SLVer Bullet: Straight-Line Verification for Bulletproofs

Brandon Goodell[1], Rigo Salazar[1], Freeman Slaughter[1,2,3,✉], and Luke Szramowski[1]

[1]Cypher Stack
[2]Clemson University, Clemson SC, USA
[3]University of South Florida, Tampa FL, USA
✉fslaughter@usf.edu

January 13, 2026

Abstract

Eagen introduced a framework for proof-of-computation over elliptic curves, replacing costly group operations with efficient computations over a function field. This work lacks clear protocol descriptions, formal security proofs, and a thorough efficiency analysis; we improve upon the literature, present a detailed protocol description, then analyze completeness, soundness, and complexity. We apply our construction to the Schnorr and Bulletproofs protocols, highlighting how this reduces verifier costs significantly compared to the original proposals. This tradeoff most clearly benefits privacy-centered constructions like blockchain cryptocurrencies and Tor routing - where proofs must be verified repeatedly - but more generally may be used to improve verification time in generic curve-based zero-knowledge proof systems.

1 Introduction

Eagen [Eag22] proposed a method to verify sums of points in a prime-order elliptic curve group $\mathbb{G}$ over a prime field K. Parker [Para] applied the method within a rank-1 constraint system to verify scalar multiplication. We call the method "straight-line verification" (SLV) because it uses straight lines as random challenges in the Fiat-Shamir heuristic. In Eagen's SLV, logarithmic derivatives simplify the verification equations. Also, provers pre-compute expensive operations for verifiers, who only compute linear combinations in the base field. Such precomputation helps when proofs are verified many times. Unfortunately, Eagen's SLV suffers from problems.

Eagen assumed that a certain system of polynomial equations admit rational solutions. In general, it does not. We discuss this in detail in Section A.4.1, and explicitly state this system in Equation A.2. Bassa [Bas] contributed by stating an explicit protocol and sketching soundness arguments; to bypass the rationality issue with Equation A.2, Bassa lifted the problem to the surface

Keywords: zero-knowledge proof, elliptic curve cryptography, divisor, Bulletproofs, cryptocurrency. MSC: 11T71, cryptography.

of pairs of elliptic curve group points, over which solutions *are* rational. Yet the justification for the verification equations remains unsatisfactory, and the soundness arguments yield an unnecessarily high soundness error.

We formalize an SLV drawing on the work of both Eagen and Bassa. Our work clarifies and extends the existing literature. We derive verification equations, and observe they are a simple reduction of Eagen's. Specifically, that three distinct non-identity collinear points, say P, Q, and R, satisfy $P + Q + R = O$ according to the group law. If their x-coordinates are x_P, x_Q, and x_R, and the interpolating line has slope λ, then the Weierstrass form gives $\lambda^2 = x_P + x_Q + x_R$ in the base field. For any derivation d on the function field $K(\mathcal{E})$ over K, we get $dX_R = -dX_P - dX_Q$, a key equation in our reduction.

We also use Eagen's techniques to reduce the number of prover field divisions to just a single one. We present security proofs, efficiency analyses, and applications to discrete-logarithm-based protocols. As examples, we transform the Schnorr identification and Bulletproofs range proving systems into SLV proofs. We select these protocols as they are simple, secure, popular, and readily applicable in cryptocurrencies such as Monero or Salvium.

1.1 Paper Structure

We assume the reader knows basic concepts of groups, rings, fields, and the affine and projective planes. Our notation largely follows [Sil09], with some departures concerning divisors (see Appendix A). Section 2 presents the protocol. Section 2.3 computes completeness and soundness errors. Section 3 illustrates the computational advantages of divisor-based verification within Schnorr and Bulletproof systems. Appendix A provides background and notation.

1.2 Notation and Assumptions, Briefly

Let $\mathbb{0}$ and $\mathbb{1}$ represent the all-zero and all-one vectors, $\mathbb{N}$ the natural numbers, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$, $\mathbb{Z}$ the integers. For a set S, we use $x \xleftarrow{\$} S$ to mean that the element x is an independent uniformly-distributed sample from set S. For any sets A and B sets, let $B^A = \{f : A \to B\}$ be the functions from A to B. For any field F and function ϕ such that $\mathrm{Im}(\phi) \subseteq F$, let $F^\times$ be the multiplicative subgroup of F and $\mathrm{supp}(\phi) = \{x \mid \phi(x) \neq 0\}$. Let $\mathbb{A}^2(F)$ and $\mathbb{P}^2(F)$ be the affine and projective planes over F. Elements of $\mathbb{A}^2(F)$ are pairs in F^2. Elements of $\mathbb{P}^2(F)$ are equivalence classes $(x\!:\!y\!:\!z)$ of triples from $F^3 \setminus \{(0,0,0)\}$ such that $(x', y', z') \in (x\!:\!y\!:\!z)$ if and only if $(x', y', z') = (\lambda x, \lambda y, \lambda z)$ for some $\lambda \in F^\times$. The affine-to-projective embedding is defined by $(x, y) \hookrightarrow (x\!:\!y\!:\!1)$.

Let $p, q \in \mathbb{N}$ prime numbers such that $q > 3$, $K = \mathbb{Z}/p\mathbb{Z}$ the field with p elements, X, Y indeterminates over K, $K[X, Y]$ the polynomial ring, $\mathcal{E}$ an elliptic curve over K with elliptic curve group $\mathbb{G} = \mathbb{G}(\mathcal{E})$ written additively and such that $|\mathbb{G}| = q$, and $K(\mathcal{E})$ the function field. For $\mathbb{G}$, there exists $e = Y^2 - X^3 - \alpha X - \beta \in K[X, Y]$ for some $(\alpha, \beta) \in K^2$ with homogenization $e^* = Y^2Z - X^3 - \alpha XZ^2 - \beta Z^3 \in K[X, Y, Z]$, such that the elements of $\mathbb{G}$ are exactly the roots of e^*. The non-identity elements of $\mathbb{G}$ are images $(x, y) \hookrightarrow (x\!:\!y\!:\!1)$, and the identity $O = (0\!:\!1\!:\!0)$ is the point at infinity on $\mathcal{E}$. The function field $K(\mathcal{E})$ is the field of fractions of $\mathcal{R} = K[X, Y]/(e)K[X, Y]$

where $(e) \subseteq K[X,Y]$ is the principal ideal generated by e. Each $f+(e) \in \mathcal{R}$ is a coset of polynomials from $K[X,Y]$ with a representative of the form $f_1(X) + f_2(X)Y$ for some $f_1, f_2 \in K[X]$, as in Section A.3, inducing an *evaluation map* on all of $\mathbb{G}$ defined by $(f+(e))(P) = f^*(x_P, y_P, z_P)$, where $f^*(X,Y,Z) = Z^{\deg f} f(\frac{X}{Z}, \frac{Y}{Z})$ is the homogenization. If $f \in K(\mathcal{E})$, then $f = \frac{f_1(X)+f_2(X)Y+(e)}{f_3(X)+(e)}$ for some $f_1, f_2, f_3 \in K[X]$ as in Section A.3, yielding an evaluation map defined on $\operatorname{supp}(f_3 + (e)) \subseteq \mathbb{G}$. Projective points in $\operatorname{supp}(f_3 + (e)) \setminus \operatorname{supp}(f_1 + f_2 Y + (e))$ are *roots* and projective points outside of $\operatorname{supp}(f_3 + (e))$ are *poles* with multiplicities defined with local rings.

For each $P \in \mathbb{G}$, let $[P]$ be a distinct formal symbol. Define an abelian group structure on these symbols such that $[P] + [Q] = [P + Q]$ for each $(P, Q) \in \mathbb{G}^2$. A finite sum of these formal symbols is called a *divisor* of $\mathcal{E}$ over K. Each $\nu \in \mathbb{Z}^{\mathbb{G}}$ implies the existence of a divisor $[\nu] = \sum_{P \in \mathbb{G}} \nu(P)[P]$. Let $\operatorname{Div}_K(\mathcal{E})$ be the set of all divisors of $\mathcal{E}$ over K, $\operatorname{supp}([\nu]) = \operatorname{supp}(\nu)$, $\deg([\nu]) = \sum_{P \in \mathbb{G}} \nu(P)$, and $\operatorname{Div}^0_K(\mathcal{E}) \subseteq \operatorname{Div}_K(\mathcal{E})$ the degree-0 divisors. Each $f \in K(\mathcal{E})^\times$ has a corresponding divisor $[\nu]$ where ν describes the multiplicity of the roots and poles of f, encoding the multiplicity of non-identity P with $\nu(P) > 0$ when P is a root, $\nu(P) < 0$ when P is a pole, $\nu(O) = -\sum_{P \in \mathbb{G}} \nu(P)$, and $\nu(P) = 0$ otherwise. If $f \in K(\mathcal{E})^\times$ with root-multiplicity pairs $(P_1, m_1), \ldots, (P_n, m_n)$ and pole-multiplicity pairs $(Q_1, r_1), \ldots, (Q_t, r_t)$ with each $m_i, r_j \in \mathbb{N}$ and distinct $P_i, Q_j \in \mathbb{G} \setminus \{O\}$, then we can map $f \mapsto \sum_{i=1}^n m_i[P_i] - \sum_{j=1}^t r_j[Q_j] - (\sum_{i=1}^n m_i - \sum_{j=1}^t r_j)[O] \in \operatorname{Div}^0_K(\mathcal{E})$. The divisors in the image are called *principal divisors*. Every principal divisor $[\nu]$ has $\sum_{P \in \mathbb{G}} \nu(P) P = O \in \mathbb{G}$. Let $\operatorname{Prin}_K(\mathcal{E})$ be the principal divisors, and define the surjective group homomorphism $\operatorname{div} : K(\mathcal{E})^\times \to \operatorname{Prin}_K(\mathcal{E})$ via this map. Then div provides the following relations.

$$\mathfrak{R} = \big\{(f, [\nu]) \mid f \in K(\mathcal{E})^\times, [\nu] \in \operatorname{Prin}_K(\mathcal{E}), \operatorname{div}(f) = [\nu]\big\} \tag{1.1}$$

$$\mathfrak{R}' = \{(f, [\nu]) \mid 3 \leq |\operatorname{supp}([\nu])| < p\} \subseteq \mathfrak{R} \tag{1.2}$$

2 Method

Suppose a prover and verifier use a scheme built in an elliptic curve group setting. Verification requires satisfying one or more linear constraints coming from the group law of $\mathbb{G}$, in the sense that a valid proof includes non-identity group elements $P_1, \ldots, P_n \in \mathbb{G}$ and integer weights $m_1, \ldots, m_n \in \mathbb{Z}$ such that $\sum_{i=1}^n m_i P_i = O$. Every principal divisor has degree 0 and corresponds to a nontrivial decomposition of O, so it is sufficient to check for the existence of some $f \in K(\mathcal{E})^\times$ such that $\sum_i m_i[P_i] - (\sum_i m_i)[O] = \operatorname{div}(f)$ as a witness of principality.

To check the witness, Weil's reciprocity provides an equality of rational functions, $\widehat{f}(\operatorname{div}(g)) - \widehat{g}(\operatorname{div}(f)) = 0$, for every pair $f, g \in K(\mathcal{E})^\times$ such that the non-identity roots and poles of f are disjoint from those of g. For a monic fixed f and variable g, the rational function has the coefficients of g and the x- and y-coordinates of the points of $\operatorname{div}(g)$ as its variables. Clearing denominators yields a verification polynomial, and reciprocity states this is a decomposition of the zero polynomial. If a random $(g, \operatorname{div}(g))$ is sampled from a large enough set and is found to be a root of the verification polynomial, then the Schwartz-Zippel Lemma implies that the verification polynomial is the zero polynomial exactly, except with a small probability (see Section A.8). We show that checking the

witness f against a random line $g = \ell$ is sufficient in Lemma 2.3.3.

2.1 Previous Descriptions

Eagen sampled $g = \ell$ using two randomly sampled points (we use P, Q here, whereas Eagen used A_0, A_1) and interpolating. Eagen's derivation of the verification equation writes derivatives with respect to the interpolating slope λ, but this is not valid without taking greater care, as follows.

A group law statement for $\mathbb{G}$, say $P + Q + R = O$, induces a system of 9 equations with 6 unknowns (see Equation A.2 in Appendix A.4.1), which influences many computations with derivations. If λ and μ are handled as independent variables, four more variables must be determined in order to solve the system (among $x_P, x_Q, x_R, y_P, y_Q, y_R$). This forces at least once choice of x_T for $T \in \{P, Q, R\}$ to be handled as a dependent variable, totally determined by the other variables. While solutions for x_T in terms of λ and μ are obtainable in the algebraic closure $\overline{K}$ using Cardano's formula, the solutions are not guaranteed to be rational functions. This x_T is (in general) not rational in λ, and writing $\frac{dx_T}{d\lambda}$ is not valid without further justification. On the other hand, solutions are rational in the variables x_P, y_P, x_Q, y_Q, side-stepping the issue.

2.2 Method Description

Assume $H : \{0,1\}^* \to \mathbb{G} \times \mathbb{G}$ is a cryptographic hash function and `Wit` is a function which[1] inputs an ordered list of group elements $P_1, \ldots, P_n$ and outputs a function field element $f = \frac{f_1(X)+Yf_2(X)+(e)}{1+(e)} \in K(\mathcal{E})$ such that $\mathrm{div}(f) = \sum_{i=1}^n [P_i]$ whenever $\sum_i [P_i]$ is a principal divisor and a distinct failure symbol $\perp$ otherwise. Define the tuple of algorithms $\Pi = (\texttt{Setup}, \texttt{Prove}, \texttt{Verify})$ as follows.

- $\texttt{Setup}(\lambda) \to \texttt{params}$. Input security parameter and output setup parameters `params`.
- $\texttt{Prove}(\texttt{params}, [\nu]) \to \pi$. The prover does the following:
  1. Parse $\sum_{i=1}^n m_i[P_i] \leftarrow [\nu]$.
  2. If $n < 3$ or $n \geq p$ or $\sum_{i=1}^n m_i P_i \neq O$, output $\perp$ and terminate.
  3. Compute `out` using Algorithm 4:[2]

$$\texttt{out} \leftarrow \texttt{Wit}(\texttt{params}, \underbrace{P_0, \ldots, P_0}_{\nu(P_0) \text{ times}}, \underbrace{P_1, \ldots, P_1}_{\nu(P_1) \text{ times}}, \cdots, \underbrace{P_{n-1}, \ldots, P_{n-1}}_{\nu(P_{n-1}) \text{ times}}).$$

  4. Compute $(P, Q) \leftarrow H(\texttt{params}, f, [\nu])$. If $P \in \{O\} \cup \{P_i\}_{i=1}^n$, output $\perp$ and terminate.
  5. If $Q \in \{O\} \cup \{P_i\}_{i=1}^n \cup \left\{P, -P, -2P, -\frac{1}{2}P\right\} \cup \{-(P + P_i)\}_{i=1}^n$, output $\perp$ and terminate.

[1] More generally, `Wit` may be constructed from any implementation of an right-inverse map div^{-1} for the surjective group homomorphism div. Every such function can be wrapped with an algorithm which inputs an ordered list of group elements $P_1, \ldots, P_n$ and outputs a distinct failure symbol $\perp$ or a function field element f such that f has a monic numerator and $div(f) = \sum_i [P_i]$, and so will satisfy the assumptions of Section 2.2.

[2] This step does not require an abort condition, as if $\texttt{out} = \perp$, then the previous step would have terminated.

6. Compute $\Delta X = x_Q - x_P$, $\Delta Y = y_Q - y_P$, $\lambda = \Delta Y / \Delta X$, and $\mu = y_P - \lambda x_P$.
7. Compute $R = -(P + Q)$ and $\Gamma_T = 3x_T^2 + \alpha$ for $T \in \{P, Q\}$.
8. Compute the following:

$$\begin{aligned} A_i =& \Delta X \cdot (y_i \Delta X - x_i \Delta Y + y_P \Delta X - x_P \Delta Y)^{-1} \text{ for each } 1 \leq i \leq n \\ D =& y_R f(P) f(Q) f(R) (\Gamma_P \Delta X - 2 y_P \Delta Y)(\Gamma_Q \Delta X - 2 y_Q \Delta Y) \end{aligned}$$

9. Compute D^{-1} and output $\pi = (f, A_1, \ldots, A_n, D^{-1})$.

- $\texttt{Verify}(\texttt{params}, [\nu], \pi) \to b \in \{0, 1\}$.

1. Parse $(f, A_1, \ldots, A_n, D^{-1}) \leftarrow \pi$ if possible; otherwise, output 0 and terminate.
2. Execute steps 1, 2, and 4 through 7 from `Prove`; if any of those steps terminate with a failure symbol $\perp$, then output 0 and terminate.
3. Compute the following:

$$\begin{aligned} f_X =& \frac{\partial f}{\partial X} = \frac{f_1'(X) + Y f_2'(X) + (e)}{1 + (e)} \\ f_Y =& \frac{\partial f}{\partial Y} = \frac{f_2(X) + (e)}{1 + (e)} \\ \Gamma_R =& 3x_R^2 + \alpha \\ D =& y_R f(P) f(Q) f(R) (\Gamma_P \Delta X - 2 y_P \Delta Y)(\Gamma_Q \Delta X - 2 y_Q \Delta Y) \\ N_1 =& 2 y_R \Delta X f(P) f(Q) f_X(R) (4 y_P y_Q \Delta Y - y_P \Gamma_Q \Delta X - y_Q \Gamma_P \Delta X) \\ N_2 =& \Gamma_R \Delta X f(P) f(Q) f_Y(R) (4 y_P y_Q \Delta Y - y_P \Gamma_Q \Delta X - y_Q \Gamma_P \Delta X) \\ N_3 =& 2 y_P y_R \Delta X f(Q) f(R) f_X(P) (\Gamma_Q \Delta X - 2 y_Q \Delta Y) \\ N_4 =& y_R \Gamma_P \Delta X f(Q) f(R) f_Y(P) (\Gamma_Q \Delta X - 2 y_Q \Delta Y) \\ N_5 =& 2 y_Q y_R \Delta X f(P) f(R) f_X(Q) (\Gamma_P \Delta X - 2 y_P \Delta Y) \\ N_6 =& y_R \Gamma_Q \Delta X f(P) f(R) f_Y(Q) (\Gamma_P \Delta X - 2 y_P \Delta Y) \end{aligned}$$

4. If any of the following hold, output 0 and terminate.
   - $P = O$, or $Q = O$, or $R = O$,
   - $P = Q$, or $P = R$, or $Q = R$,
   - there exists some $P_i \in \mathrm{supp}([\nu])$ such that $P = P_i$, or $Q = P_i$, or $R = P_i$,
   - $\Delta X \neq (\Delta X \cdot y_i - \Delta Y \cdot x_i - \Delta X \cdot \mu) A_i$ for some $1 \leq i \leq n$,
   - $D \cdot D^{-1} \neq 1$,
   - $\left(\sum_{i=1}^{6} N_i \right) D^{-1} \neq \sum_i m_i A_i$.
5. Otherwise, output 1 and terminate.

From now on, we omit `params` from the inputs of functions to be concise.

Note that as $|\mathbb{G}| > 3$ and of prime order, $\mathbb{G}$ contains no elements of order 2 or 3, but every non-identity point with $y_T = 0$ is its own inverse, a contradiction. Thus, it is not necessary to check whether any points have $y_T = 0$ before performing divisions. Moreover, some abort conditions in `Prove` steps 4 or 5 is met if and only if the sampled group law statement $P + Q + R = O$ satisfies any of the following: P, Q, or R is identity (implying the interpolating line is vertical or has no roots on $\mathcal{E}$), or a collision occurs among P, Q, R (implying the interpolating line is tangent to $\mathcal{E}$), or a collision occurs between P, Q, R and $\mathrm{supp}([\nu])$ (implying Weil's reciprocity cannot be naïvely handled as we do).

2.2.1 Usage and Efficiency

Any scalar multiplication algorithm $\texttt{sc_mul}(a, G) = aG$ can be made verifiable. Indeed, to prove that a scalar multiplication has been performed correctly, say $P = aG$ for some $G \in \mathbb{G} \setminus \{O\}$ and some $1 \leq a < \mathrm{ord}(\mathbb{G})$, the prover runs `Wit` with $[\nu] = a[G] + [-P]$. However, $[\nu]$ decomposes non-uniquely this way, so another alternative is $\sum_i s_i[2^i G] + [-P]$, where $\vec{s} \pmod p$ is a (not necessarily unique) decomposition of a into powers of 2. Different representations of $[\nu]$ lead to different computational costs. Generally, using $[\nu] = a[G] + [-P]$ requires a witness of degree $O(a)$, increasing costs impractically when a is a random element of $\mathbb{Z}/q\mathbb{Z}$ for a large q, except when witnesses $\frac{f(X,Y)+(e)}{g(X,Y)+(e)}$ can be represented with (ratios of) sparse polynomials.

A proof $\pi \leftarrow \texttt{Prove}([\nu])$ for $[\nu] = \sum_{i=1}^{n} P_i$ consists of one function field element and $n + 1$ field elements. Each function field element can be described by two polynomials from $K[X]$, each of degree up to n. Since $|K| = p$, each proof takes $O((3n + 1)\lceil \log_2(p) \rceil)$ bits. We describe operation counts for proving and verifying in Table 1.

Algorithm	d/dX	$+$ in K	$\times$	$\div$	$+$ in $\mathbb{G}$	External
`Prove`	2	$3n + 7$	$10n + 6$	1	1	1 (`Wit`, H)
`Verify`	2	$5n + 22$	$15n + 51$	0	1	1 (H)

Table 1: Operation Counts for Different Algorithms

For example, using Curve25519, we have $\lceil \log_2(p) \rceil = 255$, so each proof can be described with $32(3n + 1)$ bytes; if $n = 256$, then each proof can be described with 24.61 kilobytes. Proving requires 775 field additions, 2566 field multiplications, and 1 field division. Verifying requires 1302 field additions, 3891 field multiplications, and no field divisions.

2.2.2 Issues and Risks

A potential issue was brought to us in private communication [Bas25] about whether the Schwartz-Zippel Lemma is applicable as desired when sampling slope and intercept before computing roots, as opposed to sampling the roots before interpolating to find the slope and intercept. We present statistical evidence that the two cases result in the same distribution of roots in Sage code [bgg].

The points P and Q may be selected non-interactively by utilizing the (strong) Fiat-Shamir heuristic [FS87]. This paradigm is often used to turn an interactive zero-knowledge proof into a

non-interactive signature scheme. The Fiat-Shamir transform can implemented practically with a cryptographically secure hash function with codomain $(\mathbb{G} \setminus \{O\})^2$, but care must be taken to avoid vulnerabilities with the weak Fiat-Shamir transform [NHB24, BPW16, DMWG23]. All public parameters and transcript communications must be included in the hash input. To avoid vulnerabilities like the so-called Frozen Heart attack [Jon, Mil] stemming from the weak form of the transform, the hash digest must have been generated from: commitments, witnesses, previous randomness, user tags, counters, etc.

We note that if the witness f has a multiplicative factor which is a p^{th} power, say $f = g^p \cdot h$ for some function field elements g, h, then applying a derivation provides $df = pg^{p-1}hdg + g^p dh$. Since $\mathrm{char}(K) = p$, we obtain $df = g^p dh$, which is not dependent on dg, allowing problematic malleability. In practice, p is quite large, so this issue only matters in the case when f has an impractically large degree. Thus, we restrict our security arguments to witnesses f whose corresponding divisor has $3 \leq |\mathrm{supp}(\mathrm{div}(f))| < p$, ensuring no p^{th} power multiplicative factors. Moreover, if the witness is non-zero but has a p^{th} power as a multiplicative factor, the derivative may vanish, in part or in totality. This is not a practical concern, as this condition requires the degree of f to be an integer multiple of p, which is generally quite large.

2.2.3 Extensions

Variations of the scheme can be deployed with groups of non-prime order presuming that all computations are checked to occur within the appropriate prime-order subgroup, although simply checking for small subgroup membership is not sufficient to yield a secure scheme.

Batch verification of multiple witnesses $f_1, \ldots, f_M$ for principal divisors $[\nu_1], \ldots, [\nu_M]$ may be possible by sampling coefficients $\gamma_1, \ldots, \gamma_M \in K$ and testing the witness $\prod_j f_j^{\gamma_j}$ for the divisor $\sum \gamma_j[\nu_j]$. Furthermore, this can be simplified by asking the verifier to sample a single random value $\gamma \in K$, then demonstrating the proof for $\sum_j \gamma^j[\nu_j]$; however, see Section 2.2.1, as this may be expensive if done without care.

2.3 Security Properties

A line generated from our protocol (by sampling P, Q and then interpolating $R = -(P + Q)$) may cause the protocol to fail under certain conditions.

Lemma 2.3.1. Suppose an honest prover uses the support of a principal divisor, S, with cardinality n. If $2(n + 1) < q$, then the honest prover fails[3] by outputting $\perp$ with probability $\kappa \leq \frac{3(n+3)}{q+2}\left(1 - \frac{2n+4}{3(q+1)}\right)$.

Proof. We first show the total number of such lines which do not cause the protocol to abort is at least $\frac{(q-(n+1))(q-(2n+5))}{6}$ and at most $\frac{(q-(n+1))(q-(n+2))}{6}$. We then show the total number of lines which our protocol may sample is $1 + \frac{3}{2}(q-1) + \frac{(q-1)(q-5)}{6}$. Then, the completeness error κ is at

[3] In a previous version of this paper, our argument had errors. This was communicated by Bassa [Bas25], for which we issue special thanks.

least $\frac{(q-(n+1))(q-(2n+5))}{6}$ / $1+\frac{3}{2}(q-1)+\frac{(q-1)(q-5)}{6}$ and at most $\frac{(q-(n+1))(q-(n+2))}{6}$ / $1+\frac{3}{2}(q-1)+\frac{(q-1)(q-5)}{6}$. Then, we show that these bounds imply $\kappa \leq \frac{3(n+3)}{q+2}\left(1-\frac{2n+4}{3(q+1)}\right)$.

Given $(P,Q,R) \in \mathbb{G}^3$ such that $P+Q+R=O$, we have that if P,Q,R are all non-identity, all distinct, and not in S, then certainly $P \notin \{O\} \cup S$. For this choice of P, we must have $Q \notin \{O, P, -P, -2P, -2^{-1}P\}$. Indeed, each of these cases leads to a contradiction: if $Q=O$, then Q is not non-identity; if $Q=P$, then Q is not distinct; if $Q=-P$, then R is not non-identity; if $Q=-2P$, then $R=P$; if $Q=-2^{-1}P$, then $R=Q$. We must also have $Q \neq -(P+T)$ for any $T \in S$, because otherwise $R=-(P+Q)=-(P+(-(P+T)))=T \in S$, a contradiction.

On the other hand, for any triple (P,Q,R) such that $P+Q+R=O$, $P \in \mathbb{G} \setminus (\{O\} \cup S)$, and $Q \in \mathbb{G} \setminus (\{O\} \cup S \cup \{P,-P,-2P,-2^{-1}P\} \cup \{-(P+T) \mid T \in S\})$ then the line corresponding to $P+Q+R=O$ does not cause a failure.

Hence, $P \in \mathbb{G} \setminus (\{O\} \cup S)$ without restriction, and $Q \in \mathbb{G} \setminus (\{O\} \cup S \cup \{P,-P,-2P,-2^{-1}P\} \cup \{-(P+T) \mid T \in S\}$ without restriction. However, each valid pair corresponds to a triple $P+Q+R=O$, and this triple has 6 permutations occur. Moreover, the set $\{O,P\} \cup S$ consists of $n+2$ distinct points which Q must always avoid. However, the elements $\{-P,-2P,-2^{-1}P\} \cup \{-(P+T) \mid T \in S\}$ may already be inside of S. Assuming this is the case means Q must avoid only $n+2$ elements, but if all of these points instead reside outside of S, then Q must avoid $2n+5$ elements. These are the most extreme cases, and all other cases are intermediate to these. Thus, Q must avoid between $2n+5$ and $n+2$ elements.

So, we have at least $(q-(n+1))(q-(2n+5))/6$ and at most $(q-(n+1))(q-(n+2))/6$ triples which do not cause protocol failure.

We now count all lines. Any line $P+Q+R=O$ with some non-identity element can, without loss of generality, be permuted such that $P \neq O$, and for all lines, Q is identity or not. All lines with some identity element can, without loss of generality, be permuted such that $R=O$. All lines with $P,Q,R \neq O$ with a collision of points can be permuted such that $P=Q$, and for all lines, $R=Q$ or not. Thus, all lines $P+Q+R=O$ fall into one of the following disjoint cases:

(a) $P=Q=O$

(b) $P \neq O$ and $Q=O$

(c) $P,Q \neq O$ and $R=O$

(d) $P,Q,R \neq O$ and $P=Q=R$

(e) $P,Q,R \neq O$ and $P=Q \neq R$

(f) P,Q,R non-identity and distinct.

In the first case, $P+Q+R=O$ and $P=Q=O$ implies $R=O$; we only have one triple $O+O+O=O$ of this form.

In the second case, we have $P+O+R=O$, so $R=-P$. In the third case, $P+Q+O=O$, which is just a permutation of the second case. Thus, we have one triple from these cases for every

non-identity element-inverse pair in $\mathbb{G}$, i.e. we have $(q-1)/2$ triples in the second and third cases together.

In the fourth case, $P+P+P=O$, so $3P=O$. Since $\mathbb{G}$ is prime and $q>3$, this cannot occur, and we have no lines from the fourth case.

In the fifth case, we have $P+P+R=O$, so $R=-2P$. We have one triple of this form for each non-identity group element, i.e. we have $q-1$ triples in the fifth case.

In the sixth case, we have three distinct, non-identity points. Note that for such a triple $P+Q+R=O$, we have $P\neq O$ and $Q\notin\{O,P,-P,-2P,-2^{-1}P\}$. Indeed, if $P=O$, we are in the first case, which we already counted. If $P\neq O$ and $Q=R=O$, then $P+O+O=O$, so $P=O$, and we are in the first case again. If $P\neq O$ and $Q=P$, then we have the case $P+P+R=O$, so we are in the fifth case. If $P\neq O$ and $Q=-P$, then $R=O$ and we are in the third case. If $P\neq O$ and $Q=-2P$, then $P+(-2P)+R=O$, so $R=P$, so the points are not distinct as in case five. If $P\neq O$ and $Q=-2^{-1}P$, then $P+(-2^{-1}P)+R=O$, so $R=Q$ and the points are not distinct, also as in case 5.

On the other hand, if $P\neq O$ and $Q\notin\{O,P,-P,-2P,-2^{-1}P\}$, then we must have that P,Q,R are non-identity and distinct.

Hence, $P\in\mathbb{G}\setminus\{O\}$ without restriction, and $Q\in\mathbb{G}\setminus\{O,P,-P,-2P,-2^{-1}P\}$ without restriction. However, for each such triple, we see the line corresponding to this triple appearing 6 times in total. Thus, we have $(q-1)(q-5)/6$ lines in total in the sixth case.

We summarize these data in Table 1.

Case	**Condition**	**# of triples/lines**
1	$P=Q=R=O$	1
2	$P+O+R=O \Rightarrow R=-P$	$\frac{q-1}{2}$
3	$P+Q+O=O \Rightarrow Q=-P$	$\frac{q-1}{2}$ (permutations of Case 2)
4	$P+P+P=O \Rightarrow 3P=O$	0 (since $q>3$ prime)
5	$P+P+R=O \Rightarrow R=-2P$	$q-1$
6	Distinct, non-identity P,Q,R with $P+Q+R=O$	$\frac{(q-1)(q-5)}{6}$

Table 1: Counting all possible triples (P,Q,R) with $P+Q+R=O$ in $\mathbb{G}$.

Now we compute the completeness error as the ratio of the number of lines which cause failure to the total number of lines. With at least $(q-(n+1))(q-(2n+5))/6$ and at most $(q-(n+1))(q-(n+2))/6$ triples which do not cause protocol failure, out of $1+\frac{3}{2}(q-1)+(q-1)(q-5)/6$ total lines, then under the random oracle model, we see that we have the following probability of

a successful protocol execution and completeness error.

$$\begin{aligned}
1-\kappa \geq & \frac{(q-(n+1))(q-(2n+5))/6}{1+\frac{3}{2}(q-1)+(q-1)(q-5)/6} \\
\kappa \leq & 1-\frac{(q-(n+1))(q-(2n+5))}{6+9(q-1)+(q-1)(q-5)} \\
\leq & \frac{(n+3)(3q-(2n+1))}{(q+1)(q+2)} \\
\leq & \frac{(n+3)(3(q+1)-(2n+4))}{(q+1)(q+2)} \\
\leq & \frac{(n+3)(3-\frac{2n+4}{q+1})}{q+2} \\
\leq & \frac{3(n+3)}{q+2}\left(1-\frac{2n+4}{3(q+1)}\right) \qquad (2.1)
\end{aligned}$$

□

We present code enumerating all elliptic curves up to a user-provided bit length to verify the line counts used to compute Equation 2.1 at [bgg].

Theorem 2.3.2. Π is an incomplete proving system for the relation $\mathfrak{R}'$ from Equation 1.2 with completeness error $\kappa \leq \frac{3(n+3)}{q+2}\left(1-\frac{2n+4}{3(q+1)}\right)$.

Proof. Due to Lemma 2.3.1, we only need to check correctness conditioned on the event that an honest prover does not fail. Presume that $f \in K(\mathcal{E})$, $[\nu] = \mathrm{div}(f)$, and the prover successfully honestly executes `Prove` without failure. Then the following occurs during the execution of `Prove`.

(a) The prover executes `Prove` step 3 to compute $\texttt{out} \leftarrow \texttt{Wit}(\texttt{params}, [\nu])$. Following Lemma C.0.1, $\texttt{out} = f \in K(\mathcal{E})$ with certainty, where $\perp \neq f = \frac{f_1(X)+Yf_2(X)+(e(X,Y))}{f_3(X)+(e(X,Y))}$ and $f_3(X) = 1$.

(b) The prover executes `Prove` step 4 to compute $(P, Q) \leftarrow H(\texttt{params}, f, [\nu])$, interpolates these points, and executes `Prove` step 7 to compute $R = -(P+Q)$. The constraints that $P, Q, -(P+Q) \neq O$ are exactly that none of these points are identity. The constraints that $P \neq Q$, $2P + Q \neq O$, and $P + 2Q \neq O$ are exactly that the three points are distinct. The constraints that $P \neq P_i$, $Q \neq P_i$, and $Q \neq -(P + P_i)$ are exactly that none of these points lie in the support of ν. Thus, the set $\{P, Q, R\}$ is sampled uniformly from the set of all 3-sets of collinear non-identity points on $\mathcal{E}$ which do not collide with any P_i. All such triples have an interpolating line which intersects $\mathcal{E}$ transversally at these three distinct non-identity points.

(c) Conditioned upon success until this point and an honest prover, parsing in `Prove` succeeds with certainty, and $f_3(X) = 1$, so the computations in `Prove` step 6 succeed with certainty. Since the prover has not yet failed, each $y'_T \neq \lambda$ and each $f(T) \neq 0$, so the prover succeeds at `Prove` up to step 8 with certainty, outputting π in step 9.

Then during verification, the following occurs.

(a) Since the prover was honest and did not fail, parsing succeeds with certainty in `Verify` step 1, so the verifier does not output 0 in this step.

(b) Moreover, the witness is valid $f = \frac{f_1(X)+f_2(Y)+(e)}{1+(e)} \in K(\mathcal{E})$, and the verifier obtains the pair (P, Q) in `Verify` step 2.

(c) The verifier computes $R = -(P + Q)$ and checks for a collision. Conditioned on an honest prover with a successful execution of `Prove`, the points P, Q, R do not collide with O, each other, or the points $\{P_i\}$, so the verifier does not output 0 in `Verify` step 2.

(d) Conditioned on an honest prover, a successful execution of `Prove`, and the verifier not yet terminating, then all the computations in `Verify` step 3 succeed with certainty and the verifier obtains the following.

$$\begin{aligned}
N_1 =& 2y_R\Delta X f(P)f(Q)f_X(R)(4y_Py_Q\Delta Y - y_P\Gamma_Q\Delta X - y_Q\Gamma_P\Delta X)\\
N_2 =& \Gamma_R\Delta X f(P)f(Q)f_Y(R)(4y_Py_Q\Delta Y - y_P\Gamma_Q\Delta X - y_Q\Gamma_P\Delta X)\\
N_3 =& 2y_Py_R\Delta X f(Q)f(R)f_X(P)(\Gamma_Q\Delta X - 2y_Q\Delta Y)\\
N_4 =& y_R\Gamma_P\Delta X f(Q)f(R)f_Y(P)(\Gamma_Q\Delta X - 2y_Q\Delta Y)\\
N_5 =& 2y_Qy_R\Delta X f(P)f(R)f_X(Q)(\Gamma_P\Delta X - 2y_P\Delta Y)\\
N_6 =& y_R\Gamma_Q\Delta X f(P)f(R)f_Y(Q)(\Gamma_P\Delta X - 2y_P\Delta Y)\\
D =& y_R f(P)f(Q)f(R)(\Gamma_P\Delta X - 2y_P\Delta Y)(\Gamma_Q\Delta X - 2y_Q\Delta Y)
\end{aligned}$$

(e) Conditioned on an honest prover, a successful execution of `Prove`, and the verifier not yet terminating, all the following are satisfied with certainty.

$$\begin{aligned}
A_i =& (y_i - \lambda x_i - \mu)^{-1} \text{ for each } i\\
N_1 =& 2y_R\Delta X f(P)f(Q)f_X(R)(4y_Py_Q\Delta Y - y_P\Gamma_Q\Delta X - y_Q\Gamma_P\Delta X)\\
N_2 =& \Gamma_R\Delta X f(P)f(Q)f_Y(R)(4y_Py_Q\Delta Y - y_P\Gamma_Q\Delta X - y_Q\Gamma_P\Delta X)\\
N_3 =& 2y_Py_R\Delta X f(Q)f(R)f_X(P)(\Gamma_Q\Delta X - 2y_Q\Delta Y)\\
N_4 =& y_R\Gamma_P\Delta X f(Q)f(R)f_Y(P)(\Gamma_Q\Delta X - 2y_Q\Delta Y)\\
N_5 =& 2y_Qy_R\Delta X f(P)f(R)f_X(Q)(\Gamma_P\Delta X - 2y_P\Delta Y)\\
N_6 =& y_R\Gamma_Q\Delta X f(P)f(R)f_Y(Q)(\Gamma_P\Delta X - 2y_P\Delta Y)\\
D =& y_R f(P)f(Q)f(R)(\Gamma_P\Delta X - 2y_P\Delta Y)(\Gamma_Q\Delta X - 2y_Q\Delta Y)
\end{aligned}$$

As the prover was honest and successful, all these computations are correctly computed, and the verifier sees the following.

- $(y_i - \lambda x_i - \mu)A_i = 1$ for each $1 \le i \le n$
- $2y_T\Delta Y \neq \Gamma_T \cdot \Delta X$ for any $T \in \{P, Q, R\}$

– $\Delta X \neq (\Delta X \cdot y_i - \Delta Y \cdot x_i - \Delta X \cdot \mu)A_i$ for some $1 \leq i \leq n$
– $D \cdot D^{-1} \neq 1$
– $\left(\sum_{i=1}^{6} N_i\right) D^{-1} \neq \sum_i m_i A_i$

So, the verifier does not output 0 in `Verify` step 4.

(f) In Equation B.8 of the Appendix, we show $\delta(\widehat{f}(X_P, Y_P, X_Q, Y_Q)) \mid_{P,Q} = \left(\sum_{i=1}^{6} N_i\right) \cdot D^{-1}$. We also have $\widehat{f}(x_P, y_P, x_Q, y_Q) = f(P)f(Q)f(R)$, and by Weil reciprocity, $f(P)f(Q)f(R) = \prod_{i=1}^{n}(y_i - \lambda x_i - \mu)^{m_i}$, where $Y - \lambda X - \mu$ is the line interpolating P, Q, and R. Hence, $\delta(\widehat{f}) \mid_{P,Q} = \sum_{i=1}^{n} \frac{m_i}{y_i - \lambda x_i - \mu}$. The verifier checked above that $(y_i - \lambda x_i - \mu)A_i = 1$, so $\delta(\widehat{f}) \mid_{P,Q} = \sum_i m_i A_i$, implying verification is passed; the verifier outputs 1 with certainty.

By inspection, it is clear that the only case of failure is in the case of a bad sample of (P, Q), namely the completeness error from Lemma 2.3.1. □

We note that the scheme can be completed with respect to $\mathcal{R}'_{\text{div}}$ from Equation 1.2 by allowing provers to include random context strings $\texttt{ctx} \in \{0,1\}^*$ in their proofs for use as prefixes for the hash computing P, Q, resampling `ctx` until `Prove` succeeds. Moreover, the scheme can be extended to all of $\mathcal{R}_{\texttt{div}}$ from Equation 1.1 by modifying `Wit` to accept divisors with $|\text{supp}(\nu)| < 3$ as input, covering all of $\mathcal{R}_{\texttt{div}}$.

However, this completeness error is already negligible in the security parameter governing $|\mathbb{G}| = q$, so completing the scheme with respect to $\mathcal{R}'_{\text{div}}$ is not practically necessary in the case that q is even moderately large.

Lemma 2.3.3. Let $\nu : \mathcal{E} \to \mathbb{Z}$ such that $|\text{supp}(\nu)| < \infty$, and let $f \in K(\mathcal{E})^\times$. If, for every $\ell \in K(\mathcal{E})$ with $\deg(\ell) = 1$ and whose roots are disjoint from the roots and poles of f, $\widehat{f}(\text{div}(\ell)) = \widehat{\ell}([\nu])$, then $[\nu] = \text{div}(f)$.

Proof. Define $\delta : \mathcal{E} \to \mathbb{Z}$ such that $\delta(T) = \nu(T) - \text{ord}_T(f)$ for every $T \in \mathcal{E}$. Then $\text{div}(f) = [\nu] + [\delta] \in \text{Prin}(\mathcal{E})$. Thus, it is sufficient to show that $\delta(T) = 0$ for every $T \in \mathcal{E}$ (or, equivalently, $[\delta] = [O]$).

In pursuit of contradiction, assume some $P_0 \in \text{supp}(\delta)$. By Weil reciprocity, $\widehat{f}(\text{div}(\ell)) = \widehat{\ell}(\text{div}(f)) = \widehat{\ell}([\nu] + [\delta])$. By Theorem A.6.1, $\widehat{\ell}([\nu] + [\delta]) = \widehat{\ell}([\nu]) \cdot \widehat{\ell}([\delta])$. By assumption, $\widehat{f}(\text{div}(\ell)) = \widehat{\ell}([\nu])$. Hence, $\widehat{\ell}([\nu]) = \widehat{\ell}([\nu])\widehat{\ell}([\delta])$, so $1 = \widehat{\ell}([\delta])$. However, this must hold for any line ℓ, a contradiction unless $[\delta] = [0]$. □

Theorem 2.3.4. Π has soundness error $\frac{n}{2|K|(|K|-1)}$ for the relation from Equation 1.2.

Proof. We show that if $f \in K(\mathcal{E})$ has no p^{th}-power factors, $\nu : \mathbb{G} \setminus \{O\} \to \mathbb{Z}$ has $|\text{supp}(\nu)| = n \geq 3$, and $\pi \in \frac{K[X,Y]}{(e)} \times K^{n+1}$, say $\pi = (f, A_1, \ldots, A_n, D^{-1})$, is any proof such that $\texttt{Verify}([\nu], \pi) = 1$, then $[\nu] = \text{div}(f)$ with probability at least $1 - \frac{n}{2|K|(|K|-1)}$.

Assume $\texttt{Verify}([\nu], \pi) = 1$. Then $\pi = (f, A_1, \ldots, A_n, N_1, N_2, N_3, N_4, N_5, N_6, D^{-1})$ for some $f \in K(\mathcal{E})$ and some $A_1, \ldots, A_n, N_1, N_2, N_3, N_4, N_5, N_6, D^{-1} \in K$, otherwise the verifier would output 0 in the first step, a contradiction. Moreover, the verifier computes challenge points with

the hash function H, say $(P, Q) \leftarrow H(\texttt{params}, f, [\nu])$, such that $P \neq \pm Q$; indeed, otherwise the verifier would output 0 in the second step, a contradiction. The verifier computes $R = -(P + Q)$, and finds that $P \neq \pm R$ and $Q \neq \pm R$ and $\{P, Q, R\} \cap \{P_i\}_{i=1}^{n} = \emptyset$, otherwise the verifier would output 0 in the third step, a contradiction. Thus, the verifier successfully computes ΔX, ΔY, λ, μ, D, and N_i for $i = 1, \ldots, 6$, yielding the following

$$\begin{aligned}
\Delta X =& x_Q - x_P, \quad \Delta Y = y_Q - y_P, \quad \lambda = \frac{\Delta Y}{\Delta X}, \quad \mu = y_P - \lambda x_P \\
N_1 =& 2y_R \Delta X f(P) f(Q) f_X(R)(4y_P y_Q \Delta Y - y_P \Gamma_Q \Delta X - y_Q \Gamma_P \Delta X) \\
N_2 =& \Gamma_R \Delta X f(P) f(Q) f_Y(R)(4y_P y_Q \Delta Y - y_P \Gamma_Q \Delta X - y_Q \Gamma_P \Delta X) \\
N_3 =& 2y_P y_R \Delta X f(Q) f(R) f_X(P)(\Gamma_Q \Delta X - 2y_Q \Delta Y) \\
N_4 =& y_R \Gamma_P \Delta X f(Q) f(R) f_Y(P)(\Gamma_Q \Delta X - 2y_Q \Delta Y) \\
N_5 =& 2y_Q y_R \Delta X f(P) f(R) f_X(Q)(\Gamma_P \Delta X - 2y_P \Delta Y) \\
N_6 =& y_R \Gamma_Q \Delta X f(P) f(R) f_Y(Q)(\Gamma_P \Delta X - 2y_P \Delta Y) \\
D =& y_R f(P) f(Q) f(R)(\Gamma_P \Delta X - 2y_P \Delta Y)(\Gamma_Q \Delta X - 2y_Q \Delta Y)
\end{aligned}$$

At this point, the verifier finds:

- $(y_i - \lambda x_i - \mu) A_i = 1$ for each $1 \leq i \leq n$
- $2y_T \Delta Y \neq \Gamma_T \cdot \Delta X$ for any $T \in \{P, Q, R\}$
- $\Delta X \neq (\Delta X \cdot y_i - \Delta Y \cdot x_i - \Delta X \cdot \mu) A_i$ for some $1 \leq i \leq n$
- $D \cdot D^{-1} \neq 1$
- $\left(\sum_{i=1}^{6} N_i\right) D^{-1} \neq \sum_i m_i A_i$

Hence, the verifier is assured that the following is satisfied with certainty.

$$\left(\sum_{i=1}^{6} N_i\right) \cdot D^{-1} = \sum_i \frac{m_i}{y_i - \lambda x_i - \mu}$$

Moreover, since the verifier verified the numerators and denominators in this equation, the verifier concludes the verification equations from Section B.1 are satisfied. In particular, Equation B.8 is satisfied. Hence, the following holds.

$$\delta\big(\widehat{f}(X_P, Y_P, X_Q, Y_Q)\big) \,\big|_{(X_P, Y_P, X_Q, Y_Q) = (x_P, y_P, x_Q, y_Q)} = \sum_i \frac{m_i}{y_i - \lambda x_i \mu}$$

Of course, the verifier also knows that, for the line ℓ interpolating P and Q, the equality $\sum_i \frac{m_i}{y_i - \lambda x_i \mu} = \sum_i \delta\big(\ell(X_i, Y_i)\big) \,\big|_{(X_i, Y_i) = (x_i, y_i)}$ holds, where this time δ is the logarithmic derivative

on $K(\mathcal{E}^n)$. As before, this δ is a homomorphism, so the verifier concludes the following equality.

$$\delta\big(\widehat{f}(X_P, Y_P, X_Q, Y_Q)\big)\Big|_{(X_P,Y_P,X_Q,Y_Q)=(x_P,y_P,x_Q,y_Q)} = \delta\left(\prod_{i=1}^{n} \ell(X_i, Y_i)^{m_i}\right)\Bigg|_{\{(X_i,Y_i)\}_i=\{(x_i,y_i)\}_i} \tag{2.2}$$

The verifier recalls that, for a fixed f (and therefore fixed (P_i, m_i) pairs), both $f(P)f(Q)f(R)$ and $\prod_i \ell(X_i, Y_i)^{m_i}$ are rational functions in the variables (X_P, Y_P, X_Q, Y_Q), so (x_P, y_P, x_Q, y_Q) is a root of $f(P)f(Q)f(R) - \prod_i \ell(X_i, Y_i)^{m_i}$.

The Schwartz-Zippel lemma states that, if $g \in K[Z_1, \ldots, Z_n]$, $S \subseteq K^n$ is a finite set, $\vec{z} \xleftarrow{\$} S$ is sampled uniformly at random, and $g(\vec{z}) = 0$, then the probability that $g \neq 0$ in $K[\{Z_i\}_i]$ is at most $\frac{n}{|S|}$. This extends to rational functions by clearing denominators.

In this case, the polynomial ring is the coordinate ring of the product variety $\mathcal{E}^2$, namely $K[X_P, Y_P, X_Q, Y_Q]/(e(X_P, Y_P), e(X_Q, Y_Q))$, the sampled evaluation quadruple is (x_P, y_P, x_Q, y_Q), and the rational function is $f(P)f(Q)f(R) - \prod_i \ell(X_i, Y_i)^{m_i}$. The hash function is used to sample (P, Q). The sampled x-coordinates $x_P \neq x_Q$ without regard to order comes from a set with at most $\frac{1}{2}|K|(|K| - 1)$ elements. Given a pair $(x_1, x_2) \in K^2$ such that there exists any $(y_1, y_2) \in K^2$ such that both (x_1, y_1) and (x_2, y_2) are on $\mathcal{E}$, then there are at most two choices of y_1 which work with x_1, and at most two choices of y_2 which work with x_2. Thus, $|S| = 4 \cdot \left(\frac{1}{2}|K|(|K| - 1)\right) = 2p(p-1)$. Thus, the probability that $f(P)f(Q)f(R) - \prod_i \ell(X_i, Y_i)^{m_i} \neq 0$ is at most $\frac{n}{2|K|(|K|-1)} = \frac{n}{2p(p-1)}$. Now the conclusion of the proof is simply a corollary of Lemma 2.3.3. □

2.4 Computational Refinements

We describe ways to improve the protocol above.

2.4.1 Seed Trees

A standard improvement utilizes seed trees to effectively communicate the randomness used to generate the verifier's challenge points. Suppose that a prover has N different proofs, where the randomness $\texttt{seed}_i$ was used along with a CSPRNG to generate P_i and Q_i for $i = 1, \ldots, N$. The first improvement is obvious: instead of communicating $\{(P_i, Q_i)\}_{i=1}^N$, the prover can communicate $\{\texttt{seed}_i\}_{i=1}^N$, then the verifier may compute the points themselves from the CSPRNG. As a second improvement, one can go even farther: instead of the prover communicating each $\texttt{seed}_i$ to the verifier separately, they can utilize seed trees, wherein they place a master seed `master_seed` at the root of a Merkle tree, which is then used to generate each seed $\texttt{seed}_i$ at the leaves. In this manner, the prover can communicate the randomness used in their proofs in a way that is logarithmic in N, the total number of proofs. The benefit that this approach provides is most apparent when utilized non-interactively, as it permits for more compact proof sizes.

2.4.2 Interpolation

In this section, we compare the Lagrange method of interpolation against the barycentric method of interpolation for witness construction.

We are given n distinct interpolation node-value pairs $\{(x_i, f_i)\}_{i=1}^n$, where there exists some function f such that $f_i = f(x_i)$ for each $1 \leq i \leq n$. To interpolate these data is to find an interpolating polynomial p, called an *interpolant*, satisfying $p(x_i) = f_i$ for each $1 \leq i \leq n$. There exists an interpolant p such that $\deg p \leq n-1$, but interpolants can be of arbitrarily high degree. Although interpolants are not unique in general, interpolants are unique up to this degree bound. Indeed, if $q(x) \neq p(x)$ is another interpolating polynomial that also has $\deg(q) \leq n-1$, then $p(x) - q(x)$ is a polynomial with degree at most $n-1$ with n roots, namely $x = x_i$ for $1 \leq i \leq n$. Thus, $p(x) - q(x)$ is exactly the zero polynomial[4].

Lagrange's method is ubiquitous in theoretical descriptions, due to its simplicity, but it is sub-optimal and often not the preferred practical choice [BT]. Other options are available like Newton's, though here we focus on barycentric. Barycentric interpolation yields efficiently-evaluated representations of interpolants, where interpolation weights can be efficiently recomputed when nodes are updated, and offer a wide range of other numerical benefits [BT]. Numerical evidence is available showing barycentric benefits in practice [Var], [Mon25, fab]. Lagrange's method uses interpolant $p(x) = \sum_i f_i \ell_i(x)$ with the basis functions

$$\ell_i(x) = \prod_{i \neq j} \frac{x - x_j}{x_i - x_j}$$

which satisfy $\ell_i(x_j) = \delta(i,j)$ for each $1 \leq i, j \leq n$, where $\delta(i,j)$ is the Kronecker delta function, evaluating to 1 when $i = j$ and 0 else. This interpolant is a polynomial in $K[X]$ satisfying $p(x_j) = f_j$ for each $1 \leq j \leq n$, and has $\deg p \leq n-1$. A Lagrange interpolant $p(x)$ naïvely requires $O(n^2)$ arithmetic operations to compute the coefficients of an interpolant, and interpolants take $O(n^2)$ operations to evaluate at an arbitrary point. A new Lagrange interpolant must be computed to include new interpolation nodes x_{n+1} or to change node values, $x_i \to x_i'$.

For Barycentric interpolation, note that the numerator of $\ell_i(x)$ can be written

$$\frac{\prod_{j=1}^n x - x_j}{x - x_i}.$$

The denominator of ℓ_i is $\prod_{i \neq j} x_i - x_j$. Let $w_i = \left(\prod_{i \neq j} x_i - x_j\right)^{-1}$ and $\ell(x) = \prod_i x - x_i$. Then

[4]From another standpoint, distinct nodes yield Vandermonde matrices with non-zero determinate, implying the Vandermonde matrix is invertible. In turn, solutions are unique.

$$
\begin{aligned}
p(x) =& \sum_{i=1}^{n} f_i \ell_i(x) \\
=& \sum_{i=1}^{n} f_i \cdot \frac{\left(\prod_{j=1}^{n} x - x_j\right)}{x - x_i} \cdot \left(\prod_{i \neq j} x_i - x_j\right)^{-1} \\
=& \sum_{i=1}^{n} f_i \cdot \frac{\left(\prod_{j=1}^{n} x - x_j\right)}{x - x_i} w_i \\
=& \sum_{i=1}^{n} f_i \cdot \ell(x) \cdot \frac{w_i}{x - x_i} \\
=& \ell(x) \sum_{i} \frac{f_i w_i}{x - x_i}
\end{aligned}
$$

That is to say, we use the barycentric basis functions $\left\{\frac{w_i \ell(x)}{x - x_i}\right\}_{i=1}^{n}$. Equivalently,

$$
p(x) = \frac{\sum_{i=1}^{n} \frac{w_i}{x - x_i} f_i}{\sum_{i=1}^{n} \frac{w_i}{x - x_i}}. \tag{2.3}
$$

Evaluating $p(x)$ at any point costs $O(n)$ divisions, multiplications, additions. Updating an interpolant by introducing a new node x_{n+1} requires only computing new weights w_i^* by setting $w_i^* = \frac{w_i}{x_i - x_{n+1}}$ for each $1 \leq i \leq n$ and $w_{n+1}^* = \left(\prod_{i \neq n+1} x_i - x_{n+1}\right)^{-1}$. Recursive updating algorithms are possible [BT]. Further, arbitrary choices of w_i in Equation 2.3 instead of using $w_i = \left(\prod_{i \neq j} x_i - x_j\right)^{-1}$ provides a rational (not necessarily polynomial) interpolant

2.4.3 Fast Polynomial Multiplication

If the scheme only needs to handle divisors with some uniform bound on support cardinalities (i.e. only divisors $[\nu]$ with $|\mathrm{supp}(\nu)| \leq N$ for a fixed $N \in \texttt{params}$), then all witnesses will have bounded degrees. In this case, a careful choice of $\mathrm{char}(K) = p$ may allow elliptic curve fast Fourier transform (ECFFT) speedups to polynomial multiplication in $K[X, Y]$. In addition, for certain special fields, ECFFT methods can interpolate in $O(n \cdot \log(n))$, though for a general field it has a marginally worse runtime of $O(n \cdot \log(n)^2)$ [fab], hence we still recommend the barycentric method for interpolation.

3 Applications

In this section, we highlight two practical use cases. Any time a public key is computed from a private key, or a Pedersen commitment is opened, or a message is signed with a Schnorr signature, SLVer Bullet provides verifiability. By utilizing divisors, these applications benefit from a

marked computational improvement to the verifier's checks, but moreover they also enjoy the zero-knowledge property from their underlying zero-knowledge proof, which cannot be said of every protocol utilizing divisors from the literature.

In this next section, we use $\star$ for the entrywise vector product, a.k.a. the Hadamard or Schur product. We define $\vec{y} = (1, y, \ldots, y^{n-1})$ as the Vandermonde vector for a non-zero field element $y \in K^\times$, which permits the verifier to generate a test vector from a single field element that ensures the validity of the prover's computations. We refer to [MS25] for a proof that distinct values of y lead to linearly independent test vectors $\vec{y}$, so it is sufficient to query the prover for distinct values. Finally, for a cyclic group $\mathbb{G}$, we store group generators in vectors, namely $G = (G_1, \ldots, G_n)$ where $G_i \in \mathbb{G}$ for each i. Utilizing additive notation then, for $a \in K$, we write aG for the weighted sum $\sum_{i=1}^{n} a_i G_i$. This collapses n elliptic curve multiplications into a single point, so knowledge of multiple discrete logarithms may be demonstrated concisely.

3.1 Schnorr Identification Scheme

To begin, we showcase the classic Schnorr protocol from [Sch91], which is a zero-knowledge proof system that is complete, sound, and zero-knowledge. In Algorithm 1 we describe how the Schnorr identification protocol is modified by the divisor approach presented above. The incorporation of divisors alters the soundness and completeness error negligibly, but accelerates the verifier's final check, and hence reduces the proof size. While it is unfounded to expect that divisors by themselves to attain zero-knowledge, the Schnorr protocol enjoys this feature, and hence this protocol represents an efficient zero-knowledge proof system.

Improved Verification Schnorr Scheme		
Public: G, $P = a \cdot G$, $B_i = 2^i G$ for $i = 0, \ldots, k$		
Private: $s = (s_0, \ldots, s_k) \in K^{k+1}$ such that $a = s_0 + s_1 2 + \cdots + s_k 2^k$		
Prover		**Verifier**
Sample $r_i \xleftarrow{\$} K$ for $i = 0, \ldots, k$		
Set $\rho = r_0 + r_1 2 + \cdots + r_k 2^k$		
Set $R = \rho G$	$\xrightarrow{R}$	
	$\xleftarrow{c}$	Sample $c \xleftarrow{\$} K^\times$
Form $z = s + cr = (z_0, \ldots, z_k)$		
Set $\nu = \sum_{i=0}^{k} [z_i B_i] + [-P] + [-cR]$		
$\pi \leftarrow \texttt{Prove}(\nu)$	$\xrightarrow{\pi}$	
		Recompute ν
		Check: $\texttt{Verify}(\nu, \pi) = 1$

Algorithm 1: A computational improvement to the verification step in the Schnorr protocol.

In Algorithm 1, we present an improvement upon the traditional Schnorr scheme interactive protocol, leveraging the divisors framework. Utilizing divisors, the prover can replace the traditional last step communicating z alone with the exhibited communication of the proof π. Then the

verifier's check in the above portion - which would require $k+2$ costly elliptic curve point multiplications - can be replaced with an efficiently computable divisor sum. So with a marginal increase in communication, the verifier can offload the bulk of their computation to the prover. This is exactly the case that cryptocurrencies are concerned with, because a transaction proof only needs to be generated once, but has to be verified many times - hence offloading the verifiers' computations onto the prover leads to reduced costs overall.

Theorem 3.1.1. Algorithm 1 is a zero-knowledge proof protocol demonstrating knowledge of a discrete logarithm.

Proof. Because the Schnorr scheme is zero-knowledge, this desirable property is carried over in our divisors approach. More precisely, if any party can extract information about the secret from the divisors, then they could have in the original Schnorr scheme as well. The completeness and soundness error change due to the arguments presented in Section 2.3 (specifically Theorems 2.3.2 and 2.3.4), but the change is negligible in $|\mathbb{G}|$. □

The result is a 3-pass interactive protocol that preserves the zero-knowledge property of the Schnorr scheme (importantly, the divisor approach presented in [Bas] is not zero-knowledge) while drastically reducing the verifier's computations, with a minimal hit to communication. After applying the Fiat-Shamir heuristic to transform this zero-knowledge proof protocol into a signature scheme, where we reiterate that all the public parameters must be included in the hash input, the proof's verification time reduce heavily at the cost of an increase in the proof size.

This foundation leads to more generic zero-knowledge proof systems in which the verifier may offload computation by avoiding expensive elliptic curve computations, speeding up verification time. We note that the above approach can be applied to the Chaum-Pedersen [CP93] or Okamoto [Oka93] schemes as well, with minimal changes. As such, this particularly pertains to cryptocurrencies like Lelantus Spark [JF21] or Salvium [Pro], or indeed, FCMP++ in Monero [Parb]. Divisors are of marked interest for cryptocurrencies, because transactions must be repeatedly verifiable over the life of the blockchain - a single proof gets verified a great many times, so it behooves the protocol designers to offload verifier computations onto the prover, who must only generate the proof once. Below, we highlight how this divisors framework can be used to improve the verification time for the Bulletproofs protocol [BBB$^+$17] specifically, which is implemented in a wide range of cryptocurrency settings. In future work, we hope to showcase how, through MuSig [Tar] and linkable ring signature [LW05] techniques, this can be optimized further such that the prover only needs to send a single batch proof.

3.2 Bulletproofs

These divisor notions can also be applied to the verification procedure in the range proof of Bulletproofs [BBB$^+$17]. Bulletproofs offers an astoundingly efficient way to demonstrate that a committed value $v \in K$, taken in the original paper to be $K = \mathbb{Z}_p$ with p large enough compared to n,

is constrained to some interval $[0,\, 2^n - 1]$. Specifically, it provides a proof of the following relation:

$$\mathfrak{R}_R = \{n \in \mathbb{N}, g, h \in \mathbb{G};\ v, \gamma \in K \mid V = vg + \gamma h \text{ and } v \in [0,\, 2^n - 1]\}.$$

The construction is quite clever; we recreate it below for completeness.

The main idea is to utilize the binary representation of a vector, were $a_L \in \{0,\, 1\}^n$ represents the bits of v in binary, so that $\langle a_L,\, \vec{2}\rangle = v$. If the prover can produce a certificate that a_L is binary, and genuinely represents the binary bits of v, then the verifier can be convinced that $0 \leq v \leq 2^n - 1$. To this end, the prover utilizes Pedersen commitments and inner product arguments to convince the verifier that they possess knowledge of an opening a_L such that

$$\langle a_L,\, \vec{2}\rangle = v \text{ and } a_R = \mathbb{1} - a_L \text{ and } a_L \star a_R = \mathbb{0}.$$

This first check ensures that a_L represents v, while the second and third guarantee that a_L is a binary vector. We present the main SLVer Bullet proof generation and verification in Algorithm 2.

Theorem 3.2.1. The SLVer Bullet range proof presented in Algorithm 2 enjoys completeness, witness-extended emulation (a more robust notion of soundness), and is zero-knowledge.

For a proof of this claim when applied strictly to Bulletproofs, which SLVer Bullet is based on, we refer to [BBB$^+$17]. Traditionally, the Bulletproofs verification procedure involves three separate checks: the first ensures that $t = t_0 + t_1 x_0 + t_2 x_0^2$, the second that L and R are correct, and the third that t was computed correctly. The third is simply an inner product computation on witnesses that are contained in π_1 and π_2, so can be performed efficiently by the verifier. The first and second checks however involve elliptic curve point multiplications, which are costly. Hence, the divisor framework may be effectively applied to improve the computational overhead inherent in these checks, with only a slight hit to the completeness and soundness error. In the original Bulletproofs paper, the verifier must compute $6n+11$ elliptic curve point multiplications: $2n+1$ to take products with S, 2 for T_1, T_2, and V, then n for G and H'. Utilizing divisors result in SLVer Bullet, where these $6n + 11$ point multiplications can be reduced to divisor additions, which may be verified much more efficiently than computing the elliptic curve multiplications.

Conclusion

In this work, we extend Eagen's framework for proof-of-computation over elliptic curve points by addressing key limitations, such as lack of protocol clarity, security proofs, and efficiency analysis. We present a refined protocol with detailed pseudocode, rigorous analysis, and zero-knowledge proof extensions. By applying this approach to the Schnorr and Bulletproofs zero-knowledge proof protocols, we obtain highly efficient zero-knowledge proofs with improved verification times, resulting in protocols better suited for real-world cryptocurrency applications.

SLVer Bullet Proof Protocol		
Public: $g, h \in \mathbb{G}$, $G, H \in \mathbb{G}^n$, $V = vg + \gamma h$ **Private**: $v \in K$, where $v \in [0,\, 2^n - 1]$		
PROVER		**VERIFIER**
Compute $a_L \in \{0,\, 1\}^n$ s.t. $\langle a_L,\, \vec{2}\rangle = v$ Set $a_R = \mathbb{1} - a_L$ Sample $\alpha, \rho, \gamma \xleftarrow{\$} K$ and $s_1, s_2 \xleftarrow{\$} K^n$ Set $A = \alpha h + a_L G + a_R H$ and $S = \rho h + s_L G + s_R H$	$\xrightarrow{A,\ S}$	
	$\xleftarrow{y,\ z}$	Sample $y, z \xleftarrow{\$} K^\times$
Define $l(x) = (a_L - z\mathbb{1}) + s_L x$, $r(x) = \vec{y} \star (a_R + z\mathbb{1} + s_R x) + z^2\vec{2}$, and set $\langle l(x),\, r(x)\rangle = t_0 + t_1 x + t_2 x^2$ Set $\delta = t_0 - z^2 v$ For $i = 1, 2$: sample $\tau_i \xleftarrow{\$} K$ and set $T_i = t_i g + \tau_i h$ for $i = 1, 2$	$\xrightarrow{T_1,\ T_2}$	
	$\xleftarrow{x_0}$	Sample $x_0 \xleftarrow{\$} K^\times$
Compute $L = l(x_0)$, $R = r(x_0)$ Set $t = \langle L,\, R\rangle$ Define $\tau = \tau_2 x_0^2 + \tau_1 x_0 + z^2\gamma$ Set $\mu = \alpha + \rho x_0$ Define H' so that $H'_i = y^{-i+1} H_i$ Set $\delta = (z - z^2)\langle \mathbb{1},\, \vec{y}\rangle - z^3\langle \mathbb{1},\, \vec{2}\rangle$ Set $v_1 = [A] + [x_0 S] + [-(L + z\mathbb{1})G]$ $\quad +[(z\vec{y} + z^2\vec{2} - R)H'] + [-\mu h]$ Set $v_2 = [z^2 V] + [(\delta - t)g] + [-\tau h]$ $\quad +[x_0 T_1] + [x_0^2 T_2]$ Compute $\pi_1 \leftarrow \texttt{Prove}(v_1)$ Compute $\pi_2 \leftarrow \texttt{Prove}(v_2)$	$\xrightarrow{L, R, t, \pi_1, \pi_2,}$	
		Parse $v_i \leftarrow \pi_i$ for $i = 1, 2$ Check $\texttt{Verify}(v_1,\, \pi_1) \stackrel{?}{=} 1$ Check $\texttt{Verify}(v_2,\, \pi_2) \stackrel{?}{=} 1$ Check $\langle L,\, R\rangle \stackrel{?}{=} t$

Algorithm 2: The proof generation and verification procedure for SLVer Bullet. Here, $G, H \in \mathbb{G}^n$ are vectors of basepoints, so given $a, b \in \mathbb{Z}_p^n$, we write $aG = \sum_{i=1}^n a_i G_i$ and $bH = \sum_{i=1}^n b_i H_i$.

A Notation In Greater Detail, and Preliminary Definitions

A.1 Polynomials and Rational Functions

Let X, Y, Z be indeterminates over $\overline{K}$, the algebraic closure of K. Finite linear combinations in products of X, Y, and Z are *polynomials*. We use the following usual notation for rings of polynomials over $\overline{K}$ and K.

- $\overline{K}[X,Y] = \left\{\sum_{i=0}^{n}\sum_{j=0}^{m} a_{i,j}X^iY^j \mid a_{i,j} \in \overline{K}\right\}$
- $K[X,Y] = \left\{\sum_{i=0}^{n}\sum_{j=0}^{m} a_{i,j}X^iY^j \mid a_{i,j} \in K\right\}$
- $\overline{K}[X,Y,Z] = \left\{\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} a_{i,j,k}X^iY^jZ^k \mid a_{i,j,k} \in \overline{K}\right\}$
- $K[X,Y,Z] = \left\{\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} a_{i,j,k}X^iY^jZ^k \mid a_{i,j,k} \in K\right\}$

For $f \in K[X,Y]$, say $f = \sum_i \sum_j a_{ij}X^iY^j$, define $\deg(f) = \max\{i+j \mid a_{ij} \neq 0\}$ (and similarly for the other polynomial rings). Each of these rings (say R) admits a structure-preserving map $\deg : R \to \mathbb{N}_0 = \mathbb{N}\cup\{0\}$ such that $\deg(fg) = \deg(f)+\deg(g)$ and $\deg(f+g) \leq \max\{\deg(f), \deg(g)\}$, where we follow the convention that the degree of the zero polynomial is $-\infty$. Also, $\deg(f) = 0$ if and only if f is a non-zero constant polynomial.

If $d \in \mathbb{N}_0$ and $f \in \overline{K}[X,Y]$, and for all $\lambda \in \overline{K}^\times$ we have that $f(\lambda X, \lambda Y) = \lambda^d f(X,Y)$, then we say f is *homogeneous with degree d*; otherwise, *inhomogeneous*. Define an equivalence relation on the subset of homogeneous polynomials in $\overline{K}[X,Y,Z]$ such that every $f(X,Y,Z)$ relates to $\lambda f(X,Y,Z)$ for every $\lambda \in \overline{K}^\times$. Denote the equivalence class of f under this equivalence relation with $f(X:Y:Z)$, and the set of equivalence classes under this relation with $\overline{K}[X:Y:Z]$. Define $\deg : \overline{K}[X:Y:Z] \to \mathbb{N}_0$ by declaring $\deg(f(X:Y:Z)) = \deg(f(X,Y,Z))$. Similarly define $K[X:Y:Z]$ and its degree function. The rings $\overline{K}[X,Y]$, $K[X,Y]$, $\overline{K}[X,Y,Z]$, and $K[X,Y,Z]$ are all integral domains. We also define the respective fields of rational functions $\overline{K}(X,Y)$, $K(X,Y)$, $\overline{K}(X,Y,Z)$, and $K(X,Y,Z)$ as usual.

A.2 Affine and Projective Planes

Define $\mathbb{A}^2(\overline{K}) = \left\{(x,y) \in \overline{K}^2\right\}$, $\mathbb{A}^3(\overline{K}) = \left\{(x,y,z) \in \overline{K}^3\right\}$, $\mathbb{A}^2(K) = \left\{(x,y) \in K^2\right\}$, and $\mathbb{A}^3(K) = \left\{(x,y,z) \in K^3\right\}$. We say the elements of $\mathbb{A}^2(\overline{K})$ are *affine points* and the elements of $\mathbb{A}^2(K)$ are *K-rational* affine points. For any ideal $I \subseteq \overline{K}[X,Y]$, the *affine vanishing set* of I is $v_a(I) = \left\{(x,y) \in \mathbb{A}^2(\overline{K}) \mid f(x,y) = 0 \text{ for all } f \in I\right\}$. If $f(X,Y) \in \overline{K}[X,Y]$, we define the affine vanishing set for f as the affine vanishing set for the principally generated $I = (f)$, which we denote with $v_a(f)$.

Given affine vanishing set V, let $I(V) = \left\{f \in \overline{K}[X,Y] \mid f(x,y) = 0 \text{ for all } (x,y) \in V\right\}$ be the *ideal of V*. It is easy to check that $I(V)$ is an ideal of $\overline{K}[X,Y]$. We say an affine vanishing set V is *defined over K*, denoted with shorthand V/K, if $I(V)$ is generated as an ideal in $\overline{K}[X,Y]$ by a subset of $K[X,Y]$. For any affine vanishing set V, let $I(V/K) = I(V) \cap K[X,Y]$. It is easy to show

that $I(V/K)$ is an ideal. Moreover, V is defined over K if and only if $I(V)$ is generated as an ideal in $\overline{K}[X,Y]$ by $I(V/K)$, i.e. $I(V) = I(V/K)\overline{K}[X,Y]$.

We say V is an *affine variety* if $I(V)$ is a prime ideal in $\overline{K}[X,Y]$. The *affine coordinate ring* of an affine variety V/K is the quotient ring $K[V] = \frac{K[X,Y]}{I(V/K)}$. The *function field* of V is the field of fractions associated with $K[V]$, i.e. the set of all formal fractions $\frac{f}{g}$ where $f, g \in K[V]$ and $g \neq 0$, under an equivalence relation defined by relating $\frac{f_1}{g_1}$ to $\frac{f_2}{g_2}$ if and only if $f_1 g_2 - f_2 g_1 = 0$. We denote the function field $K(V)$.

Define an equivalence relation on $\mathbb{A}^3(\overline{K}) \setminus \{\mathbb{0}\}$ by relating (x,y,z) to (x',y',z') if and only if there exists $\lambda \in \overline{K}^\times$ such that $(x',y',z') = (\lambda x, \lambda y, \lambda z)$. Denote the equivalence class of (x,y,z) with $(x:y:z)$. Use $\mathbb{P}^2(\overline{K})$ to be the set of equivalence classes under this relation. Similarly define $\mathbb{P}^2(K)$ in the evident way. We say the elements of $\mathbb{P}^2(\overline{K})$ are *projective points* and the elements of $\mathbb{P}^2(K)$ are the *K-rational* projective points. For any homogeneous ideal $I \subseteq \overline{K}[X,Y,Z]$, the *projective vanishing set* of I is $v_p(I) = \big\{(x:y:z) \in \mathbb{P}^2(\overline{K}) \mid f(x,y,z) = 0 \text{ for all } f \in I\big\}$. Since the generating set for I contains only homogeneous polynomials, the presence of any representative in $(x:y:z)$ which kills all $f \in I$ implies all representatives kill all $f \in I$. For any $f \in \overline{K}[X,Y,Z]$, we define the projective vanishing set for f as the projective vanishing set for the ideal $I = (f)$, and we use the notation $v_p(f)$.

Affine points $P = (x,y) \in \mathbb{A}^2(\overline{K})$ (or $\mathbb{A}^2(K)$, respectively) naturally map to projective points $P^* = (x:y:1) \in \mathbb{P}^2(\overline{K})$ (or $\mathbb{P}^2(K)$, respectively). This map is injective, but $\mathbb{P}^2$ contains many copies of $\mathbb{A}^2$ and this map is not unique. Projective points $P = (x:y:z) \in \mathbb{P}^2(\overline{K})$ with $z \neq 0$ naturally map to affine points $P_* = (\frac{x}{z}, \frac{y}{z}) \in \mathbb{A}^2(\overline{K})$ in a similar way, but surjectively. The same statement as above holds with $\mathbb{A}^2(K)$ and $\mathbb{P}^2(K)$ replacing their counterparts everywhere.

There exists natural *homogenization* maps carrying $f \in K[X,Y]$ (or $f \in \overline{K}[X,Y]$) to a homogeneous $f^* \in K[X,Y,Z]$ (or $f^* \in \overline{K}[X,Y,Z]$, respectively) where $f^* = Z^{\deg(f)} f(\frac{X}{Z}, \frac{Y}{Z})$. This map is injective. We can also dehomogenize a homogeneous polynomial $g \in K[X:Y:Z]$ (or $\overline{K}[X,Y,Z]$) to a polynomial in the affine coordinate ring $g \mapsto g_*(X,Y) = g(X:Y:1) \in K[X,Y]$ (or $\overline{K}[X,Y]$, respectively). This map is surjective. The distinguished polynomial $e(X,Y) = Y^2 - X^3 - \alpha X - \beta$ has homogenization $e^*(X,Y) = Y^2Z - X^3 - \alpha XZ^2 - \beta Z^3$. The map $(x,y) \hookrightarrow (x:y:1)$ sends zeroes of $f \in K[X,Y]$ to zeroes of $f^* \in K[X:Y:Z]$.

Using these maps, we can translate $\mathbb{A}^2$ to $\mathbb{P}^2$ and back again. In this way, we define *projective closure* of an affine variety V as the projective vanishing set $\overline{V}$ such that $I(\overline{V}) = \{f^* \mid f \in I(V)\}$. We also define the function field for a projective variety $V \subseteq \mathbb{P}^2$ as the function field for the affine restriction [5].

A.3 Plane Curves

A *plane affine curve* $\mathcal{F}$ is just the affine vanishing set of some $f \in \overline{K}[X,Y]$, i.e. $\mathcal{F} = v_a(f)$. If g is a multiplicative factor of f, then the curve $\mathcal{G}$ corresponding to g is called a *component* of $\mathcal{F}$.

We define the *intersection number* of a curve $\mathcal{F}$, simple/non-singular points of $\mathcal{F}$, singular points

[5] This restriction is more delicate in general than discussed here; see [Sil09].

of $\mathcal{F}$, and the multiplicity of a point P on $\mathcal{F}$ as in [HKTT08]. In particular, if $\ell \in \overline{K}[X,Y]$ is a line which is not a component of $\mathcal{F}$, the *multiplicity* of P is $\min m \mid \ell$ a line through P where m is the intersection number of ℓ and $\mathcal{F}$ at P. Multiplicity theory is a rich rabbit hole preceding Zariski, so we do not go into further detail.

In the sequel, we use the distinguished family of polynomials parameterized by $\alpha, \beta \in K^2$ of the form $e = Y^2 - X^3 - \alpha X - \beta \in K[X,Y]$ such that $4\alpha^3 + 27\beta^2 \neq 0$; these are exactly the non-singular elliptic curves. We note that some authors consider the term "elliptic curve" to refer specifically to non-singular elliptic curves, while others do not; see [Sil09, Dum]. While we make a distinction for pedagogical reasons, we do not consider singular elliptic curves here.

Plane affine curves may have any of the properties of affine vanishing sets discussed above. A *plane projective curve* $\mathcal{F}$ is just the projective vanishing set of some homogeneous $f \in \overline{K}[X:Y:Z]$, i.e. $\mathcal{F} = v_p(f)$. In both cases, we define $\deg(\mathcal{F}) = \deg(f)$. Plane projective curves may have any of the properties of projective vanishing sets discussed above.

Recall the distinguished family of polynomials $e = Y^2 - X^3 - \alpha X - \beta \in K[X,Y]$ such that $4\alpha^3 + 27\beta^2 \neq 0$. Define $\mathcal{E} = v_p(e)$ as the plane projective curve of the homogenization e^*. It is easy to show that the only $(x:y:z) \in \mathcal{E}$ with $z = 0$ is $(0:1:0)$. We call this the *point at infinity* on $\mathcal{E}$, and we write $O = (0:1:0)$. It is also easy to show that $\mathcal{E}$ is a projective variety, $I(\mathcal{E}/K) = (e) \subseteq K[X:Y:Z]$ is the associated ideal over K, and the associated function field is as follows.

$$K(\mathcal{E}) = \left\{ \frac{f(X,Y) + (e)}{g(X,Y) + (e)} \mid f, g \in K[X,Y], g \notin (e) \right\} \tag{A.1}$$

Except in the case of confusion or for clarification, we often omit the ideal (e) from our notation in the sequel, writing $f/g \in K(\mathcal{E})$, with the understanding that the numerator and denominator of every element in the function field is a coset modulo this ideal.

However, representing a function field element with coefficient representations of two arbitrary bivariate polynomials is expensive. Indeed if $m = \deg(f)$ and f is bivariate, then we can write $f = \sum_{i=0}^{m} \sum_{j=0}^{i} c_{i,j} X^{i-j} Y^j$, requiring $1 + 2 + 3 + \cdots + m + (m+1) = \frac{1}{2}(m+1)(m+2)$ coefficients from K. Thus, if $\deg(f), \deg(g) \leq m$, then we can represent $\frac{f}{g}$ with $(m+1)(m+2)$ coefficients from K. On the other hand, more efficient representations are available. Indeed, if $f \in K[X,Y]$,

then as $K[X,Y] = K[X][Y]$, we have the following.

$$\begin{aligned}
f(X,Y) =& \sum_{j=0}^{m} b_j(X)Y^j \text{ where } b_j \in K[X] \text{ for } 0 \le j \le m \\
=& \sum_{j=0}^{\lceil m/2 \rceil} b_{2j}(X)Y^{2j} + b_{2j+1}(X)Y^{2j+1} \text{ where } b_{m+1} = 0 \\
=& \sum_{j=0}^{\lceil m/2 \rceil} Y^{2j}\left(b_{2j}(X) + b_{2j+1}(X)Y\right) \\
=& \sum_{j=0}^{\lceil m/2 \rceil} \left(X^3 + \alpha X + \beta\right)^j \left(b_{2j}(X) + b_{2j+1}(X)Y\right) \\
=& \underbrace{\left(\sum_{j=0}^{\lceil m/2 \rceil} \left(X^3 + \alpha X + \beta\right)^j b_{2j}(X)\right)}_{a(X)} + \underbrace{\left(\sum_{j=0}^{\lceil m/2 \rceil} \left(X^3 + \alpha X + \beta\right)^j b_{2j+1}(X)\right) Y}_{b(X)Y}
\end{aligned}$$

where $a(X) = \sum_{j=0}^{\lceil m/2 \rceil} (X^3+\alpha X+\beta)^j b_{2j}(X) \in K[X]$ and $b(X) = \sum_{j=0}^{\lceil m/2 \rceil} (X^3+\alpha X+\beta)^j b_{2j+1}(X) \in K[X]$. Representing a polynomial f this way requires $\deg(a) + \deg(b) + 2$ elements of K. So, we can forget our above representation of elements of the function field $\frac{f}{g}$ with $f, g \in K[X,Y]$, and use $\frac{a_1(X)+b_1(X)Y}{a_2(X)+b_2(X)Y}$ for some $a_1, a_2, b_1, b_2 \in K[X]$, representing the function field element with only $\deg(a_1) + \deg(a_2) + \deg(b_1) + \deg(b_2) + 4$ elements of K instead.

We may go further, however. For each $a(X)+b(X)Y+(e)$, we have an element $a(X)-b(X)Y+(e) \in K[X,Y]/(e)$ satisfying the following due to the ideal $I(\mathcal{E}/K) = (e)$ and the multiplicative absorption property of ideals.

$$\begin{aligned}
(a_1(X) + b_1(X)Y + (e))(a_1(X) - b_1(X)Y + (e)) =& a_1^2 - b_1^2Y^2 + (e) \\
=& a_1^2 - (X^3 + \alpha X + \beta)b_1^2 + (e)
\end{aligned}$$

Hence, for any $\frac{a_1(X)+b_1(X)Y+(e)}{a_2(X)+b_2(X)Y+(e)} \in K(\mathcal{E})$, we can apply an identity map by multiplying and dividing by $(a_2(X) - b_2(X)Y + (e))$, yielding another representative whose denominator does not depend on Y as follows

$$\begin{aligned}
&\frac{a_1(X) + b_1(X)Y + (e)}{a_2(X) + b_2(X)Y + (e)} \cdot \frac{a_2(X) - b_2(X)Y + (e)}{a_2(X) - b_2(X)Y + (e)} \\
&= \frac{a_1(X)a_2(X) - a_1(X)b_2(X)Y + a_2(X)b_1(X)Y - b_1(X)b_2(X)Y^2 + (e)}{(a_2(X))^2 - (b_2(X))^2Y^2 + (e)} \\
&= \frac{f_1(X) + f_2(X)Y + (e)}{f_3(X) + (e)}
\end{aligned}$$

where $f_1(X) = a_1(X)a_2(X) - (X^3 + \alpha X + \beta)b_1(X)b_2(X)$, $f_2(X) = a_2(X)b_1(X) - a_1(X)b_2(X)$, and $f_3(X) = (a_2(X))^2 - (X^3 + \alpha X + \beta)(b_2(X))^2$. This requires only $\deg(f_1(X)) + \deg(f_2(X)) +$

$\deg(f_3(X)) + 3$ elements of K.

Thus, we write every element of $K(\mathcal{E})$ as $\frac{f_1(X)+f_2(X)Y+(e)}{f_3(X)+(e)}$ for some polynomials $f_0(X)$, $f_1(X)$, $f_2(X) \in K[X]$. Later, we outperform this slightly, as we practically only consider elements such that $\deg(f_3(X)+(e)) = 0$, i.e. $f_3(X)+(e) = 1+(e)$, requiring only $\deg(f_1)+\deg(f_2)+2$ coefficients to describe f.

A.4 Elliptic Curve Groups

For the remainder of this manuscript, we assume all elliptic curve groups have prime order strictly greater than 3.

Recall the distinguished polynomial $e(X,Y) = Y^2 - X^3 - \alpha X - \beta \in K[X,Y]$ where $4\alpha^3 + 27\beta^2 \neq 0$, and consider the plane projective curve $\mathcal{E} = v_p(e)$. Note $I(\mathcal{E}/K) = (e)$. This $\mathcal{E}$ is an elliptic curve with affine coordinate ring $K[\mathcal{E}] = \frac{K[X,Y]}{(e)}$ and function field $K(\mathcal{E}) = \mathrm{Frac}(K[\mathcal{E}])$. For a natural number $n \in \mathbb{N}$, the surface $\mathcal{E}^n$ is a variety with coordinate ring $K[\mathcal{E}^n] = K[X_1, Y_1, \ldots, X_n, Y_n]/I(\mathcal{E}^n/K)$, where $I(\mathcal{E}^n/K)$ is the ideal generated by the set $\{e(X_i, Y_i) \mid i \in [n]\}$, and function field $K(\mathcal{E}^n) = \mathrm{Frac}(K[X_1, Y_1, \ldots, X_n, Y_n]/I(\mathcal{E}^n/K))$.

We additionally take Bézout's theorem for granted (see Theorem I.7.8 in [Har13]).

Theorem A.4.1 (Bézout's Theorem)**.** Let $\mathcal{F}_1$, $\mathcal{F}_2$ be distinct curves in $\mathbb{P}^2(\overline{K})$, having degrees n_1, n_2, respectively. Let $\mathcal{F}_1 \cap \mathcal{F}_2 = \{P_1, \ldots, P_s\}$, and say, for each $1 \leq i \leq s$, that point P_i has multiplicity m_i. Then $\sum_{i=1}^{s} m_i = n_1 n_2$.

The projective variety corresponding to $\mathcal{E}$ is $\mathbb{G}(\mathcal{E}) = v_p(e)$. Then e^* is the homogenization of e defined earlier. This $\mathbb{G}(\mathcal{E})$ admits an abelian group operation $+$ following the elliptic curve group law, which determines decompositions of the group identity in $\mathbb{G}(\mathcal{E})$ as follows. First, the point at infinity $O = (0:1:0)$ is distinguished as the group identity. Next, collinear points define decompositions of O. Bezout's theorem implies every line intersects $v_p(\mathcal{E})$ at 3 points, counting multiplicities. Hence, every set of collinear points is a triple $(P, Q, R) \in \mathbb{G}^3$ such that $P+Q+R = O$. We have three cases: P has multiplicity 3, 2, or 1.

- If $P = Q = R$, then $3P = O$, i.e. P has multiplicity 3. This is contraindicated for a non-singular elliptic curve $\mathcal{E}$ of prime order $p > 3$.
- If $P = Q$, then $P + P + R = 2P + R = O$, so $2P = -R$. In this case, we sum points with themselves, however this implies that the line ℓ interpolating P and R intersects $\mathcal{E}$ at P with multiplicity 2. Since $\mathcal{E}$ is non-singular, it must be that ℓ is *tangent* to $\mathcal{E}$ at P.
- If P, Q, and R are all distinct, we have two sub-cases: one of these points is an inverse of another, or not.
  - If $P = -Q$, then $P+Q+R = O$ implies that $R = O$. In this case, the line interpolating P, Q, and R includes the point at infinity O. The lines passing through O and intersecting $\mathcal{E}$ at a non-identity point are precisely the *vertical lines*. Similarly for the case $P = -R$.

– All other cases are when P, Q, and R are distinct, not inverses of each other, and correspond to lines which are neither vertical nor tangent.

That is to say, to find $-P$, we use the other root on the line passing through P which is *vertical*; to sum a point P with itself we use the other root on the line passing through P which is *tangent to* $\mathcal{E}$; to add two generic non-identity points P, Q we use the third root R of the line ℓ interpolating P, Q, setting $P + Q = -R$. These rules are sufficient for generating the elliptic curve group $\mathbb{G}(\mathcal{E})$. In the sequel, we denote this $\mathbb{G}$, as we use the same $\mathcal{E}$ throughout.

We emphasize that group points like P, Q, R are elements of $v_p(e)$, and thus are points from the projective plane, of the form $(x : y : z)$. All points (x, y) from $\mathbb{A}^2$ are embedded as $(x : y : 1)$, and the only solution to e^* in $\mathbb{P}^2$ with $z = 0$ is exactly $(0 : 1 : 0)$, so if P, Q, and R are non-identity, then these have $z \neq 0$. That is to say, each $(x : y : z)$ has a representative $(\frac{x}{z}, \frac{y}{z}, 1)$, and we may re-label our variables to assume the non-identity points can all be written $(x : y : 1) \in \mathbb{P}^2$. Such a point admits the dehomogenization, an affine point $(x, y) \in \mathbb{A}^2$ (which can, in turn, be re-embedded into $\mathbb{P}^2$). For this reason, we often conflate non-identity points like P, Q, $R \in v_p(e)$ with their dehomogenizations (x_P, y_P), (x_Q, y_Q), and (x_R, y_R), clarifying whether we are handling affine or projective points with context in the text.

However, the y coordinates are *almost* uniquely determined by their corresponding x coordinate. Indeed, for each $T \in \{P, Q, R\}$, only two elements of K satisfy $y_T^2 = x_T^3 + \alpha x_T + \beta$. Distinguishing which y_T corresponds to the point T simply requires a convention for interpreting a parity/sign bit. Thus, representing (x_T, y_T) requires only one more bit than representing x_T alone.

In the sequel, we leave $\mathcal{E}$ implicit and denote this $\mathbb{G}$, as we use the same $\mathcal{E}$ throughout. Moreover, the group law as described here is equivalent to a group law defined for rational functions, not merely lines; the proof of this is beyond the scope of this text. See [Sil09] for details.

A.4.1 Group Law and Summing Points

We use lines to compute $P + Q$ explicitly as a rational function for a pair $(P, Q) \in (\mathbb{G} \setminus \{O\})^2$ such that $P \neq \pm Q$; this is useful in our derivations in Appendix B later. As both are not the identity and $P \neq \pm Q$, the line interpolating P and Q is not vertical. By Bézout's theorem, the line must intersect $\mathcal{E}$ at an additional point R, which may coincide with P or Q. In the case where the three points are distinct, then these intersection points all have a multiplicity of 1. The group law states $P + Q + R = O$. We compute (x_R, y_R) to represent R.

Note that, as elements of $\mathbb{G}$, we may write

$$P = (x_P : y_P : z_P) = \left\{(x, y, z) \mid \forall \lambda \in K^\times, (x, y, z) = (\lambda x_P, \lambda y_P, \lambda z_P)\right\}$$

and $Q = (x_Q : y_Q : z_Q)$ similarly. As non-identity elements, $P \neq O = (0 : 1 : 0)$ so $z_P \neq 0$, and the equivalence class $(x_P : y_P : z_P)$ contains a representation of the form $(x, y, 1)$ (and similarly for Q. Thus, we lose no generality by assuming $z_P = z_Q = 1$, in which case we can write $P = (x_P : y_P : 1)$ and $Q = (x_Q : y_Q : 1)$. Then P and Q may be dehomogenized to affine points (x_P, y_P), $(x_Q, y_Q) \in \mathcal{E}$;

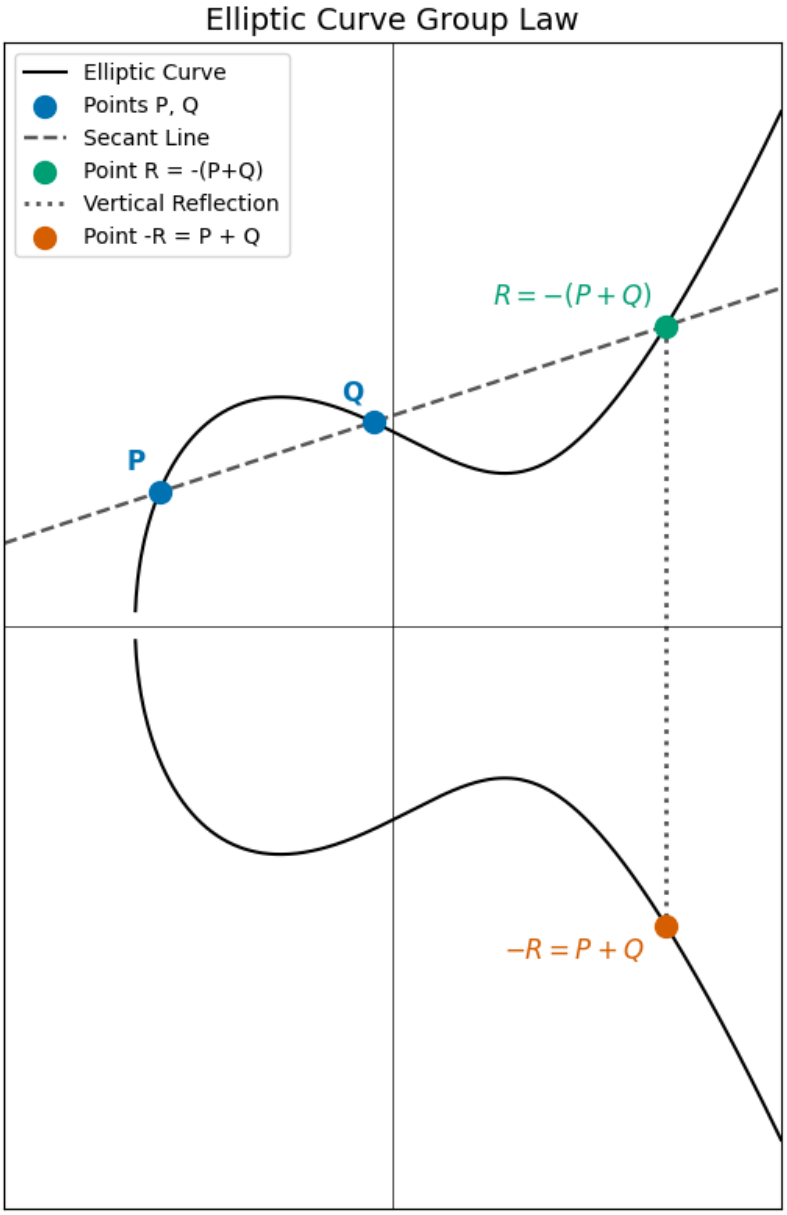


Table 1: The points P, Q, R are collinear, so their sum is $P + Q + R = O$, where O is the group identity. In particular, $R = -(P + Q)$ and its reflection across the X-axis is $-R = P + Q$.

we identify $P = (x_P : y_P : 1)$ with (x_P, y_P), and similarly for Q, in our discussion below, keeping in mind that we are actually transferring between points on the projective plane to points on the affine plane via the dehomogenization and homogenization maps.

Define $\lambda = \frac{y_P - y_Q}{x_P - x_Q}$ and $\mu = y_P - \lambda x_P$ as functions of P and Q. These λ, μ parameterize the interpolating line $\ell(X, Y) = Y - \lambda X - \mu \in K[X, Y]$, and therefore $y_T = \lambda x_T + \mu$ for each $T \in \{P, Q, R\}$. Also, since $e(x_T, y_T) = 0$, we have $(\lambda x_T + \mu)^2 = x_T^3 + \alpha x_T + \beta$. Thus, the points P, Q, R have x coordinates which are distinct roots of the third degree polynomial $X^3 - \lambda^2 X^2 + (\alpha - 2\lambda\mu)X + (\beta - \mu^2)$ in X. All the roots are accounted for, so the polynomial splits into linear factors.

$$(X - x_P)(X - x_Q)(X - x_R) = X^3 - \lambda^2 X^2 + (\alpha - 2\lambda\mu)X + (\beta - \mu^2)$$

The linear independence of $\{1, X, X^2, X^3\}$ implies we may match coefficients, so $-\lambda^2 = -x_P - x_Q - x_R$, where x_P and x_Q are already given.

$$\begin{cases} x_R &= \lambda^2 - x_P - x_Q \\ y_R &= \lambda x_R + \mu \end{cases}$$

Recalling λ and μ defined above are rational functions, it is clear that these are both rational in (x_P, s_P, x_Q, s_Q), as promised. Moreover, the group law and the elliptic curve equation demand the following constraints are satisfied.

$$\begin{cases} \lambda = & \frac{y_Q - y_P}{x_Q - x_P} \\ \mu = & y_P - \lambda x_P \\ y_P = & \lambda x_P + \mu \\ y_Q = & \lambda x_Q + \mu \\ y_P^2 = & x_P^3 + \alpha x_P + \beta \\ y_Q^2 = & x_Q^3 + \alpha x_Q + \beta \end{cases} \tag{A.2}$$

We emphasize that, in our notation, lowercase terms like (x_T, y_T) denote the coordinate pairs of a point denoted with an uppercase term T, whereas uppercase terms like (X_T, Y_T) refer to variable pairs of transcendental elements, which we handle as coordinate pairs of unspecified points.

A.5 Divisors over Elliptic Curves

For each $T = (x : y : z) \in \mathbb{G}$, let $[T]$ be a formal symbol corresponding to T. A *divisor* is a finite sum of these formal symbols. These sums can be summed, so divisors generate an additive group which we call $\mathrm{Div}_K(\mathcal{E})$. We use functions $\nu : \mathbb{G} \to \mathbb{Z}$ with finite supports to represent a divisor $[\nu] = \sum_{T \in \mathbb{G}} \nu(T)\,[T]$. The *degree* of $[\nu]$ is $\deg([\nu]) = \sum_{T \in \mathbb{G}} \nu(T)$. Let $\mathrm{Div}_K^0(\mathcal{E})$ denote the set of degree-zero divisors. Note $\mathrm{Div}_K^0(\mathcal{E}) \subseteq \mathrm{Div}_K(\mathcal{E})$ and is closed under addition, i.e. is a subgroup.

Given any $f \in K(\mathcal{E})^\times$, we can define a map $\mathcal{E} \to \mathbb{Z}$ by mapping $T \mapsto \mathrm{ord}_T(f)$, and this has with finite support as the numerator and denominator of f both have finite degree (and hence a f has a finite number of roots and poles, even counting multiplicity). Denote $\mathrm{div}(f) = \sum_{T\in\mathcal{E}} \mathrm{ord}_T(f)[T]$. Define $\mathrm{Prin}(\mathcal{E}) = \{\mathrm{div}(f) \mid f \in K(\mathcal{E})\}$. Note we have the following chain of subgroups: $\mathrm{Prin}(\mathcal{E}) \subseteq \mathrm{Div}^0_K(\mathcal{E}) \subseteq \mathrm{Div}_K(\mathcal{E})$. This leads us back to the following relations mentioned in Section 1.2

$$\mathfrak{R} = \big\{(f, [\nu]) \mid f \in K(\mathcal{E})^\times, [\nu] \in \mathrm{Div}^0_K(\mathcal{E}), \mathrm{div}(f) = [\nu]\big\}$$

$$\mathfrak{R}' = \big\{(f, [\nu]) \mid |\mathrm{supp}(\nu)| \geq 3, f \neq g^p \cdot h \quad \forall g, h \in K(\mathcal{E})^\times\big\} \subseteq \mathcal{R}_{\mathrm{div}}$$

We take the following for granted.

Theorem A.5.1. Let $f \in K(\mathcal{E})$ be nonzero, $v_a(f) \setminus \{O\} = \{P_1, \ldots, P_n\}$, and $v_a(f^{-1}) \setminus \{O\} = \{Q_1, \ldots, Q_s\}$ with $n + s \geq 1$. If each P_i has multiplicity $m_i \in \mathbb{N}$ and each Q_j has multiplicity $t_j \in \mathbb{N}$, and $\sum_i m_i - \sum_j t_j \neq 0$, then:

(a) $\sum_i m_i - \sum_j t_j > 0$ implies f has a pole at O with multiplicity $\sum_i m_i - \sum_j t_j$,

(b) $\sum_i m_i - \sum_j t_j < 0$ implies f has a root at O with multiplicity $\sum_j t_j - \sum_i m_i$, and

(c) $\sum_i m_i P_i - \sum_j t_j Q_j = O$.

In particular, $\mathrm{div}(f) = \sum_i m_i[P_i] - \sum_j t_j[Q_j] - (\sum_i m_i - \sum_j t_j)[O]$, $\deg(\mathrm{div}(f)) = 0$, and $\mathrm{div}(f)$ represents a nontrivial decomposition of $O \in \mathbb{G}$, namely $\sum_i m_i P_i - \sum_i t_j Q_j = O$.

If $f_1, f_2 \in K(\mathcal{E})$ have $(v_p(f_1^*) \cup v_p(1/f_1^*)) \cap (v_p(f_2^*) \cup v_p(1/f_2^*)) = \emptyset$, we say f_1 and f_2 have *disjoint roots and poles*. If f_1 and f_2 have disjoint roots and poles, it is easy to check that $v_p(f_1^* f_2^*) = v_p(f_1^*) \cup v_p(f_2^*)$ and $v_p(1/(f_1^* f_2^*)) = v_p(1/f_1^*) \cup v_p(1/f_2^*)$. Thus, for such an f_1, f_2 pair, we have that $\mathrm{div}(f_1 f_2) = \mathrm{div}(f_1) + \mathrm{div}(f_2)$. More generally, div is a group homomorphism from the multiplicative subgroup of the function field $K(\mathcal{E})^\times$ to the additive (sub)group $\mathrm{Prin}(\mathcal{E})$.

Note that if f_1 and f_2 do not have disjoint roots and poles, then the common roots and poles may partially or completely cancel in the product $f_1 f_2$. This expresses itself as terms canceling in the sum $\mathrm{div}(f_1) + \mathrm{div}(f_2)$.

A.6 Weil Reciprocity

Every $f \in K(\mathcal{E})$ induces a homomorphism. Indeed, let $[\eta] = \sum_{T\in\mathbb{G}} \eta(T)[T]$ be any divisor such that the support of η is disjoint from the roots and poles of f. Then let $\widehat{f}([\eta]) = \prod_{T\in\mathbb{G}} f^*(T)^{\eta(T)}$; this is well-defined because the points of $[\eta]$ are disjoint from the roots and poles of f.

Theorem A.6.1. For each $f \in K(\mathcal{E})$, if $[\eta_1]$ and $[\eta_2]$ have supports that are disjoint from the roots and poles of f, then $\widehat{f}([\eta_1 + \eta_2]) = \prod_{T\in\mathbb{G}} f^*(T)^{\eta_1(T)+\eta_2(T)} = \widehat{f}([\eta_1])\widehat{f}([\eta_2])$.

For the remainder of this document, we assume that $[\eta_1]$ and $[\eta_2]$ have mutually disjoint supports.

Theorem A.6.2 (Weil Reciprocity Law)**.** Let $f, g \in K(\mathcal{E})^\times$ with disjoint supports that also exclude O, where $\mathrm{div}(f) = \sum_T \nu(T)[T]$ and $\mathrm{div}(g) = \sum_T \eta(T)[T]$. Then

$$\widehat{f}(\mathrm{div}(g)) = \prod_{T \in \mathbb{G}} f^*(T)^{\eta(T)} = \prod_{S \in \mathbb{G}} g^*(S)^{\nu(S)} = \widehat{g}(\mathrm{div}(f)).$$

In particular, if g has a degree 1 polynomial representing its numerator $aX + bY + c + (e)$ and $1 + (e)$ as the denominator, then $\widehat{g}([\nu]) = \prod_T (ax_T + by_T + c)^{\nu(T)}$. For the remainder of this document, when we refer to $f, g \in K(\mathcal{E})^\times$, we will assume that they have disjoint supports which exclude O, the point at infinity.

A.7 Derivations

Polynomials admit formal derivatives and partial derivatives as usual: if $f = \sum_i a_i X^i$, define $f' = \sum_i i a_i X^{i-1}$, and if $f = \sum_{i,j} a_{i,j} X^i Y^j$, define $\frac{\partial}{\partial X} f = \sum_{i,j} i a_{i,j} X^{i-1} Y^j$ and $\frac{\partial}{\partial Y} f = \sum_{i,j} j a_{i,j} X^i Y^{j-1}$.

More generally, given any field extension $K \subseteq L$, an additive group homomorphism $d : L \to L$ is said to be a *derivation over* K if $d(K) = \{0\}$ and d satisfies Leibniz' rule: $d(ab) = d(a)b + ad(b)$. Given an intermediate field, $K \subseteq F \subseteq L$, such that $d(F) \subseteq F$, the restriction of d over K on L to F yields a derivation over K on F. In the case $L = \overline{K}(X, Y)$, derivations $\frac{\partial}{\partial X}$ and $\frac{\partial}{\partial Y}$ are defined as usual (and $\frac{\partial}{\partial Z}$ is defined over $\overline{K}(X, Y, Z)$ similarly). All derivations are linear combinations of these derivations $\frac{\partial}{\partial X}$, $\frac{\partial}{\partial Y}$, and $\frac{\partial}{\partial Z}$. Indeed,

(i) for any $a, b \in L$, every linear combination $a(X, Y)\frac{\partial}{\partial X} + b(X, Y)\frac{\partial}{\partial Y}$ is a derivation over K on L, and

(ii) every derivation over K on L is of the form $a(X, Y)\frac{\partial}{\partial X} + b(X, Y)\frac{\partial}{\partial Y}$, for some $a, b \in \overline{K}(X, Y)$.

For $F \subseteq \overline{K}(X, Y)$, the derivations satisfying $d(F) \subseteq F$ are precisely the derivations with corresponding $(a, b) \in F^2$.

Similarly, $\overline{K}(X, Y, Z)$ has derivations of the form $d = a\frac{\partial}{\partial X} + b\frac{\partial}{\partial Y} + c\frac{\partial}{\partial Z}$ for some $a, b, c \in \overline{K}(X, Y, Z)$, and if all a, b, c are in an intermediate field $K \subseteq F \subseteq \overline{K}(X, Y, Z)$, then d is a derivation on F.

Later, derivations are used with elements of the function field $K(E)$, where the variables X and Y are not independent (they satisfy a polynomial relation, namely the defining equation of the curve. [Bas25]), but those derivations are inherited from the usual ones on $\overline{K}(X, Y)$.

Given any derivation $d : L^\times \to L$ over K, there is a natural map $\delta : L^\times \to L$ mapping $f \mapsto \frac{df}{f}$. Note that given any $f, g \in L$, we have that

$$\delta(fg) = \frac{d(fg)}{fg} = \frac{d(f)g + fd(g)}{fg} = \frac{d(f)}{f} + \frac{d(g)}{g} = \delta(f) + \delta(g).$$

In particular, δ is a group homomorphism from the multiplicative subgroup of L to the additive group of L.

A.8 Schwartz-Zippel

We take the following for granted.

Lemma A.8.1. Let $0 \neq f \in \overline{K}[Z_1, Z_2, \ldots, Z_n]$ with $\deg(f) = d$, and let S be a finite subset of $\overline{K}$. If $s_1, s_2, \ldots, s_n$ are sampled independently and uniformly from S, then the probability that $f(s_1, \ldots, s_n) = 0$ is at most $\frac{d}{|S|}$.

It is not strictly necessary for S to be a subset of an algebraically closed field, but this definition of Schwartz-Zippel is sensible for our purposes.

B Deriving Verification Equations

Much of the narrative herein are used in supporting arguments for our completeness and soundness proofs above. All the derivations here are included in full for reference.

B.1 Logarithmic Derivative of a Function Field Element at the Zeroes of a Line

In this section, we formalize taking the logarithmic derivative of $f(P)f(Q)f(R)$ for collinear, distinct, non-identity points $P, Q, R \in \mathcal{E}$. Indeed, for any $T \in \mathcal{E}$, $f(T)$ is just an element of K, and so $dT = 0$, yet a variable point is a rational function of transcendental elements with non-zero derivation in general.

Recall that any points P, Q, R which are collinear, distinct, and non-identity necessarily satisfy $P + Q + R = O$ in the group law.

To handle P, Q, R as variable points on the curve $\mathcal{E}$, say by introducing transcendental elements X_P, Y_P, X_Q, and Y_Q, where X_P corresponds to the x-coordinate of (variable) point P, and Y_P is the y-coordinate of P, and similarly for (variable) point Q. Then a derivation on $K(\mathcal{E})$ extends naturally to a derivation on $K(\mathcal{E}^2)$ via partial derivations $\frac{\partial}{\partial X_P}$, $\frac{\partial}{\partial X_Q}$, $\frac{\partial}{\partial Y_P}$, and $\frac{\partial}{\partial Y_Q}$.

Given P and Q such that $P \neq Q$ and such that the interpolating line betwixt them isn't tangent to $\mathcal{E}$, there is a unique R satisfying $P + Q + R = O$, so we only need transcendentals X_P, Y_P, X_Q, and Y_Q to determine X_R and Y_R; moreover these are rational functions in X_P, Y_P, X_Q, and Y_Q.

Note that although these are all transcendental over K, they are not indeterminates and satisfy algebraic relationships between each other, such as the Weierstrass equation $Y^2 = X^3 + \alpha X + \beta$ or the relationship that Y is uniquely determined by X up to sign.

We start with an arbitrary derivation d over K on the function field $K(\mathcal{E})$. Since $Y^2 - X^3 - \alpha X - \beta = 0$, we have $dY = \frac{3X^2+\alpha}{2Y} dX$. Write $Y' = \frac{3X^2+\alpha}{2Y}$ for short. Define the following as functions of $P = (X_P, Y_P)$ and $Q = (X_Q, Y_Q)$ to compute the challenge slope.

$$\Delta X = X_Q - X_P \text{ and } \Delta Y = Y_Q - Y_P, \text{ so } \Lambda = \frac{\Delta Y}{\Delta X}$$

Now we can apply the derivation d to compute the derivative of the challenge slope.

$$
\begin{aligned}
d\Lambda =& \frac{\Delta X d\Delta Y - \Delta Y d\Delta X}{(\Delta X)^2} \\
=& \frac{d\Delta Y - \Lambda d\Delta X}{\Delta X} \\
=& \frac{(dY_Q - \Lambda dX_Q) - (dY_P - \Lambda dX_P)}{\Delta X} \\
=& \frac{\left(Y'_Q - \Lambda\right) dX_Q - \left(Y'_P - \Lambda\right) dX_P}{\Delta X}
\end{aligned} \tag{B.1}
$$

Set the challenge line intercept: $M = Y_P - \Lambda X_P$. Then we have the following derivative of the challenge intercept.

$$
\begin{aligned}
dM =& dY_P - \Lambda dX_P - X_P d\Lambda \\
=& \left(Y'_P - \Lambda\right) dX_P - X_P d\Lambda \\
=& \left(Y'_P - \Lambda\right) dX_P - X_P \frac{\left(Y'_Q - \Lambda\right) dX_Q - \left(Y'_P - \Lambda\right) dX_P}{\Delta X} \\
=& \left(Y'_P - \Lambda\right) dX_P - \frac{X_P}{\Delta X}\left(Y'_Q - \Lambda\right) dX_Q + \frac{X_P}{\Delta X}\left(Y'_P - \Lambda\right) dX_P \\
=& \left(1 + \frac{X_P}{\Delta X}\right)\left(Y'_P - \Lambda\right) dX_P - \frac{X_P}{\Delta X}\left(Y'_Q - \Lambda\right) dX_Q \\
=& \frac{X_Q}{\Delta X}\left(Y'_P - \Lambda\right) dX_P - \frac{X_P}{\Delta X}\left(Y'_Q - \Lambda\right) dX_Q
\end{aligned} \tag{B.2}
$$

We note that by Vieta's formula (or simply by expanding), we have the following for X_R and its derivative.

$$
X_R = \Lambda^2 - X_P - X_Q \quad \Rightarrow \quad dX_R = 2\Lambda d\Lambda - dX_P - dX_Q
$$

Similarly for Y_R and its derivative.

$$
Y_R^2 = X_R^3 + \alpha X_R + \beta \quad \Rightarrow \quad dY_R = \frac{3X_R^2 + \alpha}{2Y_R} dX_R
$$

Now we restrict our attention to a derivation d such that $d(Y_T - \lambda X_T) = 1$ for a fixed $\lambda \in K$ and for $T \in \{P, Q\}$. Ignoring subscripts, we get the following.

$$
1 = dY - \lambda dX \quad \Rightarrow \quad dX = (Y' - \lambda)^{-1}
$$

By applying the above to Equation B.1 and B.2, one can obtain that $d\Lambda = dM = 0$, so

$$\begin{aligned} dX_R =& - dX_P - dX_Q \\ =& - \frac{1}{Y'_P - \lambda} - \frac{1}{Y'_Q - \lambda} \end{aligned} \tag{B.3}$$

$$\begin{aligned} dY_R =& \frac{3X_R^2 + \alpha}{2Y_R} (-dX_P - dX_Q) \\ =& - \frac{3X_R^2 + \alpha}{2Y_R} \left(\frac{1}{Y'_P - \lambda} + \frac{1}{Y'_Q - \lambda} \right) \end{aligned} \tag{B.4}$$

Now we can compute $df(P)$, $df(Q)$, and $df(R)$ as follows. Note our abuse of notation; by $df(P)$, we mean $(d(f))(P)$. That is, the specific function of df evaluated specifically at P rather than the derivation of $f(P)$, which would obviously be 0.

$$\begin{aligned} df(P) =& f_X(P) dX_P + f_Y(P) dY_P \\ =& (f_X(P) + f_Y(P) Y'_P) dX_P \\ =& \frac{f_X(P) + f_Y(P) Y'_P}{Y'_P - \lambda} \end{aligned} \tag{B.5}$$

$$\begin{aligned} df(Q) =& (f_X(Q) + f_Y(Q) Y'_Q) dX_Q \\ =& \frac{f_X(Q) + f_Y(Q) Y'_Q}{Y'_Q - \lambda} \end{aligned} \tag{B.6}$$

$$\begin{aligned} df(R) =& (f_X(R) + Y'_R f_Y(R)) dX_R \\ =& - (f_X(R) + Y'_R f_Y(R)) \cdot \left(\frac{1}{(Y'_P - \lambda)} + \frac{1}{(Y'_Q - \lambda)} \right) \end{aligned} \tag{B.7}$$

We now compute $\delta(\widehat{f})$ as a rational function of (X_P, Y_P, X_Q, Y_Q).

$$\begin{aligned} \delta(\widehat{f}) =& \frac{df(P)}{f(P)} + \frac{df(Q)}{f(Q)} + \frac{df(R)}{f(R)} \\ =& \frac{f_X(P) + f_Y(P) Y'_P}{(Y'_P - \lambda) f(P)} + \frac{f_X(Q) + f_Y(Q) Y'_Q}{(Y'_Q - \lambda) f(Q)} \\ & - \frac{f_X(R) + f_Y(R) Y'_R}{f(R)} \cdot \left(\frac{1}{(Y'_P - \lambda)} + \frac{1}{(Y'_Q - \lambda)} \right) \end{aligned} \tag{B.8}$$

This is the left-hand side of the verification equation. We can now evaluate Λ and M at any pair of points on the curve $P, Q \in \mathcal{E}$ to obtain $\lambda = \Lambda(x_P, y_P, x_Q, y_Q)$ and $\mu^* = M(x_P, y_P, x_Q, y_Q)$, and then substitute λ for Λ and μ^* for M.

This is the left-hand side of the verification equation. We now evaluate $\delta(\widehat{f})$ at the tuple

$(X_P, Y_P, X_Q, Y_Q) = (x_P, y_P, x_Q, y_Q)$, setting $\lambda = \Lambda(x_P, y_P, x_Q, y_Q)$, $y'_P = \frac{3x_P^2+\alpha}{2y_P}$, and $y'_Q = \frac{3x_Q^2+\alpha}{2y_Q}$ to obtain the following.

$$\begin{aligned}\delta(\widehat{f})\,\big|_{P,Q} = &\frac{f_X(P) + y'_P f_Y(P)}{f(P)(y'_P - \lambda)} - \frac{2\lambda(f_X(R) + y'_R f_Y(R))(y'_P - \lambda - \Delta X)}{f(R)\Delta X(y'_P - \lambda)} \\ &+ \frac{f_X(Q) + y'_Q f_X(Q)}{f(Q)(y'_Q - \lambda)} + \frac{2\lambda(f_X(R) + y'_R f_Y(R))(y'_Q - \lambda - \Delta X)}{f(R)\Delta X(y'_Q - \lambda)} \end{aligned} \tag{B.9}$$

This comes directly from the following.

$$\begin{aligned}&\frac{df(P)}{f(P)} + \frac{df(Q)}{f(Q)} + \frac{df(R)}{f(R)} \\ &= \left[\frac{f_X(P) + y'_P f_Y(P)}{f(P)} - \frac{f_X(R) + y'_R f_Y(R)}{f(R)}\right] \cdot \frac{1}{y'_P - \lambda} &\text{(B.10)} \\ &\quad + \left[\frac{f_X(Q) + y'_Q f_Y(Q)}{f(Q)} - \frac{f_X(R) + y'_R f_Y(R)}{f(R)}\right] \cdot \frac{1}{y'_Q - \lambda} &\text{(B.11)}\end{aligned}$$

Then B.10 equals the term

$$\frac{f_X(P)f(R) + y'_P f_Y(P)f(R) - f_X(R)f(P) - y'_R f_Y(R)f(P)}{f(P)f(R)(y'_P - \lambda)}$$

and B.11 equals the term

$$\frac{f_X(Q)f(R) + y'_Q f_Y(Q)f(R) - f_X(R)f(Q) - y'_R f_Y(R)f(Q)}{f(Q)f(R)(y'_Q - \lambda)}$$

Now, the least common multiple for the denominators in Equations B.10 and B.11 is $f(P)f(Q)f(R)(y'_P - \lambda)(y'_Q - \lambda)$, so this will be the denominator of their sum in reduced form, while the numerator will be the following.

$$\begin{aligned}&f(Q)f(R)f_X(P)y'_Q + f(Q)f(R)f_Y(P)y'_P y'_Q - f(P)f(Q)f_X(R)y'_Q - f(P)f(Q)f_Y(R)y'_Q y'_R \\ &- f(Q)f(R)f_X(P)\lambda - f(Q)f(R)f_Y(P)y'_P\lambda + f(P)f(Q)f_X(R)\lambda + f(P)f(Q)f_Y(R)y'_R\lambda \\ &+ f(P)f(R)f_X(Q)y'_P + f(P)f(R)f_Y(Q)y'_P y'_Q - f(P)f(Q)f_X(R)y'_P - f(P)f(Q)f_Y(R)y'_P y'_R \\ &- f(P)f(R)f_X(Q)\lambda - f(P)f(R)f_Y(Q)y'_Q\lambda + f(P)f(Q)f_X(R)\lambda + f(P)f(Q)f_Y(R)y'_R\lambda\end{aligned}$$

Certainly $y'_T = \frac{3x_T^2+\alpha}{2y_T}$, so we can substitute into each of the equations given above. To simplify the expressions, we factor out $\frac{1}{4y_P y_Q y_R}$ to clear the denominators and use $\Gamma_T = 3x_T^2 + \alpha$ as defined in step 7 of the `Prove` algorithm and step 3 of the `Verify` algorithm. Then the numerator takes

the following beastly expression.

$$
\begin{aligned}
&f(Q)f(R)f_X(P)2y_py_R\Gamma_Q + f(Q)f(R)f_Y(P)y_R\Gamma_P\Gamma_Q - f(P)f(Q)f_X(R)2y_Py_R\Gamma_Q \\
&- f(P)f(Q)f_Y(R)y_P\Gamma_Q\Gamma_R - f(Q)f(R)f_X(P)4\lambda y_Py_Qy_R - f(Q)f(R)f_Y(P)2\lambda y_Qy_R\Gamma_P \\
&+ f(P)f(Q)f_X(R)4\lambda y_Py_Qy_R + f(P)f(Q)f_Y(R)2\lambda y_Py_Q\Gamma_R + f(P)f(R)f_X(Q)2y_Qy_R\Gamma_P \\
&+ f(P)f(R)f_Y(Q)y_R\Gamma_P\Gamma_Q - f(P)f(Q)f_X(R)2y_Ry_Q\Gamma_P - f(P)f(Q)f_Y(R)y_Q\Gamma_P\Gamma_R \\
&- f(P)f(R)f_X(Q)4\lambda y_Py_Qy_R - f(P)f(R)f_Y(Q)2\lambda y_Py_R\Gamma_Q + f(P)f(Q)f_X(R)4\lambda y_Py_Qy_R \\
&+ f(P)f(Q)f_Y(R)2\lambda y_Py_Q\Gamma_R
\end{aligned} \tag{B.12}
$$

Since the denominator is $4y_Py_Ry_Qf(P)f(Q)f(R)(y'_P-\lambda)(y'_Q-\lambda)$ and $2y_Ty'_T=\Gamma_T$, we reduce by distributing $2y_P$ and $2y_Q$:

$$y_Rf(P)f(Q)f(R)(\Gamma_P-2\lambda y_P)(\Gamma_Q-2\lambda y_Q)$$

Now, take the numerator terms and group them by the derivatives:

$$f(P)f(Q)f_X(R)(8\lambda y_Py_Qy_R - 2y_Py_R\Gamma_Q - 2y_Ry_Q\Gamma_P) \tag{B.13}$$
$$f(P)f(Q)f_Y(R)(4\lambda y_Py_Q\Gamma_R - y_P\Gamma_Q\Gamma_R - y_Q\Gamma_P\Gamma_R) \tag{B.14}$$
$$f(Q)f(R)f_X(P)(2y_Py_R\Gamma_Q - 4\lambda y_Py_Qy_R) \tag{B.15}$$
$$f(Q)f(R)f_Y(P)(y_R\Gamma_P\Gamma_Q - 2\lambda y_Qy_R\Gamma_P) \tag{B.16}$$
$$f(P)f(R)f_X(Q)(2y_Qy_R\Gamma_P - 4\lambda y_Py_Qy_R) \tag{B.17}$$
$$f(P)f(R)f_Y(Q)(y_R\Gamma_P\Gamma_Q - 2\lambda y_Py_R\Gamma_Q) \tag{B.18}$$

Thus, the verification equation is the sum of terms B.13 through B.18 above, divided by the denominator:

$$y_Rf(P)f(Q)f(R)(\Gamma_P-2\lambda y_P)(\Gamma_Q-2\lambda y_Q).$$

Finally, we notice that we can rewrite the denominator, by replacing λ with $\frac{\Delta Y}{\Delta X}$, as:

$$\frac{1}{(\Delta X)^2}\cdot(y_Rf(P)f(Q)f(R)(\Gamma_P\Delta X-2y_P\Delta Y)(\Gamma_Q\Delta X-2y_Q\Delta Y))$$

Now, we multiply the numerator and denominator by $(\Delta X)^2$, to clear this division. This will turn terms B.13 through B.18 into the following, where we highlight like terms on the right hand side:

$$2y_R\Delta X f(P)f(Q)f_X(R)(4y_Py_Q\Delta Y - y_P\Gamma_Q\Delta X - y_Q\Gamma_P\Delta X) \tag{B.19}$$

$$\Gamma_R\Delta X f(P)f(Q)f_Y(R)(4y_Py_Q\Delta Y - y_P\Gamma_Q\Delta X - y_Q\Gamma_P\Delta X) \tag{B.20}$$

$$2y_Py_R\Delta X f(Q)f(R)f_X(P)(\Gamma_Q\Delta X - 2y_Q\Delta Y) \tag{B.21}$$

$$y_R\Gamma_P\Delta X f(Q)f(R)f_Y(P)(\Gamma_Q\Delta X - 2y_Q\Delta Y) \tag{B.22}$$

$$2y_Qy_R\Delta X f(P)f(R)f_X(Q)(\Gamma_P\Delta X - 2y_P\Delta Y) \tag{B.23}$$

$$y_R\Gamma_Q\Delta X f(P)f(R)f_Y(Q)(\Gamma_P\Delta X - 2y_P\Delta Y) \tag{B.24}$$

Then $\delta(\widehat{f})\,\big|_{P,Q}$ is the sum of terms B.19 through B.24, divided by the following:

$$y_R f(P)f(Q)f(R)(\Gamma_P\Delta X - 2y_P\Delta Y)(\Gamma_Q\Delta X - 2y_Q\Delta Y).$$

Thus, we have obtained an expression for $\delta(\widehat{f})\,\big|_{P,Q}$ as a rational function.

B.2 Logarithmic Derivative of a Line at the Zeroes and Poles of a Function Field Element

In this section, we consider the flip side of Weil reciprocity. Rather than looking at $f(P)f(Q)f(R)$ for a given f as a function of P and Q, we look at $\prod_T \ell(T)^{\nu(T)}$ as a function of coefficients of ℓ.

Indeed, the other side of Weil reciprocity is $\widehat{\ell}(\mathrm{div}(f)) = \prod_T \ell(T)^{\nu(T)}$. As above, consider the function $\prod_T \ell(T)^{\nu(T)}$ as an evaluation of some $\widehat{\ell}(X_P, Y_P, X_Q, Y_Q) \in F$. Let $\widetilde{\ell} = Y - \Lambda X - M$, where Λ and M are as in Appendix B.1. Say $\mathrm{div}(f)$ is the principal divisor $\sum_{i=1}^n \nu_i[T_i] - \nu_0[O]$ where $\nu_0 = \sum_{i=1}^n \nu_i$ and each $T_i = (x_i, y_i) \in \mathcal{E}$. Define $\widetilde{\ell}_i = \widetilde{\ell}(x_i, y_i, X_P, Y_P, X_Q, Y_Q) = y_i - \Lambda x_i - M$. Define $\underline{\ell} = \prod_{i=1}^n \widetilde{\ell}_i(T_i)^{\nu_i}$. Then $\ell(\mathrm{div}(f))$ is an evaluation of $\widehat{\ell}$ at $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$.

$$
\begin{aligned}
\underline{\ell} &= \prod_{i=1}^{n} \widetilde{\ell}_i(T_i)^{\nu_i} \\
&= \prod_{i=1}^{n} (y_i - \Lambda x_i - M)^{\nu_i} \\
\delta(\underline{\ell}) &= \sum_{i=1}^{n} \nu_i \frac{d\,(y_i - \Lambda x_i - M)}{y_i - \Lambda x_i - M} \\
&= \sum_i \nu_i \frac{dy_i - \Lambda dx_i - x_i d\Lambda - dM}{y_i - \Lambda x_i - M} \\
&= \sum_i \nu_i \frac{(y_i' - \Lambda) dx_i - x_i d\Lambda - dM}{y_i - \Lambda x_i - M} \\
&= \sum_i \nu_i \frac{1 - x_i d\Lambda - dM}{y_i - \Lambda x_i - M} \\
&= \sum_i \nu_i (y_i - \Lambda x_i - M)^{-1} \\
&= \sum_i \nu_i \Delta X (y_i \Delta X - x_i \Delta Y - M \Delta X)^{-1}
\end{aligned} \tag{B.25}
$$

where we refer to Appendix B.1 for Λ, M, $d\Lambda$, and dM, respectively. These x_i, y_i are fixed elements of K, and d is a derivation over K, so $dx_i = dy_i = 0$, explaining the last step.

C A Specific Witness Construction Function

Earlier we "black-boxed" the witness construction function to allow flexibility in implementation. In Algorithm 4, we provide a specific instantiation of a witness construction function, which uses Algorithm 3 as a subroutine. This approach was described in [Eag22] and implemented in Sage (see [Eag]), and recreated in Rust (see [Parc]). For newer implementations, we suggest other interpolation methods such as barycentric interpolation, but modified for constant-time execution.

$\texttt{line}(P, Q) \to \texttt{out} \in \{\perp\} \cup K(\mathcal{E})$
If $P \notin \mathcal{E}$ or $Q \notin \mathcal{E}$, return $\perp$.
Elif $P = Q = O$ return $1 \in K$.
Elif $P = O$ and $Q \neq O$, return $\texttt{line}(Q, Q)$.
Elif $P \neq O$ and $Q = O$, return $\texttt{line}(P, P)$.
Elif $P = Q$:
Set $\lambda = \frac{2y_P}{3x_P^2 + \alpha}$
Elif $P = -Q$:
Set $\lambda = 0$
Else:
Compute:
$\Delta y = y_Q - y_P$
$\Delta x = x_Q - x_P$
$\lambda = \frac{\Delta y}{\Delta x}$
Return (P, λ).

Algorithm 3: Input points $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$, and outputs the point-slope representation of a degree 1 polynomial corresponding to the group law statement $P + Q + (-(P + Q)) = O$. If $P = Q$, the line passes through P and is tangent to $\mathcal{E}$; if $P = -Q$ or $P = O$, the line is vertical and passes through both P and Q; otherwise, the line interpolates P and Q as usual.

$\texttt{Wit}(P_0, P_1, P_2, \ldots, P_{n-1}) \to \texttt{out} \in \{\perp\} \cup K(\mathcal{E})$

If $n = 0$ **then return** $\perp$.
If $(P_0 + P_1 + \cdots + P_{n-1}) \neq O$ **then return** $\perp$.
If $P_j = O$ for any j **then return** $\perp$.
Initialize $\texttt{queue} = \emptyset$.
Let $j = 0$.
While $j < n$:
  If $j + 1 < n$:
    Set pair $(A_{j/2}, B_{j/2}) = (P_j, P_{j+1})$.
    Set degree $m_j = 2$.
  Else:
    Set pair $(A_{j/2}, B_{j/2}) = (P_j, O)$.
    Set degree $m_j = 1$.
  Compute:
    $T_{j/2} = A_{j/2} + B_{j/2} \in \mathcal{E}$,
    $\texttt{out} \leftarrow \texttt{line}(A_{j/2}, B_{j/2})$.
    If $\texttt{out} = \perp$, output $\perp$ and terminate.
    Else:
      Parse $(U_j, \lambda_j) \leftarrow \texttt{out}$.
      $(w_j, z_j) \leftarrow U_j$.
      Form $\ell_j = (Y - z_j) - \lambda_j(X - w_j)$.
  Enqueue $\texttt{queue} \leftarrow (m_j, T_j, \ell_j)$.
  Increment j by 2.
While $\texttt{len}(\texttt{queue}) > 1$:
  Initialize $\texttt{nextQueue} = \emptyset$.
  If `len(queue)` is odd:
    Dequeue $x \leftarrow \texttt{queue}$.
    Enqueue $\texttt{nextQueue} \leftarrow x$.
  While $\texttt{len(queue)} \geq 2$:
    Dequeue (m_a, T_a, f_a), $(m_b, T_b, f_b) \leftarrow \texttt{queue}$.
    Compute:
      $T_{a,b} = T_a + T_b$,
      $\ell_{a,b} = \texttt{line}(T_a, T_b)$,
      $\texttt{numerator} = f_a \cdot f_b \cdot \ell_{a,b} \mod e(X, Y)$,
      $\texttt{denominator} = \texttt{line}(T_a, -T_a) \cdot \texttt{line}(T_b, -T_b)$,
      $m_{a,b} = m_a + m_b$.
      $g = \frac{\texttt{numerator}}{\texttt{denominator}}$.
    Enqueue $\texttt{nextQueue} \leftarrow (m_{a,b}, T_{a,b}, g)$.
  Set $\texttt{queue} = \texttt{nextQueue}$.
Pop final element (m, T, f) from queue.
If $T \neq O$ or $m \neq n$, output $\perp$ and terminate.
Output f.

Algorithm 4: Constructing a rational function whose divisor interpolates a set of elliptic curve points. Based on [Para].

Note that, in `Wit`, we compute $\frac{n}{2}$ group operations during pre-computation, and then in the

first inner loop we compute $\frac{n}{4}$ operations, then $\frac{n}{8}$, then $\frac{n}{16}$, and so on. In particular, `Wit` takes up to $\sum_{i=1}^{\infty} \frac{n}{2^i} \leq n$ group operations. For each of these, up to 3 queries are made to `line`, 3 products in $K(\mathcal{E})$ are computed, and one division in $K(\mathcal{E})$. Also, in the first inner loop, we compute $\frac{n}{4}$ sums of degrees, then $\frac{n}{8}$, and so on, so we have at most $\frac{n}{2}$ sums of degrees; we interpret these as sums in K.

Lemma C.0.1. Let $n \geq 3$ and let $\{P_0, P_1, \ldots, P_{n-1}\} \subseteq \mathbb{G}^n$ be a multi-subset of non-identity points such that $\sum_i P_i = O$ and $P_i + P_j \neq O$ for any i, j. Then $\texttt{Wit}(P_0, P_1, P_2, \ldots, P_{n-1})$ does not fail, runs in time $t = O(n)$, and outputs some f such that there exists $m \in \mathbb{Z}$ satisfying $\mathrm{div}(f) = \sum_{i=0}^{n-1}[P_i] - m[O]$.

Proof. Pre-processing in `Wit` prepares the following $\lceil \frac{n}{2} \rceil$ pairs:

$$\begin{cases} (P_0, P_1), (P_2, P_3), \ldots, (P_{n-2}, P_{n-1}); & n \text{ even} \\ (P_0, P_1), (P_2, P_3), \ldots, (P_{n-1}, O); & n \text{ odd} \end{cases}$$

and computes the interpolating line $\ell_j = \texttt{line}(A_j, B_j)$ for each pair (A_j, B_j) with $0 \leq j \leq \lceil \frac{n}{2} \rceil$. Then, the point $T_j = A_j + B_j$ is computed, the degree m_j is computed (which is 2 when $A_j \neq B_j$ are both non-identity and 1 when $A_j = O$ or $B_j = O$). Each triple (m_j, T_j, ℓ_j) is enqueued.

Once this queue is constructed, the primary loop begins. In the while-loop, `Wit` pops pairs of triples at a time and merges them. Note that for any one of these initial lines, $\mathrm{div}(\ell_j) = [A_j] + [B_j] + [-(A_j + B_j)]$ (disregarding the $[O]$ term to be concise).

We claim that all the triples which enter the queue are of the form (m, T, f) where $T = \sum_i \Omega_i$ for some non-identity points $\Omega_i \in \mathbb{G}(\mathcal{E})$, and such that $\mathrm{div}(f) = \sum_i [\Omega_i] + [-\sum_i \Omega_i]$. Certainly the initial lines satisfy this. Moreover, for each (m_f, T_f, f), (m_g, T_g, g) in the queue, say $T_f = \sum_i \Omega_i$ and $T_g = \sum_j \Theta_j$ for $\Theta_i \in \mathbb{G}(\mathcal{E})$, write the non-identity terms of the divisors $\mathrm{div}(f) = \sum_i [\Omega_i] + [-T_f]$ and $\mathrm{div}(g) = \sum_j [\Theta_j] + [-T_g]$ and notice the merged function field element is $h = fg\frac{\texttt{line}(T_f, T_g)}{\texttt{line}(T_f, -T_f)\texttt{line}(T_g, -T_g)}$. The new degree $m_f + m_g$ is computed, and $(m_f + m_g, T_f + T_g, h)$ is enqueued. For this h, we note that we have the following.

$$\begin{aligned}
\mathrm{div}(h) =& \mathrm{div}\left(fg\frac{\texttt{line}(T_f, T_g)}{\texttt{line}(T_f, -T_f)\texttt{line}(T_g, -T_g)}\right) \\
=& \mathrm{div}(f) + \mathrm{div}(g) + \mathrm{div}(\texttt{line}(T_f, T_g)) - \mathrm{div}(\texttt{line}(T_f, -T_f)) - \mathrm{div}(\texttt{line}(T_f, -T_f)) \\
=& \sum_i [\Omega_i] + [-T_f] + \sum_j [\Theta_j] + [-T_g] + [T_f] + [T_g] \\
& + [-(T_f + T_g)] - [T_f] - [-T_f] - [T_g] - [-T_g] \\
=& \sum_i [\Omega_i] + \sum_j [\Theta_j] + [-(T_f + T_g)] \\
=& \sum_i [\Omega_i] + \sum_j [\Theta_j] + \Big[- \big(\sum_i \Omega_i + \sum_j \Theta_j \big) \Big]
\end{aligned}$$

Hence, every entry of `queue` is a triple $(m, T, f) \in \mathbb{N} \times \mathbb{G}(\mathcal{E}) \times K(\mathcal{E})$ such that $\mathrm{div}(f) = \sum_i [P_i] + [-\sum_i P_i] - m[O]$ as claimed.

Furthermore, by inspection, we see that, regardless of the order in which these merge operations take place, we eventually obtain a function field element h with divisor $\sum_i [P_i] + [-(\sum_i P_i)]$. If the input points $\{P_i\}_i$ are a nontrivial decomposition of O, then $[-(\sum_i P_i)] = [O]$, so $\mathrm{div}(h) = \sum_i [P_i]$ as desired. □

Corollary C.0.2. Under the assumptions of Lemma C.0.1, the output f is not just a rational function, it is a polynomial (the degree of the denominator is 0).

Proof. Note that, while the fraction $\frac{\ell_{2i}\ell_{2i+1}\widehat{\ell_i}}{\widetilde{\ell_{2i}}\widetilde{\ell_{2i+1}}}$ is in the function field, the factorization described in the proof of Lemma C.0.1 guarantees that $\frac{\ell_{2i}\ell_{2i+1}\widehat{\ell_i}}{\widetilde{\ell_{2i}}\widetilde{\ell_{2i+1}}}$ reduces to a polynomial. This is true for every iteration. □

The implementation in [Para] performs division, and if the remainder is non-zero, the algorithm fails and terminates; Corollary C.0.1 demonstrates this check is unnecessary. In the sequel, we assume witnesses output from `Wit` have unit denominators, i.e. if $f = \frac{f_1(X)+f_2(X)Y+(e(X,Y))}{f_3(X)+(e(X,Y))}$, then $\deg(f) = 0$. Over a field, we can assume without losing generality that $f_3(X) + (e(X, Y)) = 1 + (e(X, Y))$, so we write witnesses as $f = f_1(X) + f_2(X)Y$ (leaving coset notation implicit).

References

[Bas] Alp Bassa. Soundness Proof for an Interactive Protocol for the Discrete Logarithm Relation.

[Bas25] Alp Bassa. Personal communication, 2025.

[BBB+17] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short Proofs for Confidential Transactions and More. Cryptology ePrint Archive, Paper 2017/1066, 2017.

[bgg] b-g goodell. silver-bullet. `https://github.com/cypherstack/silver-bullet`. Accessed: 2025-09-22.

[BPW16] David Bernhard, Olivier Pereira, and Bogdan Warinschi. How not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. Cryptology ePrint Archive, Paper 2016/771, 2016. `https://eprint.iacr.org/2016/771`.

[BT] Jean-Paul Berrut and Lloyd N. Trefethen. Barycentric Lagrange Interpolation. `https://people.maths.ox.ac.uk/trefethen/barycentric.pdf`. Accessed: 2025-07-19.

[CP93] David Chaum and Torben Pryds Pedersen. Wallet Databases with Observers. In Ernest F. Brickell, editor, *Advances in Cryptology — CRYPTO' 92*, pages 89–105, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[DMWG23] Quang Dao, Jim Miller, Opal Wright, and Paul Grubbs. Weak Fiat-Shamir Attacks on Modern Proof Systems. Cryptology ePrint Archive, Paper 2023/691, 2023. https://eprint.iacr.org/2023/691.

[Dum] Evan Dummit. Number Theory (part 7): Elliptic Curves. https://dummit.cos.northeastern.edu/docs/numthy_7_elliptic_curves.pdf. Accessed: 2025-07-18.

[Eag] Liam Eagen. ECIP Calculations. gist.github.com/Liam-Eagen/666d0771f4968adccd6087465b8c5bd4. Accessed: 2025-09-22.

[Eag22] Liam Eagen. Zero Knowledge Proofs of Elliptic Curve Inner Products from Principal Divisors and Weil Reciprocity. Cryptology ePrint Archive, Paper 2022/596, 2022.

[fab] fabrizio-m. fcmp-competition/ec-divisors-contest/ec-divisors-contest-src/notes. https://github.com/fabrizio-m/fcmp-competition/blob/notes/ec-divisors-contest/ec-divisors-contest-src/notes.md#evaluation-and-interpolation. Accessed: 2025-07-20.

[FS87] Amos Fiat and Adi Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-47721-7_12.

[Har13] Robin Hartshorne. *Algebraic Geometry*, volume 52. Springer Science & Business Media, 2013.

[HKTT08] James William Peter Hirschfeld, Gábor Korchmáros, Fernando Torres, and Fernando Torres. *Algebraic curves over a finite field*, volume 20. Princeton University Press, 2008.

[JF21] Aram Jivanyan and Aaron Feickert. Lelantus Spark: Secure and Flexible Private Transactions. Cryptology ePrint Archive, Paper 2021/1173, 2021.

[Jon] Marvin Jones. Vac 101: Transforming an Interactive Protocol to a Noninteractive Argument. https://vac.dev/rlog/vac101-fiat-shamir/. Accessed: 2024-10-15.

[LW05] Joseph K. Liu and Duncan S. Wong. Linkable Ring Signatures: Security Models and New Schemes. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *Computational Science and Its Applications – ICCSA 2005*, pages 614–623, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[Mil] Jim Miller. The Frozen Heart Vulnerability in Bulletproofs. https://blog.trailofbits.com/2022/04/15/the-frozen-heart-vulnerability-in-bulletproofs/. Accessed: 2024-10-15.

[Mon25] Monero Project. Fcmp++ contest, April 2025. Accessed: 2025-07-19.

[MS25] Felice Manganiello and Freeman Slaughter. HammR: A ZKP Protocol for Fixed Hamming-Weight Restricted-Entry Vectors. Cryptology ePrint Archive, Paper 2025/475, 2025.

[NHB24] Hieu Nguyen, Uyen Ho, and Alex Biryukov. Fiat-Shamir in the Wild. Cryptology ePrint Archive, Paper 2024/1565, 2024. https://eprint.iacr.org/2024/1565.

[Oka93] Tatsuaki Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In Ernest F. Brickell, editor, *Advances in Cryptology — CRYPTO' 92*, pages 31–53, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[Para] Luke "Kayaba" Parker. FCMP++ Paper. https://github.com/kayabaNerve/fcmp-plus-plus-paper. Accessed: 2025-05-25.

[Parb] Luke "Kayaba" Parker. Full-Chain Membership Proofs Development. https://www.getmonero.org/2024/04/27/fcmps.html. Accessed: 2025-06-11.

[Parc] Luke "Kayba" Parker. Fcmp++ implementation'. https://github.com/kayabaNerve/fcmp-plus-plus/tree/78754718faa21f0a5751fbd30c9495d7f7f5c2b1/crypto/divisors/src. Accessed: 2025-09-22.

[Pro] Salvium Protocol. Salvium. https://salvium.io/. Accessed: 2025-06-03.

[Sch91] C. P. Schnorr. Efficient Signature Generation by Smart Cards. *J. Cryptol.*, 4(3):161–174, January 1991. https://doi.org/10.1007/BF00196725.

[Sil09] Joseph H Silverman. *The Arithmetic of Elliptic Curves*, volume 106. Springer, 2009.

[Tar] TariLabs. The MuSig Schnorr Signature Scheme. https://tlu.tarilabs.com/cryptography/The_MuSig_Schnorr_Signature_Scheme. Accessed: 2025-07-21.

[Var] Various. Monero Research Lab chat logs. https://libera.monerologs.net/monero-research-lab/20250716#c542261. Accessed: 2025-09-22.