$. We address this problem in the Interactive Oracle Proof model [BCS16], which has demonstrated to be particularly promising for the design of proof systems in the past few years.

Context of this work. 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, [BCS16, RRR16]) 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]. In an IOPP for an error-correcting code, a verifier distinguishes between the case where a function is a codeword and the one where it is far from any codeword. The verifier only has an oracle access to the purported codeword and interacts with a prover. We defer the formal definition of an IOPP to Section 1.1.

The FRI protocol is an IOP of Proximity (IOPP) for testing proximity to Reed-Solomon codes evaluated over well-chosen evaluation points ([BBHR18a], further improved in [BKS18], [BGKS20], [BCI+20]). It admits linear prover time, logarithmic verifier time and logarithmic query complexity. While being sub-optimal for some parameters[1], the FRI protocol is highly-efficient in practice and is a crucial tool in systems deployed in the real-world.

The main drawback of RS codes is that they must have an alphabet larger than their length. AG codes [Gop77], as evaluations of a set of functions at some designated rational points on a given curve, extend the notion of Reed-Solomon codes and inherit many of their interesting properties. Therefore, replacing RS codes with AG codes is not only natural but has also led to improvements in the past. Examples of applications of AG codes include public key cryptography,

[1] For instance, [BCG+17, RR20] proposed IOPP constructions for RS codes with constant query complexity.

distributed storage, secret sharing and multi-party computation. 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.

Limitations of Reed-Solomon codes. We identify two limitations of using RS codes in IOPs.

As mentioned earlier, RS codes are the simplest case of AG codes, but possess an inherent limitation: the alphabet size must be larger than the block length of the code. Therefore, practical IOP-based succinct arguments are designed over *large fields*.

The second limitation is related to the algebraic structure of the field. RS-IOPPs [BS08, BBHR18a] require the set $D \subset \mathbb{F}$ of evaluation points to have a special structure. Concretely, the field must contain a subgroup of *large smooth order*, typically a power of 2 which is larger than the size of the non-deterministic computation to be verified. Depending on the applications of succinct non-interactive arguments, a base field might already be imposed. This is for instance the case for standard digital signature schemes. For computations whose size exceeds the order of the largest smooth subgroup of the field, RS-IOPPs known to date can no longer be used.

We observe that lowering the size of field elements does not shorten the length of IOP-based succinct non-interactive arguments (see [BCS16]). There are, however, other reasonable motivations to replace RS codes with AG ones. We explain below how AG codes could circumvent the limitations of RS codes.

Why can AG codes be useful? First, working over smaller fields lowers the cost of field operations[2]. For concrete efficiency, complexity measures such as prover time and verifier time are closely examined. Reducing significantly the size of the alphabet would have a direct impact on the binary cost of arithmetic operations. Smaller fields could enhance efficiency of proof systems since arithmetization of general circuits would be more efficient. Moreover, on the prover side, bit complexity of encoding codewords might be smaller.

A popular belief is that encoding with AG codes is an heavy task. It is surely true in general, but there are explicit families of AG codes for which there are quasilinear time encoding algorithms. We discuss more about encoding in Section 1.3. On another note, putting forward applications of AG codes can motivates the study of fast coding algorithms for AG codes, in particular in the computer algebra community.

One may be concerned by the overhead of reducing the alphabet size when targeting a soundness error less than $2^{-\kappa}$. Notice that it is possible to sample enough bits of randomness from an extension field when needed, or to repeat only some parts of the protocol (see [Sta21, BBHR18b]). For instance, soundness error of our IOPP is bounded from below by $|\mathbb{F}|^{-1}$. Reaching the targeting soundness requires to repeat the interactive phase of the IOPP s times, inducing a factor $s \simeq \frac{\kappa}{\log|\mathbb{F}|}$ multiplicative overhead for the prover. A rough estimation of bit complexities does not show evidence of a significant overhead. Overall, a proof system supporting small fields might be more efficient: any part of the protocol which does not contribute to the soundness error could benefit from cheaper field operations.

[2]Consider the application of checking the correct execution of a size n computation. Then an RS-based IOP for this problem will work over a field of size $\Omega(n)$. This means that a single addition of two field elements will cost $\Omega(\log n)$ operations. If the IOP is instead based on a code with polylogarithmic-size alphabet, the cost of a single addition is only $\Omega(\log(\mathsf{polylog}(n))$.

In addition, AG codes offers more flexibility on the choice of the field. For computations of size n, we propose AG codes for which the field is not required to admit an n-th root of unity (unlike RS-IOPP on a prime field). Specifically, for AG codes over Kummer curves, the base field needs only to have a N-th root of unity, where N divides n. For AG codes over curves in a Hermitian tower (which admits a polylogarithmic-size alphabet), *our IOPP does not involve any assumption on the alphabet*, except that it must be a degree-2 extension of a field $\mathbb{F}_q$, where q is any prime power.

Finally, the question of whether there exist concretely efficient IOPPs for AG codes is motivated by both a theoretical and practical perspective.

1.1 Definition of an IOPP for a code C

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 $\mathsf{r}(n)$ 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 $\delta := \Delta(f, C) > 0$ 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 $\mathsf{l}(n)$ of the IOPP. The query complexity $\mathsf{q}(n)$ 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.

1.2 Our results

In this subsection, we discuss the three contributions of this paper. In all this work, we state complexities in field elements and field operations, where the field is the alphabet of the considered code. Asymptotic complexities are relative to the length of the code.

- The first one is to give a clear criterion for constructing IOPPs for AG codes with linear proof lenght and sublinear query complexity. Our hope with this result is to open up new

possibilities for designing efficient probabilistic proof systems based on constant rate AG codes.

- The second contribution is a concrete instantiation for AG codes defined over Kummer-type curves. This IOPP has strictly linear prover time and strictly logarithmic verification (counted in field operations). Thus, we give a strict generalization of the FRI protocol for codes of length n over alphabet of size roughly $n^{2/3}$.

- The third one is a concrete instantiation for AG codes defined over a tower of Hermitian curves. Considering recursive towers enables to construct an IOPP for AG codes with *polylogarithmic-size* alphabet. For those codes, we give an IOPP with quasilinear prover time and polylogarithmic verification (counted in field operations).

Efficiency of our two AG-IOPP instantiations leverages the fact that proximity testing for these families of AG code can be reduced to a proximity test for a small RS code.

(1) Generic criterion for constructing AG-IOPPs. Let $\mathcal{X}_0$ be a curve defined over a finite field $\mathbb{F}$, D_0 a divisor on the curve $\mathcal{X}_0$ and $\mathcal{P}_0 \subset \mathcal{X}(\mathbb{F})$. This defines an AG code $C_0 = C(\mathcal{X}_0, \mathcal{P}_0, D_0)$. We construct a sequence of curves

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

so that $\mathcal{X}_{i+1}$ arises as the quotient of the curve $\mathcal{X}_i$ by a cyclic group $\mathbb{Z}/p_i\mathbb{Z}$ under the quotient map π_i. Such a sequence of curves exists if and only if a solvable group $\mathcal{G}$ acts on the curve $\mathcal{X}_0$.

Using these consecutive projection maps, we construct a sequence of AG codes $C_i \coloneqq C(\mathcal{X}_i, \mathcal{P}_i, D_i)$ 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. We show that such a procedure is possible if the group $\mathcal{G}$ is large enough with respect to the length of the code C_0 and under some hypotheses on the divisor D_0 overviewed in Section 2.2 and detailed in Section 4. A code fulfilling all the required conditions is called *foldable*.

Assuming that an AG code $C(\mathcal{X}_0, \mathcal{P}_0, D_0)$ of blocklength n is foldable, we show that there is an $O(\log n)$-rounds IOPP for it, with linear proof length, sublinear query complexity and constant soundness (see Theorem 7.9).

In general, we observe that the larger is the group $\mathcal{G}$ acting on $\mathcal{X}_0$ compared to n, the smaller are the query complexity and the verifier decision complexity of the protocol.

However, we notice that the hypothesis on the size of $\mathcal{G}$ is not a necessary condition for constructing an IOPP with sublinear verification. For instance, if the curve $\mathcal{X}_r$ is isomorphic to the projective line $\mathbb{P}^1$, we can continue to recurse in order to reduce even more the size of the proximity testing problem. We propose two interesting families of AG codes for which it is the case.

(2) Concrete IOPP for AG codes on Kummer curves. When $\mathcal{X}$ is a Kummer curve of the form $y^N = f(x)$, we show how to choose $\mathcal{P}$ and D to make the AG code $C = C(\mathcal{X}, \mathcal{P}, D)$ foldable. We benefit from the action of the group $\mathbb{Z}/N\mathbb{Z}$ on $\mathcal{X}$ that yields a quotient curve $\mathcal{X}/(\mathbb{Z}/N\mathbb{Z})$ isomorphic to the *projective line*. This enables us to define a sequence of codes $(C_i)_{0 \le i \le s}$ such that $C_0 = C$ and the code C_s is a *Reed-Solomon code* of dimension $(\deg D)/N + 1$.

Theorem 1.1 (Informal, see Theorem 8.1)**.** *Let $C = C(\mathcal{X}, \mathcal{P}, D) \subset \mathbb{F}^{\mathcal{P}}$ be a foldable AG code defined over a Kummer curve $\mathcal{X}$ of equation $\mathcal{X} : y^N = f(x)$ such that $\deg f = N\ell - 1$ for some integer $\ell > 0$ and N is a smooth integer, coprime with $|\mathbb{F}|$. Assume $\mathbb{F}$ contains a primitive N-th*

root of unity. The block length $n := |\mathcal{P}|$ *is a multiple of* N *and satisfies* $n < \ell N^2 |\mathbb{F}|^{1/2}$*. Let* $f : \mathcal{P} \to \mathbb{F}$ *be a purported codeword. For every proximity parameter* $\delta \in (0,1)$ *and soundness* $\varepsilon \in (0,1)$*, there exists a public-coin IOPP system* (P, V) *for* C *with perfect completeness and the following properties:*

rounds	$\mathsf{r}(n)$	$< \log n$,
proof length	$\mathsf{l}(n)$	$= O(n)$,
query complexity	$\mathsf{q}(n)$	$= O(\log n)$,
prover complexity	$\mathsf{t}_\mathsf{p}(n)$	$= O(n)$,
verifier decision complexity	$\mathsf{t}_\mathsf{v}(n)$	$= O(\log n)$.

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 recall that Hermitian codes over alphabet Σ support block length up to $|\Sigma|^{3/2}$, which is greater by a factor $n^{1/3}$ than Reed-Solomon codes.

(3) Concrete IOPP for AG codes on towers of Hermitian curves. We recall that a tower of curves consists of an infinite sequence of curves

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

such that the number of rational points of the n^{th} curve tends to infinity as n tends to infinity. Towers of curves play a prominent role in the history of AG codes as they define codes with outstanding length and correction capacity [TVZ82, BBGS14].

The Hermitian tower is an example of the widely studied Artin-Scheier extensions [Lac92, Sti08]. In this case, the curve $\mathcal{X}_i$ arises as the quotient of the curve $\mathcal{X}_{i+1}$ above modulo the action of a cyclic group of order q over the finite field $\mathbb{F}_{q^2}$, the first curve $\mathcal{X}_0$ being isomorphic to the projective $\mathbb{P}^1$. Therefore, one can test proximity to an AG-code from one of the curves $\mathcal{X}_n$ by going down along the tower and then testing proximity to a RS code, whose degree can be expressed explicitly in terms of the initial AG code.

Beyond supporting polylogarithmic-size alphabet, AG codes over the Hermitian tower happen to be more naturally "foldable". In particular, no additional assumptions on the alphabet are required.

We write $\mathsf{polylog}(n)$ for functions that are in $O\left(\log^k(n)\right)$ for some k.

Theorem 1.2 (Informal, see Theorem 8.4)**.** *Let* $C = C(\mathcal{X}, \mathcal{P}, D) \subset \mathbb{F}^{\mathcal{P}}$ *be a foldable AG code over an alphabet* $\mathbb{F}$ *of size* $|\mathbb{F}| = \Omega(\log^k(n))$ *for some constant* k*. We denote* $n = |\mathcal{P}|$*. Let* $f : \mathcal{P} \to \mathbb{F}$ *be a purported codeword. For every proximity parameter* $\delta \in (0,1)$ *and soundness* $\varepsilon \in (0,1)$*, there exists a public-coin IOPP system* (P, V) *for* C *with perfect completeness and the following properties:*

rounds	$\mathsf{r}(n)$	$< \log n$,
proof length	$\mathsf{l}(n)$	$= O(n)$,
query complexity	$\mathsf{q}(n)$	$= \mathsf{polylog}(n)$,
prover complexity	$\mathsf{t}_\mathsf{p}(n)$	$= \widetilde{O}(n)$,
verifier decision complexity	$\mathsf{t}_\mathsf{v}(n)$	$= \mathsf{polylog}(n)$.

1.3 More on the practicality of AG codes

When constructing a proximity test for a code, it is assumed that the purported codeword is given as input to the prover. Thus, the prover complexity is computed thereof. While we heavily rely

on the group of automorphisms of the curve for proving the existence of an efficient IOPP for "foldable" AG codes, we emphasize that the work of the prover and the verifier during the protocol is essentially to perform some univariate polynomial interpolation tasks, with very small degree. In particular, neither the prover nor the verifier of the IOPP system need to run an encoding algorithm for AG codes.

However, keeping applications to code-based IOP constructions in mind, the running time of the IOP prover is bounded from below by the time needed to encode codewords during arithmetization. Fast encoding algorithms for AG codes is not the most widely studied computational task, and is often a concern when suggesting constructions based on AG codes[3].

This is a reason why we focus our study on families of AG codes that are particularly likely to lead to practical implementations, as we argue next. Specifically, our study includes the two following subfamilies of one-point AG codes over small alphabets with constant rate and distance.

- The first family includes one-point AG codes over Kummer-type curves, and in particular the notorious Hermitian curve. [BRS20] proposed an encoding algorithm with quasilinear complexity $\widetilde{O}(n)$. Roughly speaking, [BRS20] method consists in translating the encoding task into a bivariate polynomial multipoint-evaluation problem. Assuming that the evaluation points are well-structured, they view a bivariate polynomial in $\mathbb{F}[X,Y]$ as a polynomial in $\mathbb{F}[X][Y]$ in order to evaluate it thanks to two univariate multipoint evaluations. It is the same idea than the one for computing m-dimensional FFT from m (univariate) FFTs.

- The second family of one-point AG codes arises from curves on the Hermitian tower and has an alphabet size polylogarithmic in the block length of the code. It is very likely that those codes could also be encoded in quasilinear time, by iteratively applying the encoding method proposed by [BRS20].

We also point out that bases for Riemann-Roch spaces related to theses codes are explicitly known.

1.4 Related works

We discuss works related to AG-based proximity testing. We emphasize that the motivation behind existing works was only theoretical. In particular, the PCP techniques used are too complex to be implemented for verifying meaningful computations.

In 2013, [BKK+13] constructed 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] reduced 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 pointed 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] proposed an IOP for boolean circuit satisfiability with linear proof length and constant query complexity. The IOP of [BCG+17] invoked the sumcheck

[3] In general, the asymptotic cost of the task of encoding an arbitrary linear code of length n is $O(n^2)$ (using a generator matrix for the code).

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.

A recent work of [RR20] constructed an IOPP for any deterministic language which can be decided in time poly(n) and space $n^{o(1)}$. In particular, [RR20, Corollary 3.6] can be applied to test proximity to AG codes. This IOPP outperforms our construction on some parameters: it has constant round and query complexities, and proof length is slightly less than n. However, it is unlikely that [RR20]'s IOPP leads to a concrete implementation, which is a motivation for our work. Indeed, prover running time is polynomial, and the inner IOPP used for achieving constant query complexity via proof composition is the heavy PCPP of [Mie09]. Mie's PCPP is a theoretical and complex tool used to achieve constant query (e.g in [BCG+17, BCG+19, BCL20]), but it is seen as impractical[4].

By contrast, we exhibit explicit families of AG codes for which we are able to construct a proximity test with linear prover running time and logarithmic verification, or quasilinear prover time with polylogarithmic verification. The main point however, is that our construction is undoubtedly much simpler to implement: the most complex task of the prover and the verifier is simply to perform univariate interpolations (with very small degrees). The technical difficulties are in analyzing the conditions allowing the construction, but the protocol itself is very similar to the FRI protocol. IOPP inspired by the FRI protocol have the inherent barrier of logarithmic query complexity. However, in practice, it is still the most efficient proximity test for RS known to date.

2 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.

2.1 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 $\mathsf{RS}[\mathbb{F}, \mathcal{P}, \rho] \coloneqq \left\{ f \in \mathbb{F}^{\mathcal{P}} \mid \deg f < \rho \cdot |\mathcal{P}| \right\}$ 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.

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

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

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 $\rho n/2$.

(b) *Randomized folding.* Choose $\mathcal{P}$ to be a multiplicative group of order 2^r generated by $\omega \in \mathbb{F}$. Then, define $\mathcal{P}' = \langle \omega^2 \rangle = \{x^2 \mid x \in \mathcal{P}\}$. Set $\pi : \mathbb{F} \to \mathbb{F}$ to be the map defined by $\pi(x) = x^2$,

[4] Proposing an alternative to [Mie09] which does not involve heavy PCP machinery would allow to narrow the gap between the best constructions known in theory and the most efficient ones used in practice.

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* $\mathbf{Fold}[\cdot, z] : \mathbb{F}^{\mathcal{P}} \to \mathbb{F}^{\mathcal{P}'}$ for any $z \in \mathbb{F}$ as follows:

$$\mathbf{Fold}[f, z] \coloneqq g + zh.$$

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 $\mathbf{Fold}[\cdot, z]$ maps $\mathsf{RS}[\mathbb{F}, \mathcal{P}, \rho]$ into $\mathsf{RS}[\mathbb{F}, \mathcal{P}', \rho]$.

(c) *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

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

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

Soundness of the protocol relies notably on Item (c). It is proved using results about distance preservation under random linear combinations, that could be roughly stated as follows: "*Let $V \subset \mathbb{F}_q^n$ be a linear code and $g, h \in \mathbb{F}_q^n$. As long as δ is small enough, if we have $\Delta(g + zh, V) \leq \delta$ for enough values $z \in \mathbb{F}_q$, then both g and h are δ'-close to V, where $\delta' = (1 - o(1))\delta$.*" (see [BBHR18a, BKS18, BGKS20, BCI+20]). Based on that, one can deduce that if $\mathbf{Fold}[f, z] = g + zh$ 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 (c) is to exhibit a codeword which is δ-close from f, based on the decomposition of Item (a).

Remark 2.1. We point out that Item (c) 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 (1) of f 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 $\deg f \leq 2d$.

2.2 Our IOPP for AG proximity testing

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

Group actions and Riemann-Roch spaces. The splitting of the polynomial f into an even and an odd part in Item (a) comes from the action of a multiplicative group of order 2 on 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 is likely to occur in a more general framework.

As soon as a group Γ acts on the curve $\mathcal{X}$, its action naturally extends on the functions on $\mathcal{X}$. Let us denote by π the canonical projection $\pi : \mathcal{X} \to \mathcal{X}/\Gamma$. If we are able to write the Riemann-Roch space associated to D as following

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

for some function μ on the curve $\mathcal{X}$ and some divisors E_j on the quotient curve that are explicitly expressed in terms of the divisor D, then we can mimic the decomposition (1) used in the FRI protocol and get a similiar IOPP.

Now assume that no point of $\mathcal{P}$ is fixed by Γ, *i.e.* for every $P \in \mathcal{P}$ and $j \in \{0, \dots, m-1\}$, $\gamma^j \cdot P \neq P$. 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 ($\star$) can be written for any function in $\mathbb{F}^{\mathcal{P}}$, not only for elements of $L_{\mathcal{X}}(D)$.

Folding operators. From the decomposition ($\star$) above, we want to define a family of folding operators $(\mathbf{Fold}[\cdot, z])_{z \in \mathbb{F}}$ from $\mathbb{F}^{\mathcal{P}}$ to $\mathbb{F}^{\mathcal{P}'}$ and a code $C' = C(\mathcal{X}/\Gamma, \mathcal{P}', D')$ such that $\mathbf{Fold}[\cdot, z](C) \subseteq C'$.

In a first approach, one could choose to define the folding operators similarly to the FRI protocol by setting for $z \in \mathbb{F}$, $\mathbf{Fold}[f, z] = \sum_{j=0}^{p-1} z^j f_j$ where the functions f_j come from the decomposition ($\star$) of $f \in \mathbb{F}^{\mathcal{P}}$. With this definition, the code C' has to be associated to a divisor D' on $\mathcal{X}/\Gamma$ such that each Riemann-Roch space $L_{\mathcal{X}/\Gamma}(E_j)$ can be embedded into $L_{\mathcal{X}/\Gamma}(D')$.

The best scenario is when the divisor D yields a decomposition of $L_{\mathcal{X}}(D)$ 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 ($\star$) of f are the same (or even equivalent) *if $\mathcal{X}$ is not the projective line*. We are then facing an issue analogous to the one described in Remark 2.1 on $\mathbb{P}^1$.

Therefore, such a choice of the folding operators does not guarantee the soundness of our protocol. We thus aim to adapt the idea at the end of Remark 2.1 to the AG setting. We introduce some *balancing* functions ν_j such that, for every $f_j \in C'$, if the product $\nu_j f_j$ also lies in C', then the function f_j belongs to the desired Riemann-Roch space $L_{\mathcal{X}/\Gamma}(E_j)$. 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.

Once we have a divisor D' that 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

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

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

Finding a decomposition ($\star$). Such a decomposition exists for a cyclic group $\Gamma = \langle \gamma \rangle$ whose order is coprime with the characteristic. A result of Kani [Kan86] states that, in this case, there exists a function μ on $\mathcal{X}$ satisfying ($\star$) such that $\gamma \cdot \mu = \zeta\mu$ where ζ is a primitive root of unity of order $|\mathcal{G}|$. Moreover, the divisor $E_{i,j}$ can be explicitly written in terms of the divisor D and the function μ.

Knowing a basis a $L_{\mathcal{X}}(D)$, we may also be able to exhibit such a decomposition without invoking Kani's theorem for some cyclic group of order divisible by the characteristic. This is exactly the strategy we use to design an IOPP for AG codes along the Hermitian tower.

Soundness preservation. The soundness of the protocol depends on the relative minimum distance of the codes C and C'. Ideally , we would like the rates of the codes C and C' to be roughly equal to prevent the relative minimum distance from dropping. In other words, we need $L_{\mathcal{X}/\Gamma}(D')$ to be not too large with respect to the components $L_{\mathcal{X}/\Gamma}(E_j)$.

A natural idea would be to choose D' as the divisor E_j with the largest Riemann-Roch space. However, balancing functions only exists for some well-chosen divisors D', whose degree can be significantly larger than the degree of the divisor E_j in ($\star$). Therefore, the divisors D and D' has to be carefully chosen to prevent the minimum distance from collapsing.

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

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

where the morphism $\pi_i : \mathcal{X}_i \to \mathcal{X}_{i+1}$ 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 $C = C(\mathcal{X}, \mathcal{P}, D)$ is said to be a *foldable AG code* (Definition 4.5) if we are able to construct a sequence of AG codes $C_i \coloneqq C(\mathcal{X}_i, \mathcal{P}_i, D_i)$ that support a family of randomized folding operators $\mathbf{Fold}\left[\cdot, \boldsymbol{z}\right] : \mathbb{F}^{\mathcal{P}_i} \to \mathbb{F}^{\mathcal{P}_{i+1}}$ with the desirable properties for our IOPP (i.e. $\mathbf{Fold}\left[\cdot, \boldsymbol{z}\right](C_i) = (C_{i+1})$, 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 4.

3 Preliminaries

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.

3.1 Functions and divisors on algebraic curves

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

A *divisor* D on $\mathcal{X}$ 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 *support* of D $\mathrm{Supp}(D)$ is the set of points P for which the coefficient n_P is non zero. We will always consider *rational* divisors, whose support only consists in $\mathbb{F}_q$-rational points. We define the *degree* of D equals $\deg D \coloneqq \sum n_P$.

The set of divisors on the curve $\mathcal{X}$ forms an additive group, denoted by $\mathrm{Div}(\mathcal{X})$. 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 $F \coloneqq \mathbb{F}(\mathcal{X})$ of the curve $\mathcal{X}$ defines a divisor

$$\mathrm{div}^F(f) = \sum_{P \in \mathcal{X}} v_P(f) P$$

where $v_P(f)$ is the valuation of the function f at the point P. The index F will be omitted when the context is clear.

We denote by $\mathrm{div}_0(f)$ (respectively $\mathrm{div}_\infty(f)$) the positive (respectively negative) part of the principal divisor $\mathrm{div}(f)$, *i.e.*

$$\mathrm{div}_0^F(f) = \sum_{\substack{P \in \mathcal{X} \\ v_P(f) > 0}} v_P(f) P \text{ and } \mathrm{div}_\infty^F(f) = \sum_{\substack{P \in \mathcal{X} \\ v_P(f) < 0}} v_P(f) P$$

so that $\mathrm{div}_F(f) = \mathrm{div}_0(f) - \mathrm{div}_\infty(f)$. The divisors $\mathrm{div}_0(f)$ and $\mathrm{div}_\infty(f)$ correspond to the loci of zeroes and poles respectively.

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

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

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

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

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

Definition 3.1. *Let $D = \sum n_P P \in \mathrm{Div}(\mathcal{X})$. For any positive integer n, we denote by $\left\lfloor \frac{1}{n} D \right\rfloor \in \mathrm{Div}(\mathcal{X})$ the divisor defined by*

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

3.2 Algebraic geometry 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.

Take $D \in \mathrm{Div}(\mathcal{X})$ and $\mathcal{P} \subset \mathcal{X}(\mathbb{F})$ of size $n \coloneqq |\mathcal{P}|$ such that $\mathrm{Supp}(D) \cap \mathcal{P} = \emptyset$. The *Algebraic Geometry (AG) code* $C = C(\mathcal{X}, \mathcal{P}, D)$ 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 $\mathcal{X} = \mathbb{P}^1$ correspond to Reed-Solomon codes. 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 $\mathcal{X}$, then $\dim L_{\mathcal{X}}(D) = \deg D - g + 1$. 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 $L(D)$.

3.3 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. Such a sequence is called a *normal series*. If $\mathcal{G}$ is solvable, its cardinality equals the product of the sizes of the factor groups.

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

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

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

The action of Γ on $\mathcal{X}$ gives a projection $\pi : \mathcal{X} \to \mathcal{X}/\Gamma$ onto the quotient curve $\mathcal{X}/\Gamma$. A point $Q \in \mathcal{X}/\Gamma$ is called a *ramification point* if 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.

4 Setting of AG codes compatible with proximity test

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 $C(\mathcal{X}, \mathcal{P}, D)$ to testing proximity to a smaller AG code, we introduce a sequence of suitable AG codes of decreasing length.

4.1 Sequence of curves

Fix a curve $\mathcal{X}$ defined over $\mathbb{F}$ and a finite solvable group $\mathcal{G} \subseteq \mathrm{Aut}(\mathcal{X})$. By solvability of $\mathcal{G}$, there exists a *normal series*, i.e. a sequence of subgroups of $\mathcal{G}$

$$\mathcal{G} = \mathcal{G}_0 \rhd \mathcal{G}_1 \rhd \cdots \rhd \mathcal{G}_r = 1, \tag{2}$$

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 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 normal series of $\mathcal{G}$.

A sequence like (2) may not be unique but if the p_i are prime, 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 $\mathcal{X}_0 \coloneqq \mathcal{X}$, as a quotient of $\mathcal{G}$. We thus define the quotient curve $\mathcal{X}_1 \coloneqq \mathcal{X}_0/\Gamma_0$. The group Γ_1 is constant on the orbits under Γ_0. Repeating the process for every $i \in \{0, \dots, r-1\}$ defines a sequence of curves recursively as follows:

$$\mathcal{X}_0 \coloneqq \mathcal{X} \text{ and } \mathcal{X}_{i+1} \coloneqq \mathcal{X}_i/\Gamma_i.$$

We set $F_i \coloneqq \mathbb{F}(\mathcal{X}_i)$ and we denote by $\pi_i : \mathcal{X}_i \to \mathcal{X}_{i+1}$ the canonical projection modulo the action of Γ_i.

$$\begin{array}{ccccccccccccc} & {\scriptstyle \Gamma_0}\circlearrowright & & {\scriptstyle \Gamma_1}\circlearrowright & & & & {\scriptstyle \Gamma_i}\circlearrowright & & {\scriptstyle \Gamma_{i+1}}\circlearrowright & & & \\ \mathcal{X}_0 & \xrightarrow{\pi_0} & \mathcal{X}_1 & \xrightarrow{\pi_1} & \cdots & \xrightarrow{\pi_i} & \mathcal{X}_i & \xrightarrow{\pi_{i+1}} & \mathcal{X}_{i+1} & \longrightarrow & \cdots & \xrightarrow{\pi_{r-1}} & \mathcal{X}_r \end{array} \tag{3}$$

Even if the sequence of curves (3) depends on the composition series (2) of $\mathcal{G}$, the last curve $\mathcal{X}_r$ is always isomorphic to the quotient $\mathcal{X}/\mathcal{G}$.

Definition 4.1. *A sequence of curves constructed as above will be called a* $(\mathcal{X}, \mathcal{G})$-sequence.

4.2 Sequence of codes

Let $(\mathcal{X}_i)$ be a $(\mathcal{X}, \mathcal{G})$-sequence. For any $i \in \{0, \dots, r-1\}$, the factor group Γ_i which acts on the curve $\mathcal{X}_i$ is cyclic of order p_i.

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

4.2.1 Evaluation points

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

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

4.2.2 Divisors

Fix a divisor $D_0 \in \mathrm{Div}(\mathcal{X}_0)$ that is globally Γ_0-invariant. This way, the support of D_0 does not meet the set $\mathcal{P}_0$. For the sake of simplicity, we will assume that D_0 is in fact supported by Γ_0-fixed points.

To make our protocol complete and sound, we need the sequence of divisors (D_i) to have the following properties:

- the divisor D_i is supported by Γ_i-invariant points;
- for each divisor D_i, its associated Riemann-Roch admits a nice decomposition like (4);
- at each step, a divisor D_{i+1} needs to be compatible with D_i and the decomposition of $L_{\mathcal{X}_i}(D_i)$ in the sense of Definition 4.3.

Definition 4.2. *Let $i \in \{0, \dots, r-1\}$. Fix a divisor $D_i \in \mathrm{Div}(\mathcal{X}_i)$ and a function $\mu_i \in F_i$. We say that μ_i* partitions *$L_{\mathcal{X}_i}(D_i)$ (with respect to the action of Γ_i) if*

$$L_{\mathcal{X}_i}(D_i) = \bigoplus_{j=0}^{p_i-1} \mu_i^j \pi_i^* L_{\mathcal{X}_{i+1}}(E_{i,j}) \tag{4}$$

with

$$E_{i,j} \coloneqq \left\lfloor \frac{1}{p_i} \pi_{i*}(D_i + j \,\mathrm{div}_{F_i}(\mu_i)) \right\rfloor \in \mathrm{Div}(\mathcal{X}_{i+1}) \text{ for } j \in \{0, \dots, p_i - 1\}, \tag{5}$$

where the floor function of a divisor is given in Definition 3.1.

Definition 4.3. *Let $i \in \{0, \dots, r-1\}$. Fix a divisor $D_i \in \mathrm{Div}(\mathcal{X}_i)$ and a function $\mu_i \in F_i$ such that μ_i partitions $L_{\mathcal{X}_i}(D_i)$. A divisor $D_{i+1} \in \mathrm{Div}(\mathcal{X}_{i+1})$ is said to be* compatible with (D_i, μ_i) *if both assertions hold.*

1. *for every $j \in \{0, \dots, p_i - 1\}$, $E_{i,j} \leq D_{i+1}$,*
2. *for every $j \in \{0, \dots, p_i - 1\}$, there exists a function $\nu_{i+i,j} \in \mathbb{F}(\mathcal{X}_{i+1})$ such that*

$$\mathrm{div}_\infty(\nu_{i+i,j}) = D_{i+1} - E_{i,j}. \tag{6}$$

The functions $\nu_{i+i,j}$ are called balancing functions.

Remark 4.4. A function μ_i provided by Theorem 4.8 would satisfy (4). However such decomposition may exist without the hypotheses of Theorem 4.8, for example if the characteristic of $\mathbb{F}$ divides $|\mathcal{G}|$.

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 $L(D_{i+1})$.

We have now described all the key components to formally define the notion of foldable codes.

Definition 4.5 (Foldable AG codes)**.** *Let $C = C(\mathcal{X}, \mathcal{P}, D)$ be an AG-code. This code is said to be* foldable *if the following conditions are satisfied.*

- *There exists a finite solvable group $\mathcal{G} \in \mathrm{Aut}(\mathcal{X})$ that acts freely on $\mathcal{P}$: a composition series of $\mathcal{G}$ (2) provides a $(\mathcal{X}, \mathcal{G})$-sequence of curves $(\mathcal{X}_i)$;*
- *There exists $e \in (0, 1)$ such that $|\mathcal{G}| > |\mathcal{P}|^e$;*
- *There exist some sequences $(\mu_i) \in F_i$ and $(D_i) \in \mathrm{Div}(\mathcal{X}_i)$ such that $D_0 = D$ and for every $i \in \{0, \dots, r-1\}$, all the following properties hold:*
  - *the divisors D_i are supported by Γ_i-fixed points,*
  - *the function μ_i partitions $L_{\mathcal{X}_i}(D_i)$ (Definition 4.2),*
  - *D_{i+1} is (D_i, μ_i)-compatible (Definition 4.3).*

Remark 4.6. The second requirement given in Definition 4.3 is definitely compelling and requires some geometric knowledge about the curves $\mathcal{X}_i$. 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 7.8, which takes a prominent role in the design of our IOPP.

To prevent the relative minimum distance of the code C_{i+1} from collapsing and thence ensure a good soundness of the protocol designed in Section 7, one may be tempted to take D_{i+1} as one of the divisors $E_{i,j}$ (5) that appear in the decomposition (4) of $L_{\mathcal{X}_i}(D)$. However the Weierstrass gaps theory indicates that balancing functions exist only when choosing a (D_i, μ_i)-compatible divisor D_{i+1} whose degree may be unexpectedly substantial (see Example 5.7). Therefore, to broaden the spectrum of foldable codes, we do not make this additional hypothesis.

4.3 RS codes are foldable AG codes

RS codes are AG codes on the projective line $\mathbb{P}^1$. Moreover the Riemann-Roch space $L_{\mathbb{P}^1}(dP_\infty)$ on $\mathbb{P}^1$ for $P_\infty = [0:1]$ is isomorphic to the set of polynomials of degree (less than or equal to) d.

In this case, the decomposition (4) is nothing but the splitting of a polynomial into an even part and an odd part, which plays a crucial role in the FRI protocol.

To make both points of views coincide, let us consider the involution $\gamma : [X_0 : X_1] \mapsto [-X_0 : X_1]$. 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. Let us choose μ as the function $x = \frac{X_0}{X_1}$. We have $\mathrm{div}(x) = P_0 - P_\infty$ with $P_0 = [1:0]$. 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 even and odd parts of respective degrees $\lfloor \frac{d}{2} \rfloor$ and $\lfloor \frac{d-1}{2} \rfloor$.

Remark 4.7. The function μ is not unique: any odd polynomial of x would make a suitable choice for μ.

Now, let us remark that 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, for any $\mathcal{P} \subset \mathbb{F}$ of size $|\mathcal{P}| = 2^r$ for a certain integer r and any degree bound d. We shall then retrieve the construction of the RS proximity test of [BBHR18a].

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 4.5. When considering its composition series

$$\mathbb{Z}/2^r\mathbb{Z} \rhd \mathbb{Z}/2^{r-1}\mathbb{Z} \rhd \cdots \rhd 1 \tag{7}$$

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 $(\mathcal{X}_i)$ with $\mathcal{X}_i = \mathbb{P}^1$. 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 $i \in \{0, \ldots, r-1\}$, $d_{i+1} \coloneqq \left\lfloor \frac{d_i}{2} \right\rfloor$. Note that there exists $r' < r$ such that $d_{r'}, \ldots, d_r$ are all equal to 0. Setting $D_i = \left\lfloor \frac{d_i}{2} \right\rfloor P_\infty$, we have Γ_i-invariant divisors fulfilling the compatibility condition given in Definition 4.3, 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.

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

The first requirement to make a sequence of codes foldable is the splitting the Riemann-Roch spaces as in (4), which mimics the decomposition in odd and even parts of univariate polynomials. Under some additional hypothesis, a decomposition like (4) always exists. Let us detail this framework.

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 $\zeta \in \overline{\mathbb{F}}$ a primitive m^{th} root of unity, lying in some algebraic closure $\overline{\mathbb{F}}$ of $\mathbb{F}$.

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

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

- *There exists a function $\mu \in \overline{\mathbb{F}}(\mathcal{X})$ such that $\gamma \cdot \mu = \zeta\mu$.*
- *For any Γ-invariant divisor $D \in \mathrm{Div}(\mathcal{X})$, when considering the Riemman-Roch spaces over the algeraic closure $\overline{\mathbb{F}}$, we have*

$$L_{\mathcal{X}}(D)_j \otimes \overline{\mathbb{F}} \simeq \mu^j \pi^* \left(L_{\mathcal{Y}} \left(\left\lfloor \frac{1}{m} \pi_* \left(D + j \,\mathrm{div}(\mu) \right) \right\rfloor \right) \otimes \overline{\mathbb{F}} \right). \tag{8}$$

Remark 4.9. If the function μ is defined over the base field $\mathbb{F}$ then the decomposition (8) is valid when considering $\mathbb{F}$-vector spaces:

$$L_{\mathcal{X}}(D)_j \simeq \mu^j \pi^* \left(L_{\mathcal{Y}} \left(\left\lfloor \frac{1}{m} \pi_* \left(D + j \,\mathrm{div}(\mu) \right) \right\rfloor \right) \right).$$

In practical instantiations, we are always able to choose μ defined over $\mathbb{F}$, even when ζ does not belong to $\mathbb{F}$. Such decomposition thus holds even if there is no m^{th} primitive root in our base field $\mathbb{F}$. However, as the evaluation set $\mathcal{P}$ is formed of orbits of size $\mathcal{G}$, such instantiations may require this primitive root to belong to $\mathbb{F}$, as it is the case for Kummer curves (see Section 5).

To handle the divisors that appears in the decomposition above, we need to get a better grasp on the zeroes and the poles of the function μ, including the ramification points of π according to the following lemma.

Lemma 4.10. *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 $P \in \mathrm{Supp}(\mu)$.*

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

$$\begin{aligned} (\gamma^j \cdot \mu)(P) &= \zeta^j \mu(P) && \text{by definition of } \mu \text{ in Th. 4.8,} \\ &= \mu(P) && \text{because } \gamma^j \in \Gamma_P. \end{aligned}$$

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

5 Foldable AG codes on Kummer curves

5.1 Preliminaries

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

$$\mathcal{X} : y^N = f(x) = \prod_{\ell=1}^{m}(x - \alpha_\ell) \tag{9}$$

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 $\mathcal{X}$ 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 $\mathcal{X}$ via the morphism $(x, y) \mapsto (x, \zeta y)$ where ζ is a primitive N^{th} root of unity. We assume that ζ belongs to $\mathbb{F}$.

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

$$\mathbb{Z}/N\mathbb{Z} \rhd \mathbb{Z}/N_1\mathbb{Z} \rhd \mathbb{Z}/N_2\mathbb{Z} \rhd \cdots \rhd \mathbb{Z}/N_{r-1}\mathbb{Z} \rhd 1, \tag{10}$$

where

$$N_i \coloneqq \prod_{j=i}^{r-1} p_j. \tag{11}$$

The i-th factor group Γ_i is isomorphic to the cyclic group of prime order $\mathbb{Z}/p_i\mathbb{Z}$. 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 $\mathcal{X}_0 \coloneqq \mathcal{X}$. By Section 4.1, the composition series (10) gives a sequence of curves $(\mathcal{X}_i)$ in which the i^{th} curve is defined by

$$\mathcal{X}_i : y^{N_i} = f(x) \tag{12}$$

and has genus

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

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

$$\overset{\gamma_0}{\circlearrowright}\ \mathcal{X}_0 \xrightarrow{\pi_0} \cdots \xrightarrow{\pi_i} \overset{\gamma_i}{\circlearrowright}\ \mathcal{X}_i \xrightarrow{\pi_{i+1}} \overset{\gamma_{i+1}}{\circlearrowright}\ \mathcal{X}_{i+1} \longrightarrow \cdots \xrightarrow{\pi_{r-1}} \mathcal{X}_r \simeq \mathbb{P}^1.$$

Example 5.1. The Hermitian curve defined over $\mathbb{F}_{q^2}$ by

$$\mathcal{X}_0 : y^{q+1} = x^q + x. \tag{13}$$

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

Stabilized points. Let us denote P_∞^i the unique point at infinity on the curve $\mathcal{X}_i$. One can easily check that

$$P_\infty^i := \begin{cases} (1:0:0) & \text{if } N > m \\ (0:1:0) & \text{otherwise.} \end{cases}$$

The points of $\mathcal{X}_0$ 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 $P_1, \ldots, P_m$ and P_∞^i.

5.2 Decomposition of Riemman-Roch spaces

The theory of Kummer extensions provides us a decomposition like (4) at each level, with $\mu_i = y$ for every $i \in \{0, \dots, r-1\}$.

Theorem 5.2. *[Mah04, Theorem 2.2] Let $D \in \mathrm{Div}(\mathcal{X}_i)$ that is Γ_i-invariant. Then*

$$L_{\mathcal{X}_i}(D_i) = \bigoplus_{j=0}^{p_i-1} L_{\mathcal{X}_{i+1}}\left(\left\lfloor \frac{1}{p_i}(\pi_i)_*(D_i + j\,\mathrm{div}_{F_i}(y))\right\rfloor\right).$$

An example of a sequence of y-compatible divisors. In order to exhibit a sequence of divisors (D_i) such that D_{i+1} is (D_i, y)-compatible for every $i \geq 0$, we need to handle the divisor associated to y and some other elementary functions on each curve $\mathcal{X}_i$, described for instance in [MQS15].

Lemma 5.3 ([MQS15])**.** *On $\mathcal{X}_i$ for every $i \in \{0, \dots, r-1\}$, we have*

1. $\mathrm{div}_{F_i}(x - \alpha_\ell) = N_i(P_\ell - P_\infty^i)$,
2. $\mathrm{div}_{F_i}(y) = P_1 + \cdots + P_m - mP_\infty^i$.

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

Lemma 5.4. *Set $D_0 = \sum_{\ell=1}^{m} a_{0,\ell}P_\ell + b_0 P_\infty^0 \in \mathrm{Div}(\mathcal{X}_0)$.*

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

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

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

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

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 $a_{0,1}, \dots, a_{0,m}, b_0$ 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 $\alpha \in \{\alpha_1, \dots, \alpha_m\}$ gives the last condition on D_{i+1} for it to be (D_i, y)-compatible by Definition 4.3, i.e. $D_{i+1} - E_{i,j} = \mathrm{div}_\infty(\nu_{i+1,j})$. □

5.3 Family of foldable codes

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

Proposition 5.5. *Let $\mathcal{X}_0$ be a Kummer curve defined by (9) with $m \equiv -1 \mod N$. Take an evaluation set $\mathcal{P}_0 \subseteq \mathcal{X}_0(\mathbb{F}) \backslash \{P_1, \dots, P_m, P_\infty^0\}$ formed by $\mathbb{Z}/N\mathbb{Z}$-orbits. Take $D_0 \in \mathrm{Div}(\mathcal{X}_0)$ satisfying hypothesis of Lemma 5.4. If $N > n^e$ for some $e \in (0,1)$, then the AG code $C = C(\mathcal{X}_0, \mathcal{P}_0, D_0)$ is foldable.*

The length of foldable codes over a Kummer curve as defined in (9) over $\mathbb{F}_q$ is bounded from above by $q + 1 + (N-1)(\kappa N - 2)\sqrt{q} - \kappa N$, using Hasse-Weil bound, write $m = \kappa N - 1$.

Remark 5.6.

1. The primitive N^{th} root ζ needs to belong to the base field $\mathbb{F}$ to ensure that the set $\mathcal{P}_0$ is not empty.
2. The condition on the coefficients of D_0 can be loosen while the previous statement still holds. If $a_{0,1}, \dots, a_{0,m}, b_0$ 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 last curve $\mathcal{X}_r$ being isomorphic to $\mathbb{P}^1$, the existence of balacing functions is trivial, if the first requirement of Definition 4.3 holds.

What happens outside these hypotheses? Lemma 5.4 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 4.5. Let us have a look at what could happen when dropping these conditions.

Example 5.7. 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 $D_0 = 18P_\infty^0$, 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 4.3 to be (D_0, y)-compatible but not the third one. One can reasonably ask the support of D_1 to consist only in $\pi_0(P_\infty^0) = P_\infty^1$, 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 $\mathcal{X}_1 : y^3 = x^5 + x$ 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 $D_1 = d_1 P_\infty^1$ that is (D_0, y)-compatible is $D_1 = 12P_\infty^1$. 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.

5.3.1 Explicit basis of the Riemman-Roch spaces

AG codes from the Kummer curve $\mathcal{X}$ associated to divisors as defined in Lemma 5.4 have been studied by Hu and Yang [HY18]. They provide a basis of the Riemman-Roch spaces in a combinatorial form.

Theorem 5.8. *[HY18, Theorem 5] Let j, $j_2, \dots, j_m$ be integers. We define*

$$E_{j,\, j_2, \dots, j_m} := y^j \prod_{\ell=2}^{m} (x - \alpha_\ell)^{j_\ell}.$$

Consider $D = \sum_{\ell=1}^{m} a_\ell P_\ell + bP_\infty$. *Set*

$$\Omega_{a_1,\dots,a_m,b} \coloneqq \Big\{ (j,\, j_2, \dots, j_m) \mid j + a_1 \geq 0,\; j_\ell = \left\lceil \frac{-j - a_\ell}{N} \right\rceil \text{ for } \ell = 2, \dots, m \text{ and } mj + N(j_2 + \cdots + j_m) \leq b \Big\}.$$

Then the elements $E_{j,\, j_2,\dots,j_m}$ *for* $(j,\, j_2, \dots, j_m) \in \Omega_{a_1,\dots,a_m,b}$ *form a basis of* $L_{\mathcal{X}}(D)$.

5.3.2 Parameters

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

Lemma 5.9. *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 $i \in \{0, \dots, r-1\}$,

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

□

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 5.10. *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 $0 \leq \deg(D_r) < n_r$ by Lemma 5.9 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 $\Delta(C_i) \leq \Delta(C_{i+1})$. On the other hand, as $\deg(C_i) < n_i$, we have $\Delta(C_i) \geq 1 - \frac{\deg D_i}{n_i}$, which concludes the proof. □

6 Foldable AG codes along the Hermitian tower

6.1 Preliminaries

Sequence of curves. We consider the sequence of function fields $\mathcal{F} = (F_i)_{i \geq 0}$ over $\mathbb{F}_{q^2}$ that is defined recursively by $F_0 = \mathbb{F}_{q^2}(x_0)$ and $F_i = F_{i-1}(x_i)$ with equations

$$x_i^q + x_i = x_{i-1}^{q+1} \text{ for } i \geq 1. \tag{14}$$

Note that this tower of function field $\mathcal{F}$ corresponds to a tower of curves $(\mathcal{X}_i)_{i\geq 0}$ such that $F_i = \mathbb{F}_{q^2}(\mathcal{X}_i)$. One can view the curve $\mathcal{X}_i$ embedded in an i-dimensional affine space with variables $(x_1, \ldots, x_i)$ defined by the equations (14).

For $i = 1$, the field F_1 is the function field of the Hermitian curve $\mathcal{H} := \mathcal{X}_1$ over $\mathbb{F}_{q^2}$.

Let $g_i := g(\mathcal{X}_i)$ denote the genus of the curve $\mathcal{X}_i$. An explicit formula was given by Pellikaan, Shen and Wee [PzSJ91, Proposition 4]. We have $g_0 = 0$ and for $i \geq 1$,

$$g_i = \frac{1}{2}\left[(q^2-1)\left((q+1)^i - q^i\right) + 1 - q^i\right] = \frac{1}{2}\cdot\left(\sum_{k=1}^{i} q^{i+1}\cdot\left(1+\frac{1}{q}\right)^{k-1} + 1 - (1+q)^i\right). \tag{15}$$

For every $i \geq 0$, the number of $\mathbb{F}_{q^2}$–rational places in F_i is given by

$$\left|\mathcal{X}_i(\mathbb{F}_{q^2})\right| = q^{i+2} + 1.$$

We have an infinite sequence of curves $(\mathcal{X}_i)_{i\geq 0}$ as follows.

$$\ldots \xrightarrow{\pi_{i+1}} \mathcal{X}_i \xrightarrow{\pi_i} \mathcal{X}_{i-1} \xrightarrow{\pi_{i-1}} \ldots \xrightarrow{\pi_1} \mathcal{X}_0 \simeq \mathbb{P}^1.$$

(γ_i acts on $\mathcal{X}_i$, γ_{i-1} acts on $\mathcal{X}_{i-1}$)

Remark 6.1. In the context of recursive towers, it is classical to index the curves the other way round compared to the notations used in Section 4.

This tower is a tower of Artin-Schreier extensions, which have been extensively studied (see for example [Sti08]). Let us recall some classical results that will be useful to design foldable AG codes along this tower.

Automorphisms and projection maps. Take $\alpha \in \mathbb{F}_{q^2}$ such that $\alpha^q + \alpha = 0$. For every $i \geq 0$, the automorphism γ_i that fixes $x_1, \ldots, x_{i-1}$ and sends x_i to $x_i + \alpha$ has order q. Then $\langle\gamma_i\rangle = \mathbb{Z}/q\mathbb{Z}$. The quotient map $\pi_i : \mathcal{X}_i \to \mathcal{X}_{i-1}$ consists in the projection onto the first i coordinates.

For every $i \geq 0$, we set Π_i to be the composition of the first i quotient maps, *i.e.*

$$\Pi_i := \pi_i \circ \pi_{i-1} \circ \cdots \circ \pi_0.$$

Behaviour of the point of infinity. In what follows, let us denote by $P_\infty^{(0)}$ the unique pole of the function x_0 in F_0, which corresponds to the point at infinity on the projective line $\mathcal{X}_0 = \mathbb{P}^1$.

Lemma 6.2. *[Sti08, Proposition 3.7.8] Let $i \geq 1$. The place $P_\infty^{(0)}$ is totally ramified in F_i, which means that the preimage $\Pi_i^{-1}\left(\left\{P_\infty^{(0)}\right\}\right)$ consists in a unique place, denoted by $P_\infty^{(i)} \in \mathcal{X}_i$. Moreover, $P_\infty^{(0)}$ is the unique place that is ramified in the tower $\mathcal{F}$.*

The peculiar behaviour of the points $P_\infty^{(i)}$ in the tower encourages us to define a sequence of codes associated with divisors $D_i \in \mathrm{Div}(\mathcal{X}_i)$ of the form

$$D_i := d_i P_\infty^{(i)} \text{ for } i \geq 1.$$

Let us focus on the principal divisors $\mathrm{div}^{F_i}(x_j)$ $(0 \leq j \leq i)$ and their valuation at the point $P_\infty^{(i)}$.

Their properties follow from the study of the basic function field $F = \mathbb{F}_{q^2}(x, y)$ which is nothing but the Hermitian function field. It is a special case of Artin-Schreier extension of $\mathbb{F}_{q^2}(x)$ and is well-known that we have

$$\mathrm{div}^F(y) = P^{(0)} - P_\infty^0.$$

Remark 6.3. The role of the variables x and y is reversed compared to the Kummer model of the Hermitian curve, studied in the previous section.

Since each extension F_i/F_{i-1} corresponds to the same Artin-Schreier extension, and that $P_\infty^{(0)}$ is fully ramified in F_i/F_0, we can deduce the form of the divisor $\mathrm{div}_{F_i}(x_i)$, given in the next lemma. The valuation of the function $x_j \in F_{i-1}$ at $P_\infty^{(j)}$ follows from the extension degrees $[F_{i-1} : F_j] = q^{i-1-j}$.

Lemma 6.4. *The following two assertions hold.*

1. *For $i \geq 1$, we have*
$$\mathrm{div}^{F_i}(x_i) = (q+1)^i \left(P^{(i)} - P_\infty^{(i)} \right),$$
where $P^{(i)}$ is the unique common zero of the functions $x_0, ..., x_i$;

2. *Let $i \geq 1$. Then for $0 \leq j \leq i-1$, the valuation of the function $x_j \in F_{i-1}$ is given by*
$$v_{P_\infty^{(i-1)}}(x_j) = -q^{i-1-j}(q+1)^j.$$

Basis of the Riemann-Roch spaces associated to the divisor $d_i P_\infty^{(i)}$. For a given $i \geq 0$, $P_\infty^{(i)}$ is the unique pole of the functions $x_0, ..., x_i$, which gives an explicit basis of the Riemman-Roch space associated to a multiple of $P_\infty^{(i)}$.

Lemma 6.5. *For all $i \leq 1$ and $m \leq 1$, the Riemann-Roch $L_{F_i}(mP_\infty^{(i)})$ is formed by linear combinations of functions in the following set:*
$$\left\{ x_0^{a_0} \cdots x_i^{a_i} \mid 0 \leq a_0 \ , \ 0 \leq a_j \leq q-1 \text{ and } \sum_{j=0}^{i} a_j q^{i-j}(q+1)^j \leq m \right\}.$$

6.2 Construction of foldable AG codes

Let us fix $i \geq 0$. We aim to define a sequence of AG codes on the tower of curves $(\mathcal{X}_i)_{i \geq 0}$ defined by
$$C(\mathcal{X}_i, \mathcal{P}_i, D_i) \text{ where } \mathcal{P}_i \subseteq \mathcal{X}_i(\mathbb{F}_{q^2}) \setminus \{P_\infty^{(i)}\} \text{ and } D_i = d_i P_\infty^{(i)}.$$

In order to obtain a sequence of foldable codes, we need to describe the Riemann-Roch spaces on a certain step from Riemann-Roch spaces on lower curves. A priori Kani's theorem does not apply, so we will have to find a decomposition by hand. We deduce such a decomposition from the explicit basis of the Riemman-Roch space given in Lemma 6.5.

Proposition 6.6. *Let $i \geq 0$. Set $D_i = d_i P_\infty^{(i)}$ for some integer d_i. Then*
$$L_{\mathcal{X}_i}(D_i) = \bigoplus_{j=0}^{q-1} x_i^j \pi_i^* \left(L_{F_{i-1}}(E_{i,j}) \right)$$
with
$$E_{i,j} \coloneqq \left\lfloor \frac{1}{q} \pi_{i*} \left(D_i - j \cdot \mathrm{div}^{F_i}(x_i) \right) \right\rfloor \text{ for } 0 \leq j \leq q-1.$$
In other words, the function $x_i \in F_i$ partitions the divisor D_i in the sense of Definition 4.2.

Proof. By Lemma 6.5, $L_{F_i}(D_i)$ is formed by linear combinations of $x_0^{a_0} \cdots x_i^{a_i}$ with non negative exponents such that $0 \leq a_j \leq q-1$ for $j \neq 1$ and $\sum_{j=0}^{i} a_j q^{i-j}(q+1)^j \leq m$. As a_j runs in $\{0, \ldots, q-1\}$, the proof is concluded by noticing that the function $x_0^{a_0} \cdots x_{i-1}^{a_{i-1}} \in F_i$ lies in $L(D_i - j \cdot \mathrm{div}^{F_i}(x_i))$ which means that $x_0^{a_0} \cdots x_{i-1}^{a_{i-1}} \in F_{i-1}$ belongs to $L(E_{i,j})$. □

To make D_{i-1} compatible with (D_i, x_i) (Definition 4.3), we need the existence of q balancing functions $\nu_{i-1,j} \in F_{i-1}$ (for every $0 \leq j \leq q-1$) such that

$$D_{i-1} - E_{i,j} = (\nu_{i-1,j})_\infty. \tag{16}$$

In our setup, we have

$$E_{i,j} = \left\lfloor \frac{d_i - j(q+1)^i}{q} \right\rfloor P_\infty^{(i-1)}.$$

Thus, we need to "balance" the divisors

$$D_{i-1} - E_{i,j} = \left(d_{i-1} - \left\lfloor \frac{d_i - j(q+1)^i}{q} \right\rfloor \right) P_\infty^{(i-1)}.$$

We are led to study the Weierstrass semigroup of $P_\infty^{(i-1)}$, denoted by $\mathcal{H}\left(P_\infty^{(i-1)}\right)$. The generators of this semigroup can be found using Lemma 6.4. In fact, $P_\infty^{(i-1)}$ is the unique commun pole of the functions $x_0, ..., x_{i-1} \in F_{i-1}$ and we know their exact valuation. Thus we have

$$\mathcal{H}\left(P_\infty^{(i-1)}\right) = \left\langle q^{i-1-k}(q+1)^k \ , \ 0 \leq k \leq i-1 \right\rangle_{\mathbb{N}}.$$

Remark 6.7. In the spirit of the FRI protocol, one could be tempted to choose D_{i-1} as $E_{i,0}$. Such a choice would be valid in the sense of Definition 4.3 if and only for every $0 \leq j \leq q-1$

$$\left\lfloor \frac{d_i}{q} \right\rfloor - \left\lfloor \frac{d_i - j(q+1)^i}{q} \right\rfloor \in \mathcal{H}\left(P_\infty^{(i-1)}\right).$$

Unfortunately, when i increases, this condition is never satisfied.

To ensure that $\deg(D_{i-1} - E_{i,j})$ is never a Weierstrass gap for $P_\infty^{(i-1)}$, we increase the degree d_{i-1} of D_{i-1}.

Theorem 6.8. *Let $i \geq 1$. Set $D_i = d_i P_\infty^{(i)}$ for some integer d_i and $D_{i-1} = d_{i-1} P_\infty^{(i-1)}$ where*

$$d_{i-1} \coloneqq \left\lfloor \frac{d_i}{q} \right\rfloor + 2g_{i-1}. \tag{17}$$

Then D_{i-1} is compatible with (D_i, x_i) (Definition 4.3).

Proof. By [Sti08, Theorem 1.6.8], we know that

$$\max\left(\mathbb{N} \setminus \mathcal{H}\left(P_\infty^{(i-1)}\right)\right) \leq 2g_{i-1} - 1.$$

Then for every $0 \leq j \leq q-1$, the difference

$$m_{i,j} := \deg(D_{i-1} - E_{i,j}) = \left(\left\lfloor \frac{d_i}{q} \right\rfloor - \left\lfloor \frac{d_i - j(q+1)^i}{q} \right\rfloor + 2g_{i-1} \right) \tag{18}$$

always belongs to the Weierstrass semigroup $\mathcal{H}\left(P_\infty^{(i-1)}\right)$. □

About the balancing functions. Since we know a $\mathbb{N}$-basis of the Weierstrass semigroup at $P_\infty^{(i-1)}$, we are able to explicit the form of the functions $\nu_{i-1,j}$. In particular, they can be chosen as product of powers of the functions $x_0, ..., x_{i-1}$. More precisely, if $a_{i,j} := (a_{i,j}(0), ..., a_{i,j}(i-1)) \in \mathbb{N}^i$ are integers such that

$$m_{i,j} = \sum_{k=0}^{i-1} a_{i,j}(k) \cdot q^{i-1-k}(q+1)^k, \tag{19}$$

then $m_{i,j} \in \mathcal{H}\left(P_\infty^{(i-1)}\right)$. The corresponding choice for the balancing function is then given by

$$\nu_{i-1,j} = \prod_{k=0}^{i-1} x_k^{a_{i,j}(k)}.$$

Note that finding a vector $a_{i,j} \in \mathbb{N}^i$ satisfying (19) leads to the study of the diophantine equation

$$m_{i,j} = \sum_{k=0}^{i-1} a_k \cdot q^{i-1-k}(q+1)^k$$

with i unknowns $a_k \in \mathbb{N}$, for which we know there exists at least a solution (and we only need one).

6.2.1 A family of foldable codes

We denote by $i_{\max}$ the level in the tower $(\mathcal{X}_i)_{i\geq 0}$, such that $\mathcal{X}_{i_{\max}}$ is the curve on which the code we want to test proximity is defined.

Proposition 6.9. *Fix an integer $i_{\max}$. Set $\mathcal{P}_0 \subseteq \mathbb{P}^1(\mathbb{F}_{q^2}) \setminus \{P_\infty^0\}$ and define $\mathcal{P}_{i_{\max}}$ as the preimage of $\mathcal{P}_0$ under the morphism $\Pi_{i_{\max}}$. Fix an integer $d_{i_{\max}}$. Then the code $C(\mathcal{X}_{i_{\max}}, \mathcal{P}_{i_{\max}}, d_{i_{\max}} P_\infty^{(i)})$ is foldable.*

Proof. The group $\mathcal{G} \simeq \mathbb{Z}/q^{i_{\max}}\mathbb{Z}$ acts on $\mathcal{X}_{i_{\max}}$ and its action on $\mathcal{P}_{i_{\max}}$ is free, by definition of $\mathcal{P}_{i_{\max}}$.
The cardinality of $\mathcal{P}_{i_{\max}}$ is equal to $|\mathcal{P}_0|\, q^{i_{\max}}$, hence $|\mathcal{G}| > |\mathcal{P}_{i_{\max}}|^e$ for some $e \in (0,1)$.
The third condition of Definition 4.5 follows from Theorem 6.8. □

To control the dimension of foldable codes, we will focus on those of the form

$$C := C\left(\mathcal{X}_{i_{\max}}, \mathcal{X}_{i_{\max}}(\mathbb{F}_{q^2}) \setminus \{P_\infty^{(i_{\max})}, (2\alpha+1)g_{i_{\max}})P_\infty^{(i_{\max})}\right) \tag{20}$$

for some $\alpha > 1/2$. In this case, we have $n_{i_{\max}} = q^{i_{\max}+2}$. We can determine a sufficient condition over $i_{\max}$ and α to get constant rate.

Lemma 6.10. *Let $R \in (0,1)$. Fix $\varepsilon \in (0,1)$. Set $i_{\max} := q^\varepsilon$ and $\alpha := Rq^{1-\varepsilon}$.*

The ratio of the dimension of the code C defined in (20) by its block length goes to R when q tends to infinity.

If $2(q^\varepsilon - 1) < q$, the relative minimum distance of C is bounded from below by $1 - R\left(1 + \frac{1}{q}\right)$.

Proof. If $\alpha > \frac{1}{2}$, the dimension of the code is equal to $(2\alpha+1)g_{i_{\max}} - g_{i_{\max}} + 1 = 2Rq^{1-\varepsilon}g_{i_{\max}} + 1$ by the Riemann-Roch Theorem. As R is fixed and q goes to infinity, we can assume that $\alpha > 1/2$ to compute the rate as

$$\lim_{q\to\infty} \frac{2Rq^{1-\varepsilon}g_{i_{\max}}}{q^{i_{\max}+2}}$$

for $i_{\max} = q^{\varepsilon}$. Lemma B.2 in Appendix B clearly implies that this limit is equal to R.

Regarding the relative minimum distance, we use the Goppa bound: if $(2\alpha+1)g_{i_{\max}} < q^{i_{\max}+2}$, then the relative minimum distance of C satisfies $\Delta(C) \geq 1 - \frac{(2\alpha+1)g_{i_{\max}}}{q^{i_{\max}+2}}$. By Proposition B.1 in Appendix B, we have

$$\frac{g_{i_{\max}}}{q^{i_{\max}+2}} \leq \frac{i_{\max}}{2q}\left(1 + \frac{i_{\max}}{q}\right),$$

which gives the expected lowerbound for our choice of α and $i_{\max}$. □

7 Folding operators for 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.

7.1 Definition of folding operators

Let $C_0 = C(\mathcal{X}_0, \mathcal{P}_0, D_0)$ be a code satisfying Definition 4.5. We consider its associated $(\mathcal{X}, \mathcal{G})$-sequence of curves $(\mathcal{X}_i)$ and its sequence of divisors (D_i).

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 $C_i = C(\mathcal{X}_i, \mathcal{P}_i, D_i)$. 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 7.1 (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

$$I_{f,P}(X) \coloneqq \sum_{j=0}^{p_i-1} X^j a_{j,P} \tag{21}$$

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

$$f_j : \left\{ \begin{array}{ccc} \mathcal{P}_{i+1} & \to & \mathbb{F}, \\ P & \mapsto & a_{j,P}. \end{array} \right. \tag{22}$$

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 7.4 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 4.3 in the second term of the sum in (23).

Definition 7.2 (Folding operator)**.** *For any $\boldsymbol{z} = (z_1, z_2) \in \mathbb{F}^2$, we define the* folding of f *to be the function* $\mathbf{Fold}[f, \boldsymbol{z}] : \mathcal{P}_{i+1} \to \mathbb{F}$ *such that*

$$\mathbf{Fold}[f, \boldsymbol{z}] \coloneqq \sum_{j=0}^{p_i-1} z_1^j f_j + \sum_{j=0}^{p_i-1} z_2^{j+1} \nu_{i+1,j} f_j \tag{23}$$

where the functions f_j are defined in Equation (22) and the functions $\nu_{i+j,j}$ in Definition 4.3.

7.2 Properties of folding operators

Our aim is to prove that the folding operators satisfy three key properties: local computability, completeness, and distance preservation. This will enable us to invoke [ABN21, Theorem 1] for the completeness and soundness of our AG-IOPP.

Given the p_i points $((\mu_i(\widehat{P}), f(\widehat{P})))_{\widehat{P} \in S_P}$, one can determine the coefficients $(a_{j,P})_{0 \le j < p}$ of $I_{f,P}$ defined in (21) 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 fast prover time and verifier decision complexity.

Lemma 7.3 (Locality)**.** *Let $\boldsymbol{z} \in \mathbb{F}^2$. For each $P \in \mathcal{P}_{i+1}$, the value of $\mathbf{Fold}[f, \boldsymbol{z}](P)$ can be computed with exactly p_i queries to f, namely at the points $\pi_i^{-1}(\{P\})$.*

Proposition 7.4 (Completeness)**.** *Let $\boldsymbol{z} \in \mathbb{F}^2$. If $f \in C_i$, then $\mathbf{Fold}[f, \boldsymbol{z}] \in C_{i+1}$.*

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 4.8, 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 $\widehat{P} \in S_P$,

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

Moreover, for all $P \in \mathcal{P}_{i+1}$, polynomials $I_{f,P}(X), \mathbf{Fold}[f, (X, 0)](P) \in \mathbb{F}[X]$ 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,

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

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 (22) coincides with $\widetilde{f}_j$ and

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

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 4.3). Thus each term of $\textbf{Fold}\left[f, \boldsymbol{z}\right]$ lies in the vector space C_{i+1}, which concludes the proof. □

We discuss the effect of the folding operation on a function which is far from the code. Roughly speaking, we want to show that, if f is δ-far from C_i, then the folding $\textbf{Fold}\left[f, \boldsymbol{z}\right]$ of f is almost δ-far from C_{i+1} with high probability over $\boldsymbol{z} \in \mathbb{F}^2$. We start with the notion of weighted agreement.

Definition 7.5 (Weighted agreement)**.** *For any function $\eta \in [0,1]^{\mathcal{P}}$, 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 a subspace $V \subset \mathbb{F}^{\mathcal{P}}$ 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 $u \in \mathbb{F}^{\mathcal{P}}$,

$$\omega_\eta(u, V) \leq 1 - \Delta(u, V). \tag{24}$$

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 7.6. *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*

$$\Pr_{z \in \mathbb{F}}\left[\omega_\eta\left(\sum_{i=0}^{l-1} z^i u_i, V\right) > 1 - \delta\right] \geq \frac{l-1}{|\mathbb{F}|}\left(\frac{2}{\varepsilon}\right)^{l+1}, \tag{25}$$

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 7.7. *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 $\varepsilon \in (0,1[$ 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

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

Proving Corollary 7.7 requires the lemma stated next. We prove Corollary 7.7, then prove Lemma 7.8.

Lemma 7.8. *Let $i \in \{0, \ldots, r-1\}$, $D_i \in \mathrm{Div}(\mathcal{X}_i)$ and $\mu_i \in \mathbb{F}(\mathcal{X}_i)$ satisfying Definition (4.2). Consider a divisor $D_{i+1} \in \mathrm{Div}(\mathcal{X}_{i+1})$ that is (D_i, μ_i)-compatible in the sense of Definition 4.3.*

Fix $j \in \{0, \ldots, p_i - 1\}$. Then a function $g \in \mathbb{F}(\mathcal{X}_{i+1})$ 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 7.7. Let $f : \mathcal{P}_i \to \mathbb{F}$ be an arbitrary function. According to Equation (22), there exist p_i function $f_j : \mathcal{P}_{i+1} \to \mathbb{F}$ such that for any $\boldsymbol{z} = (z_1, z_2) \in \mathbb{F}^2$,

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

Rewrite $\textbf{Fold}\left[f, \boldsymbol{z}\right]$ as a polynomial function in z_2, i.e. $\textbf{Fold}\left[f, \boldsymbol{z}\right] = f_{z_1} + z_2 f'_0 + z_2^2 f'_1 + \cdots + z_2^{p_i} f'_{p_i-1}$ where we set $f_{z_1} \coloneqq \sum_{j=0}^{p_i-1} z_1^j f_j$ and $f'_j \coloneqq \nu_{i+1,j} f_j$. Finally, set

$$K_0 \coloneqq \frac{p_i - 1}{|\mathbb{F}|}\left(\frac{4}{\varepsilon}\right)^{p_i} \qquad \text{and} \qquad K_1 \coloneqq \frac{p_i}{|\mathbb{F}|}\left(\frac{4}{\varepsilon}\right)^{p_i+1}.$$

Let us prove the corollary by contrapositive: assume that

$$\Pr_{\boldsymbol{z}\in\mathbb{F}^2}\left[\omega_\theta(\textbf{Fold}\left[f, \boldsymbol{z}\right], C_{i+1}) > 1 - \delta + \varepsilon\right] > K_0 + K_1$$

or in other words that $\Pr_{z_1\in\mathbb{F}}\left[\Pr_{z_2\in\mathbb{F}}\left[\omega_\theta(\textbf{Fold}\left[f, \boldsymbol{z}\right], C_{i+1}) > 1 - \delta + \varepsilon\right] > K_0\right] > K_1$.

Fix $z_1 \in \mathbb{F}$ such that $\Pr_{z_2\in\mathbb{F}}\left[\omega_\theta(\textbf{Fold}\left[f, \boldsymbol{z}\right], C_{i+1}) > 1 - \delta + \varepsilon\right] > K_0$. By Proposition 7.6, there exist $v_{z_1}, v'_1, \ldots, v'_{p_i-1} \in C_{i+1}$ and $\mathcal{T}' \subset \mathcal{P}$ such that

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

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

It means that

$$\Pr_{z_1\in\mathbb{F}}\left[\omega_\theta(f_{z_1}, C_{i+1}) \geq 1 - \delta + \frac{\varepsilon}{2}\right] \geq \Pr_{z_1\in\mathbb{F}}\left[\Pr_{z_2\in\mathbb{F}}\left[\omega_\theta(\textbf{Fold}\left[f, \boldsymbol{z}\right], C_{i+1}) > 1 - \delta + \varepsilon\right] > K_0\right] > K_1.$$

The polynomial form of f_{z_1} in z_1 enables us to reapply Proposition 7.6: 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 $j \in \{0, \dots, p_i - 1\}$, $v_j|_{\mathcal{T}} = f_j|_{\mathcal{T}}$.

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, for every $j \in \{0, \dots, p_i - 1\}$, the evaluations of v'_j and $\nu_{i+1,j} v_j$ on $\mathcal{P}_{i+1}$ are equals. They are codewords of C_{i+1}, thus this implies that both functions v_j and $\nu_{i+1,j} v_j$ belong to $L(D_{i+1})$. By Lemma 7.8, 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

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

By definition of the divisors $E_{i,j}$ (5), the function v belong to $L(D_i)$. Now let us prove that it agrees with f on $S_{\mathcal{T}} \coloneqq \bigsqcup_{P\in\mathcal{T}} S_P$.

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 7.8. Assume that $g \in L(E_{i,j})$. Then the second and third items of Definition 4.3 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 $D'_{i+1} = D_{i+1} - \operatorname{div}_\infty(\nu_{i+1,j}) = E_{i,j}$ by the third item of Definition 4.3. □

7.3 IOPP for foldable AG codes

Let $C_0 = C(\mathcal{X}_0, \mathcal{P}_0, D_0)$ be a foldable AG code over an alphabet $\mathbb{F}$. Given a family of folding operators defined as per Definition 7.2, [ABN21] yields an IOPP for C_0, which is abstracted from the FRI protocol of [BBHR18a]. We informally describe the IOPP system (P, V) for testing proximity of a function $f^{(0)} : \mathcal{P}_0 \to \mathbb{F}$ to C_0, then give its properties. A formal description will be provided in Section 8 for instantiations with concrete AG codes.

As in the FRI protocol, the IOPP is divided in two phases, referred to as COMMIT and QUERY. Before any interaction, P and V agree on:

- a $(\mathcal{X}, \mathcal{G})$-sequence of curves $(\mathcal{X}_i)$, for which we denote the length of the composition serie of $\mathcal{G}$ by r.
- a sequence of codes (C_i) where for each $i \in \{0, \dots, r\}$, $C_i = (\mathcal{X}_i, \mathcal{P}_i, D_i)$ and $\mathcal{X}_i, \mathcal{P}_i$ and D_i are defined as per Section 4,
- a sequence of functions $(\mu_i) \in \mathbb{F}(\mathcal{X}_i)$ satisfying Definition 4.2,
- a sequence of balancing functions $(\nu_{i+1})_{0 \le i < r}$ of p_i-tuples of functions in $\mathbb{F}(\mathcal{X}_{i+1})$ such that $\nu_{i+1} = (\nu_{i+1,j})_{0<j<p_i}$ and $\nu_{i+1,j}$ satisfies (6).

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

- The COMMIT phase is an interaction over r rounds between P and V. For each round $i \in \{0, \dots, r-1\}$, 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 $\mathbf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right]$. 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 7.3). 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
$$\mathbf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](P) := \sum_{j=0}^{p_i-1} \left(z_1^{(i)}\right)^j a_{j,P} + \sum_{j=0}^{p_i-1} \left(z_2^{(i)}\right)^{j+1} \nu_{i+1,j}(P) a_{j,P}.$$
- 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
$$f^{(i+1)} = \mathbf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right] \tag{26}$$
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 [BBHR18a], we call this step of verification a *round consistency test*. 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 $Q_{i+1} := \pi_i(Q_i)$ to be the random point where Equation (26) 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 $f^{(r)} \in C_r$.

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 7.9. *Let $C_0 = C(\mathcal{X}_0, \mathcal{P}_0, D_0)$ be a foldable AG code of length $n := |\mathcal{P}_0|$. By definition, C_0 admits a solvable group $\mathcal{G} \in \mathrm{Aut}(\mathcal{X}_0)$ such that $|\mathcal{G}| > n^e$ for a certain $e \in (0,1)$ and induces 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 $\gamma := \min\left(J_\varepsilon^{p_{max}}(\lambda), \frac{1}{2}(\lambda + \frac{\varepsilon}{2})\right)$. There 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 $1 - \mathsf{err}_{commit}$ over the randomness of the verifier during the COMMIT phase, where*

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

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*

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

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

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

Moreover, the IOPP system is public-coin, has round complexity $\mathsf{r}(n) < \log n$, proof length $\mathsf{l}(n) < n$ and query complexity $\mathsf{q}(n) < t p_{max} \log n + n^{1-e}$.

Proof. Lemma 7.3, Proposition 7.4 and Corollary 7.7 satisfy the conditions of [ABN21, Theorem 1]. Completeness and soundness are given by [ABN21, Theorem 1]. Let us prove the rest of the theorem:

(Round complexity) We have that $\prod_{i=0}^{r-1} p_i = \frac{n}{n_r}$, 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 $\mathsf{r}(n) \leq \log_2 n - \log_2 n_r < \log_2 n$.

(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, $\mathsf{q}(n) = t\left(\sum_{i=0}^{r-1} p_i\right) + n^{1-e} \leq t r p_{max} + n^{1-e}$.

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

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

□

8 Proximity tests for AG codes on Kummer curves and Hermitian towers

When we instantiate the AG-IOPP proposed in Section 7.3 for the setting of Kummer curves (Section 5) and curves in the Hermitian tower (Section 6), we end up with a membership test to a RS code. An RS code is itself a foldable AG code (see Section 4.3). In order to lower verifier complexity, we can extend the AG-IOPP by replacing the final test by an IOPP for RS code. This enhanced AG-IOPP is examined in this section.

8.1 How to iterate the folding to reach a code of dimension 1

We consider a sequence of foldable AG codes $(C_i)_{0\leq i\leq s}$ as provided by Section 5 (Kummer curves) or Section 6 (tower of Hermitian curves). The code $C_s = C(\mathbb{P}^1, \mathcal{P}_s, D_s)$ corresponds to a Reed-Solomon code $\mathsf{RS}[\mathbb{F}, \mathcal{P}_s, d] = \{f : \mathcal{P}_s \to \mathbb{F}; \deg f \leq d\}$, where the degree bound depends on the parameters of the code C_0. 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 FRI protocol [BBHR18a].

As in Example 4.3, we set $d_0 = d$ 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 4 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 7.3. We obtain a r-rounds IOPP system (P, V) for C_0, which is described below.

The prover P and the verifier V are given as input the description of the code C_0. The verifier V is given oracle access to a function $f^{(0)} : \mathcal{P}_0 \to \mathbb{F}$, which is also given as explicit input to the prover P.

COMMIT phase:

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

QUERY phase:

1. Repeat t 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 $\mathbf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](Q_{i+1})$,

iv. If $i < r-1$: return **reject** if and only if $f^{(i+1)}(Q_{i+1}) \neq \mathbf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](Q_{i+1})$

v. If $i = r-1$: return **reject** if and only if $\beta \neq \mathbf{Fold}\left[f^{(i)}, \boldsymbol{z}^{(i)}\right](Q_{i+1})$

2. Return **acccept**.

8.2 Properties of the AG-IOPP with Kummer curves

Assume $C_0 = C(\mathcal{X}_0, \mathcal{P}_0, D_0)$ is a foldable AG code of blocklength $n_0 = |\mathcal{P}_0|$ on a Kummer curve $\mathcal{X}_0$ (cf. Proposition 5.5). This means that $\mathcal{X}_0$ 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 $\eta \in \mathbb{N}$.

Proposition 5.10 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 $C_s = C(\mathcal{X}_s, \mathcal{P}_s, D_s)$ corresponds to a RS code

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

of blocklength $|\mathcal{P}_s| = \alpha$, which is itself a foldable AG code (see Example 4.3).

Theorem 8.1 (Kummer case)**.** *Let $C = (\mathcal{X}_0, \mathcal{P}_0, D_0)$ be a foldable AG code on a Kummer curve satisfying the hypotheses of Proposition 5.5 with N a η-smooth integer. Denote $n = |\mathcal{P}_0|$. The IOPP (P, V) described in Section 8.1 has perfect completeness and soundness as stated in Theorem 7.9. Moreover, for t repetitions of the QUERY phase, we have:*

rounds complexity	$\mathsf{r}(n)$	$< \log n$,
proof length	$\mathsf{l}(n)$	$< n$,
query complexity	$\mathsf{q}(n)$	$\leq t\eta \log_2 n + 1$,
prover complexity	$\mathsf{t_p}(n)$	$= O_\eta(n)$,
verifier decision complexity	$\mathsf{t_v}(n)$	$= O_\eta(t \log n)$.

Proof. Noticing that the round complexity is now $\mathsf{r}(n) = s + s'$, straightforward calculations show that complexity, query complexity and proof length computed in the proof 7.9 still hold. Completeness and soundness also follow from [ABN21]. We estimate prover complexity and verifier complexity below.

(Prover complexity) Fix a round index $i < r-1$. The balancing functions $\nu_{i+1,j} : \mathcal{P}_{i+1} \to \mathbb{F}$ can be precomputed since they do not depend on $f^{(i)}, \boldsymbol{z}^{(i)}$ (see Remark 8.2). To simplify notation, denote $f = f^{(i)}$. For any $\boldsymbol{z} = (z_1, z_2) \in \mathbb{F}^2$, computing the successive powers $(z_1^j, z_2^j)_{0 \leq j < p_i}$ takes $2(p_i - 2)$ multiplications. For each $P \in \mathcal{P}_{i+1}$, an honest prover must compute the coefficients $(a_{j,P})_{0 \leq j, <P}$ 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. Notice that $\mu_i = y$, so computing $\mu_i(\widehat{P})$ for $\widehat{P} \in S_P$ is done for free. Univariate interpolation for a polynomial of degree $< p_i$ can be done in $O(p_i^2)$ by Lagrange interpolation. Overall, one can honestly evaluate $\mathbf{Fold}\left[f, \boldsymbol{z}\right] : \mathcal{P}_{i+1} \to \mathbb{F}$ with $|\mathcal{P}_{i+1}|\, O(p_i^2)$ 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 $f^{(1)}, \ldots, f^{(r-1)}$ is $O_\eta(n)$.

(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 $O(p_i^2)$ operations, then computes $\mathbf{Fold}[f, \boldsymbol{z}](P)$ in a number of operations which is independent of n. Hence, verifier complexity for repetition parameter t is $\mathsf{t_v}(n) = O_\eta(t \log(n))$.

□

Remark 8.2. We give the cost of precomputing the evaluation tables of the balancing functions. Letting $\nu_{i+1,j}$ be as defined in proof of Lemma 5.4, 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. Thus, the evaluations of $\nu_{i+1,1}, \ldots \nu_{i+1,p_i-1}$ on $\mathcal{P}_{i+1}$ are obtained with $O((\log m + p_i)\,|\mathcal{P}_{i+1}|)$ operations.

We give an example of an AG code over a Kummer curve where $p_{max} = 2$.

Example 8.3. On $\mathbb{F}_{q^2}$ with $q = 2^{61} - 1$ (9^{th} Mersenne prime), we consider the curve

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

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 $\mathcal{P}_0 \subset \mathcal{X}_0(\mathbb{F}_{q^2})$ 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 $\varepsilon = 2^{-6.55}$. By Theorem 7.9,

$$\mathsf{err}_{commit} \leq \frac{\log(n)}{\left|\mathbb{F}_{q^2}\right|} \left(1 + \frac{4}{\varepsilon}\right) \left(\frac{4}{\varepsilon}\right)^2 \approx 2^{4.33+6+3\cdot 6.55-121} \leq 2^{-91}$$

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

where $1 - \delta = (1 - \lambda + \varepsilon)^{\frac{1}{3}} \leq 0,51384$. Hence

$$\mathsf{err}_{query}(\delta) \leq 0,51384 + \frac{20}{2^{6.55}} \approx 0,72728.$$

By running the QUERY phase with repetition parameter $t \geq 199$, we get $(\mathsf{err}_{query})^t \leq 2^{-91}$ and $\mathsf{err}(\delta) \leq 2^{-90}$. 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 $r + 1$.

8.3 Properties of the AG-IOPP with towers of Hermitian curves

Theorem 8.4. *Let $C = (\mathcal{X}, \mathcal{P}, D)$ be a foldable AG code with alphabet $\mathbb{F} = \mathbb{F}_{q^2}$ on a tower of Hermitian curves satisfying the hypotheses of Proposition 6.9. Letting e be the index of the curve $\mathcal{X}$ in the Hermitian tower $(\mathcal{X}_i)_{i\geq 0}$, the length $n = |\mathcal{P}|$ of C is at most q^{e+2}. The IOPP (P, V) described in Section 8.1 has perfect completeness, and soundness as stated in Theorem 7.9. Moreover, we have:*

rounds complexity	$\mathsf{r}(n)$	$< \log n$,
proof length	$\mathsf{l}(n)$	$< n$,
query complexity	$\mathsf{q}(n)$	$\leq tq \log n + 1$,
prover complexity	$\mathsf{t_p}(n)$	$= O(n \cdot \mathsf{M}(q) \log(q))$,
verifier decision complexity	$\mathsf{t_v}(n)$	$= O(\log n \cdot \mathsf{M}(q) \log(q))$.

Proof. The proof follows from proof of Theorem 8.1, replacing η by q. Prover and verifier complexities are computed from the cost of computing the coefficients of a univariate polynomial of degree less than q from its evaluation on points forming an arithmetic progression in $\mathbb{F}_{q^2}$. This interpolation task can be done in $\mathsf{M}(q)\log q + O(\mathsf{M}(q))$ base field operations [BS05], where $\mathsf{M}(d)$ denotes the cost of multiplying two degree-d univariate polynomials. □

Given a foldable code as in Proposition 6.9, the IOPP constructs a sequence of codes as follow:

$$C_i \coloneqq C(\mathcal{X}_{i_{\max}-i}, \mathcal{P}_{i_{\max}-i}, D_{i_{\max}-i}) \text{ where } \mathcal{P}_{i-1} = \pi_i(\mathcal{P}_i) \text{ and } D_i = d_i P_\infty^{(i)}$$

with the integers d_i defined recursively by

$$d_{i-1} \coloneqq \left\lfloor \frac{d_i}{q} \right\rfloor + 2g(\mathcal{X}_{i-1}).$$

Unlike the Kummer case, we have to increase the degree of divisor by twice the genus of the curve at each step to make sure the compatibility hypotheses of Definition 4.3 are valid. This has a counterpart: the dimension of the codes C_i decreases much slowly than their block length. A foldable code in the sense of Definition 4.5 may induce of a sequence of codes in which the last code $C_{i_{\max}}$ is trivial. In this case, the protocol would no longer be sound. We thus need to control the dimension of the code $C_{i_{\max}}$. This is the purpose of the remaining of this section.

Remark 8.5. In light of the Kummer case in which the group $\mathbb{Z}/N\mathbb{Z}$ is factored as much as possible, if q is some prime power $q = p^\ell$, we could split the group $\mathbb{Z}/q\mathbb{Z}$ acting at each level to make ℓ intermediary steps. The verifier and the prover would perform polynomial interpolations of degree p, and the verifier would make only p queries at each step. However, for each of these steps, we would have to make the new divisor grow to fulfill the compatibility conditions as above. If the rate increases too much, the relative minimum distance drops and the total number queries to target a designated soundness may be tremendous. The loss in terms of soundness error per QUERY phase seem to be much more significant than the aforementioned advantages.

8.3.1 Bounding the rate of the underlying Reed-Solomon code

We aim to bound the dimension of the code Reed-Solomon code $C_{i_{\max}}$. Let us compute the degree $d_{i_{\max}}$ of the divisor $D_{i_{\max}}$ on $\mathbb{P}^1$.

Lemma 8.6. *For $1 \le j \le i_{\max}$, we have*

$$d_{i_{\max}-j} \le \left\lfloor \frac{d_{i_{\max}}}{q^j} \right\rfloor + \sum_{k=1}^{j} \left\lfloor \frac{2g_{i_{\max}-k}}{q^{j-k}} \right\rfloor + (j-1).$$

Proof. It follows from the definition of the degrees d_i given in (17) and by induction on j. □

Using Lemma 8.6 which bounds the genera g_i for $i \le i_{\max}$, we can get an upperbound on d_0.

Corollary 8.7. *Let us assume that $2(i_{\max}-1) < q$. The degree d_0 of the divisor D_0 on $\mathbb{P}^1$ is bounded from above by*

$$d_0 \le \left\lfloor \frac{d_{i_{\max}}}{q^{i_{\max}}} \right\rfloor + (i_{\max}-1)\left(1 + \frac{i_{\max}}{6} \cdot (3q - 4 + 2i_{\max})\right)$$

Proof. By Lemma 8.6, we have the following bound over d_0:

$$d_0 \leq \left\lfloor \frac{d_{i_{\max}}}{q^{i_{\max}}} \right\rfloor + \sum_{i=0}^{i_{\max}-1} \left\lfloor \frac{2g_i}{q^i} \right\rfloor + i_{\max} - 1.$$

It is thus enough to estimate the sum $\sum_{k=0}^{i_{\max}-1} \left\lfloor \frac{2g_k}{q^k} \right\rfloor$. By Proposition B.1,

$$\begin{aligned}\sum_{k=0}^{i_{\max}-1} \left\lfloor \frac{2g_k}{q^k} \right\rfloor &\leq \sum_{k=0}^{i_{\max}-1} (kq + k(k-1)) \\ &= (q-1) \cdot \frac{i_{\max}(i_{\max}-1)}{2} + \frac{i_{\max}(i_{\max}-1)(2i_{\max}-1)}{6} \\ &= \frac{i_{\max}(i_{\max}-1)}{2} \cdot \left(q - \frac{4}{3} + \frac{2i_{\max}}{3} \right),\end{aligned}$$

which gives the expected result. □

Depending on the length of the code $C_{i_{\max}}$, we can determine a sufficient condition on $i_{\max}$ that ensures that the code C_0 is not trivial. Let us denote by n_0 the size of $\mathcal{P}_0$. It satisfies $n_0 \leq q^2$. The rate of C_0 is equal to $\frac{d_0+1}{n_0}$. Also we have $n_{i_{\max}} := \#\mathcal{P}_{i_{\max}} = q^{i_{\max}} n_0$.

Corollary 8.8. *Let us fix $\rho \in (0,1)$. If*

$$\left\lfloor \frac{d_{i_{\max}}}{q^{i_{\max}}} \right\rfloor + (i_{\max}-1)\left(1 + \frac{i_{\max}}{6} \cdot (3q - 4 + 2i_{\max})\right) + 1 < \rho n_0,$$

then the rate of the code C_0 is less than ρ.

8.3.2 Foldable codes with constant rate which are endowed with an IOPP with designed soundness

In this paragraph, we focus on foldable codes of the form

$$C = C_0 = C\left(\mathcal{X}_{i_{\max}}, \mathcal{X}_{i_{\max}}(\mathbb{F}_{q^2}) \setminus \{P_\infty^{(i_{\max})}, (2\alpha+1)g_{i_{\max}})P_\infty^{(i_{\max})}\right) \tag{20}$$

for some $\alpha > 1/2$, as in Section 6.2.1. The evaluation set $\mathcal{P}_{i_{\max}}$ is the whole set of rational points of $\mathcal{X}_{i_{\max}}$ minus the point of at infinity $P_\infty^{(i_{\max})}$, *i.e.* $n_{i_{\max}} = q^{i_{\max}+2}$.

Proposition 8.9. *Let us fix $\rho \in (0,1)$. The rate of the RS code $C_{i_{\max}}$ below C_0 is less than ρ if*

$$2i_{\max}^3 + 3i_{\max}^2(2\alpha + q - 1) + i_{\max}(6\alpha(q-1) + 7) - 6\rho q^2 < 0.$$

Proof. For q large enough, we can assume that $2i_{\max} - 1 < q$. Using Proposition B.1, we get an upperbound over $d_{i_{\max}}$:

$$d_{i_{\max}} \leq (2\alpha+1)\frac{i_{\max}}{2} q^{i_{\max}}(q + (i_{\max}-1)).$$

From Corollary 8.8, a sufficient condition for the underlying RS code to have a rate less than ρ is

$$(2\alpha+1)\frac{i_{\max}}{2}(q + (i_{\max}-1)) + (i_{\max}-1)\left(1 + \frac{i_{\max}}{6}(3q - 4 + 2i_{\max})\right) + 1 < \rho q^2$$

Multiplying the inequality by 6, expanding and simplifying, we get our condition.

□

Now assume that $i_{\max} = q^{\varepsilon}$ for $\varepsilon \in (0,1)$. In the constant rate regime described in Lemma 6.10, we have $\alpha i_{\max} = Rq$. The condition above becomes

$$2i_{\max}^3 + 3i_{\max}^2(q-1) + i_{\max}(6Rq+7) < 6q^2\left(\rho - R\left(1-\frac{1}{q}\right)\right).$$

If $R\left(1-\frac{1}{q}\right) < \rho$, the right handside is positive. Let us give a rough estimation of the largest ε such that $i_{\max} = q^{\varepsilon}$ satisfies this inequality. The left handside begin equivalent to $3q^{1+2\varepsilon}$, we have $\varepsilon \simeq \frac{1}{2}(1 + \log_q\left(\rho - R\left(1-\frac{1}{q}\right)\right)$.

Table 1 displays some examples of level $i_{\max}$ and initial rate R of foldable codes for which the AG-IOPP reduce the proximity test to testing RS codes of rate ρ. In terms of the soundness of the protocol, it means that λ as defined in Theorem 7.9 is greater than $1-\rho$.

q	$i_{\max}$	n	R	$1-\rho >$
2^4	3	2^{20}	1/8	1/3
2^5	5	2^{35}		
2^4	4	2^{24}	1/16	1/3
2^5	3	2^{25}		3/4
	5	2^{35}		1/2
2^6	4	2^{36}		3/4
	5	2^{42}		2/3
	7	2^{54}		1/2
2^4	3	2^{20}	1/32	1/2

Table 1: Example of parameters of foldable codes of rate R along the Hermitian tower. Alphabet is $\mathbb{F}_q^2$ and block length is n. The last column gives a bound on the minimal distance of the RS code.

Acknowledgments

The third author thanks Marc Perret for his precious advices in the early days of this project. The authors are grateful to Daniel Augot for suggesting to work on this project and for many valuable discussions. They also thank Eli Ben-Sasson and Alessandro Chiesa for their helpful insights. 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*. This work was also funded in part by the grant ANR-21-CE39-0009 - BARRACUDA from the French National Research Agency.

References

[ABN21] Daniel Augot, Sarah Bordage, and Jade Nardi. Efficient multivariate low-degree tests via interactive oracle proofs of proximity for polynomial codes. *Electron. Colloquium Comput. Complex.*, page 118, 2021.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight Sublinear Arguments Without a Trusted Setup. In Bha-

vani 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.

[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.

[BBHR18a] 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.

[BBHR18b] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. *IACR Cryptol. ePrint Arch.*, page 46, 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.

[BCL20] Jonathan Bootle, Alessandro Chiesa, and Siqi Liu. Zero-Knowledge Succinct Arguments with a Linear-Time Prover. *IACR Cryptol. ePrint Arch.*, page 1527, 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.

[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.

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

[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.

[HY18] Chuangqiang Hu and Shudi Yang. Multi-point codes over Kummer extensions. *Designs, Codes and Cryptography*, 86:211–230, 2018.

[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.

[Lac92] Gilles Lachaud. Artin-Schreier curves, exponential sums, and coding theory. *Theoretical Computer Science*, 94(2):295–310, 1992.

[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.

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

[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.

[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.

[PzSJ91] Ruud Pellikaan, Ba zhong Shen, and Gerhard J. M. van Wee. Which linear codes are algebraic-geometric? *IEEE Trans. Inf. Theory*, 37:583–602, 1991.

[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.

[Sta21] StarkWare. ethSTARK Documentation. *IACR Cryptol. ePrint Arch.*, page 582, 2021.

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

[Sti08] Henning Stichtenoth. *Algebraic Function Fields and Codes*. Springer Publishing Company, Incorporated, 2nd edition, 2008.

[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.

A Proof of Proposition 7.6

Proposition 7.6 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 $v_0, \dots, v_{l-1} \in V$ 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 $i \in \{0, \dots, l-1\}$ 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

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

which gives the first item of the proposition.

B Properties of the genera of the curves in the Hermitian tower

To estimate the parameters of the foldable codes we define along the Hermitian tower, we need to handle the genera of the curves in this tower. From the formulae (15), we deduce deduce a bound over the genus of the curve $\mathcal{X}_i$ for small i (Proposition B.1) and the asymptotic behaviour of the ratio of g_i by q^{i+2} for $i = q^\varepsilon$ when q goes to infinity (Lemma B.2).

Proposition B.1. *For $i \geq 1$, we have*

$$g_i \leq \frac{q^{i+1}}{2} \sum_{k=1}^{i} \binom{i}{k} \frac{1}{q^{k-1}} \leq \frac{iq^{i+1}}{2} \sum_{k=1}^{i} \left(\frac{i}{q}\right)^{k-1} \leq \frac{i}{2} q^{i+1} + \frac{i(i-1)}{2} q^i,$$

the last inequality holding only if $2(i-1) < q$.

Proof. Starting from the second formula of (15), we can write

$$g_i = \frac{1}{2} \cdot \left(q^{i+1} \sum_{k=1}^{i} \left(1 + \frac{1}{q}\right)^{k-1} + 1 - (1+q)^i \right) \leq \frac{q^{i+1}}{2} \cdot q \cdot ((1 + 1/q)^i - 1) = \frac{q^{i+1}}{2} \cdot \sum_{k=1}^{i} \binom{i}{k} \frac{1}{q^{k-1}},$$

using that the term outside the geometric sum is non positive. Note that if $k \geq 2$, then we can bound the binomials coefficients as follows

$$\binom{i}{k} = \frac{i(i-1)\cdots(i-k+1)}{k(k-1)\cdots 2} \leq \frac{i(i-1)^{k-1}}{2},$$

as the denominator is greater than 2 and the factors $i-1, i-2, ..., i-k+1$ are all lesser than $i-1$. Factoring and using this upperbound over the binomials coefficients, we get

$$g_i \leq \frac{q^{i+1}}{2} \cdot \left(i + \frac{i}{2} \sum_{k=2}^{i} \left(\frac{i-1}{q} \right)^{k-1} \right) = \frac{iq^{i+1}}{2} \left(1 + \frac{1}{2} \cdot \left(\frac{i-1}{q} \right) \cdot \sum_{k=2}^{i} \left(\frac{i-1}{q} \right)^{k-2} \right).$$

Assuming that $2(i-1) < q$, we can bound the sum $\sum_{k=2}^{i} \left(\frac{i-1}{q} \right)^{k-2}$ by 2, which concludes the proof. □

Lemma B.2. *Fix $\varepsilon \in (0,1)$ and set $i = q^{\varepsilon}$. Then*

$$\frac{g_i}{q^{i+2}} \underset{q \to \infty}{\sim} \frac{1}{2q^{1-\varepsilon}}.$$

Proof. From the first formula of (15), we get

$$\frac{2gi_{\max}}{q^{i_{\max}+2}} = \left(1 - \frac{1}{q^2} \right) \left[\left(1 + \frac{1}{q} \right)^{q^{\varepsilon}} - 1 \right] + \frac{1}{q^{i_{\max}+2}} - \frac{1}{q^2}$$

Let us examine the asymptotic behaviour of $\left(1 + \frac{1}{q}\right)^{q^{\varepsilon}}$ when q goes to infinity. Set $h = q^{-1}$.

$$\left(1 + \frac{1}{q} \right)^{q^{\varepsilon}} = \exp\left(h^{1-\varepsilon} \cdot \frac{\log(1+h)}{h} \right) = \exp\left(h^{1-\varepsilon} \left(1 - \frac{h}{2} + o(h) \right) \right) = 1 + h^{1-\varepsilon} + o(h^{1-\varepsilon})$$

Therefore, we have

$$\left(1 + \frac{1}{q} \right)^{q^{\varepsilon}} - 1 \sim \frac{1}{q^{1-\varepsilon}}.$$

□