SLVer Bullet: Straight-Line Verification for Bulletproofs

Brandon Goodell, Rigo Salazar, Freeman Slaughter, Luke Szramowski

Cypher Stack

June 12, 2025

Disclaimer

This paper is not yet completed; future updates to this document are forthcoming soon.

1 Introduction

In [Eag22], Eagen proposed a method for checking whether a sum of points in an elliptic curve group has been computed correctly. The approach offloads computational costs faced by verifiers to provers, which is useful when verification must be performed repeatedly. Although the iterative witness construction algorithm proposed in [Eag22] is correct, [Eag22] is rather informal, lacking formal protocol descriptions, claims, proofs, or efficiency analyses. In [Para], Parker used Eagen's method in rank-1 constraint systems to verify scalar multiplication of group elements, proposing a protocol based on Eagen's arguments. In [Bas], Bassa contributed towards formalizing Eagen's method, explicitly describing Parker's proof of scalar multiplication and sketching arguments towards proofs of soundness.

However, the soundness arguments in [Bas] are not without obstacles. In [Eag22], rational solutions to certain systems of polynomial equations are assumed to exist. Unfortunately, these equations do not admit such solutions in general, so the verification equations proposed in [Eag22] and studied in [Bas] are not justified. Bassa lifted to the surface of pairs of elliptic curve group points, partially resolving the problem.

Bassa's approach implies changes to the verification equations of [Eag22], however. We formalize the correct verification equations, and consider the security of Eagen's approach. We present a corrected version of the general protocol, and explicitly compute the completeness and soundness errors. We show how our corrected scheme can be used to verify scalar multiplication as in Parker's proposed protocol. We then apply this approach to zero-knowledge proof systems to illustrate the computational advantage offered by divisors. In particular, the prover can pre-compute division operations for the verifier, which are the computational bottlenecks on the verifier's side. In particular, we transform the interactive Schnorr identification protocol [Sch91] and an interactive Bulletproofs

range-proving protocol [BBB+17], and benchmark efficiency. We choose Schnorr and Bulletproofs because these protocols are simple, secure, popular, and readily applicable in cryptocurrencies such as Monero or Salvium.

1.1 Change log

This document will be updated occasionally, especially when security-sensitive results come to light. We summarize such changes here.

- 11 June 2025. Initial preprint.

2 Notation and preliminary definitions

We begin assuming the reader has knowledge of basic algebra concepts related to groups, rings, and fields. Our notation and background primarily follows [Sil09], but we depart notationally in a few notable ways, especially with regard to divisors.

2.1 Polynomials and rational functions

For a set S, we use $x \xleftarrow{\$} S$ to mean that the element x was sampled uniformly at random from set S. Let $\mathbb{1}$ and $\mathbb{0}$ be the vectors of all 1's and 0's, respectively. If R is a ring, we write $R^\times = R \setminus \{0\}$ to denote the multiplicative subgroup. Let p be a prime, K a finite field with $\mathrm{char}(K) = p$, algebraic closure $\overline{K}$, and multiplicative subgroup $K^\times = \{x \in K \mid x \neq 0\}$. Let X, Y, Z be indeterminates over $\overline{K}$. 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.

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

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

We also define the respective field extensions of rational functions $\overline{K}(X,Y)$, $K(X,Y)$, $\overline{K}(X,Y,Z)$, and $K(X,Y,Z)$ as usual. Given $P = (x,y) \in \overline{K}^2$, the following is a well-defined map on rational

functions $f \in \overline{K}(X,Y)$.

$$\mathrm{ord}_P(f) = \begin{cases} m > 0; & P \text{ is a root of } f \text{ with multiplicity } m \\ m < 0; & P \text{ is a pole of } f \text{ with multiplicity } |m| = -m \\ 0; & \text{otherwise} \end{cases} \tag{1}$$

We define $\mathrm{ord}_P(f)$ similarly for $f \in K(X,Y)$.

If $d, n \in \mathbb{N}_0$ and $f = \sum_{ij} a_{ij} X^i Y^j$ has some $\lambda \in K^\times$ such 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 $K[X,Y,Z]$ such that every $f(X,Y,Z) \in K[X,Y,Z]$ relates to $f(\lambda X, \lambda Y, \lambda Z)$ for every $\lambda \in K^\times$. 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 $K[X:Y:Z]$. Similarly define $\overline{K}[X:Y:Z]$. 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 $\overline{K}[X,Y]$, $K[X,Y]$, $\overline{K}[X,Y,Z]$, $K[X,Y,Z]$, $\overline{K}[X:Y:Z]$, and $K[X:Y:Z]$ are all integral domains.

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.

2.2 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 $I(V/K) = \{f \in K[X,Y] \mid f(x,y) = 0 \text{ for all } (x,y) \in V\} = I(V) \cap K[X,Y]$. 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 $K(V)$ 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 $f_1 g_2 - f_2 g_1 = 0$.

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 $(x',y',z') = (\lambda x, \lambda y, \lambda z)$. Use $\mathbb{P}^2(\overline{K})$ to denote 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 $v_p(I) = \{(x:y:z) \in \mathbb{P}^2(\overline{K}) \mid f(x,y,z) = 0 \text{ for all } f \in I\}$. Since the generating set for I contains only homogeneous polynomials, the presence of any representative in $(x:y:z)$ which kills f implies all representatives kill f. 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 $v_p(f)$.

Given a projective vanishing set V, we overload notation and define $I(V)$ to be the homogeneous ideal of $\overline{K}[X:Y:Z]$ generated by the set $\{f \in \overline{K}[X:Y:Z] \mid f(x:y:z) = 0 \text{ for all } (x:y:z) \in V\}$. We say a projective vanishing set V is *defined over* K, denoted with shorthand V/K, if $I(V)$ is generated as an ideal in $\overline{K}[X:Y:Z]$ by a subset of $K[X:Y:Z]$. We say V is a *projective variety* if its homogeneous ideal $I(V)$ is a prime ideal in $\overline{K}[X,Y,Z]$. The *projective coordinate ring* of an projective variety V/K is the quotient ring $K[V] = \frac{K[X:Y:Z]}{I(V/K)}$.

Affine points $P = (x,y) \in \mathbb{A}^2(\overline{K})$ naturally map to projective points $P^* = (x:y:1) \in \mathbb{P}^2(\overline{K})$. This map is injective. 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.

Using these maps, we can make sense of embedding $\mathbb{A}^2$ in $\mathbb{P}^2$, and projecting $\mathbb{P}^2$ back to $\mathbb{A}^2$. 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[1].

2.3 Plane curves

A *plane affine curve* $\mathcal{F}$ is just the affine vanishing set of some $f \in \overline{K}[X,Y]$, i.e. $\mathcal{F} = v_a(f)$. Define $\deg(\mathcal{F}) = \deg(f)$. 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. $\mathcal{F} = v_p(f)$. 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 denote $O = (0:1:0)$. It is also easy to show that $\mathcal{E}$ is a projective variety and its

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

associated function field is the following.

$$K(\mathcal{E}) = \left\{ \frac{f(X,Y) + I(\mathcal{E}/K)}{g(X,Y) + I(\mathcal{E}/K)} \mid f, g \in K[X,Y], g \notin I(\mathcal{E}/K) \right\} \tag{2}$$

Except in the case of confusion or for clarification, we often omit the ideal $I(\mathcal{E}/K)$ from our notation in the sequel, 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 two arbitrary bivariate polynomials is expensive, requiring up to $\frac{1}{2}\left((\deg(f)+1)(\deg(f)+2) + (\deg(g)+1)(\deg(g)+2)\right)$ elements of K. To see why, note there are $\deg(h)+1$ ways to select exponents i, j such that $i + j = \deg(h)$ for any polynomial h, and we must count all monomials with $0 \leq i + j \leq \deg(h)$. Thus, we have $1+2+3+\cdots+\deg(h)+(\deg(h)+1) = \frac{1}{2}(\deg(h)+1)(\deg(h)+2)$ terms to describe in an arbitrary h. Moreover, we must describe f and g both.

For any $f = \sum_{i=0}^{n}\sum_{j=0}^{m} c_{i,j}X^iY^j \in K[X,Y]$, we have the following.

$$\begin{aligned}
f(X,Y) &= \sum_{i=0}^{n}\sum_{j=0}^{m} c_{i,j}X^iY^j && (3)\\
&= \sum_{j=0}^{m} a_j(X)Y^j \text{ where } a_j(X) = \sum_{i=0}^{n} c_{i,j}X^i && (4)\\
&= \sum_{j=0}^{\lfloor m/2 \rfloor} a_{2j}(X)Y^{2j} + a_{2j+1}(X)Y^{2j+1} && (5)\\
&= \sum_{j} Y^{2j}\left(a_{2j}(X) + a_{2j+1}(X)Y\right) && (6)\\
&= \sum_{j} \left(X^3 + \alpha X + \beta\right)^j \left(a_{2j}(X) + a_{2j+1}(X)Y\right) && (7)\\
&= \underbrace{\left(\sum_{j} \left(X^3 + \alpha X + \beta\right)^j a_{2j}(X)\right)}_{a(X)} + \underbrace{\left(\sum_{j} \left(X^3 + \alpha X + \beta\right)^j a_{2j+1}(X)\right)}_{b(X)} Y && (8)\\
&= a(X) + b(X)Y && (9)
\end{aligned}$$

where $a(X) = \sum_{j=0}^{\lfloor m/2 \rfloor}(X^3+\alpha X+\beta)^j a_{2j}(X) \in K[X]$ and $b(X) = \sum_{j=0}^{\lfloor m/2 \rfloor}(X^3+\alpha X+\beta)^j a_{2j+1}(X) \in K[X]$. Representing a polynomial f this way requires $\deg(a) + \deg(b)$ elements of K. So, we can forget our above representation of elements of the function field $\frac{f+I(\mathcal{E}/K)}{g+I(\mathcal{E}/K)}$ with $f, g \in K[X,Y]$, and use $\frac{a_0(X)+b_0(X)Y+I(\mathcal{E}/K)}{a_1(X)+b_1(X)Y+I(\mathcal{E}/K)}$ for some $a_0, a_1, b_0, b_1 \in K[X]$, representing the function field element with only $\deg(a_0) + \deg(a_1) + \deg(b_0) + \deg(b_1)$ elements of K instead.

We may go further, however. For each, we have an element $a_1(X) - b_1(X)Y + I(\mathcal{E}/K) \in K[X,Y]/I(\mathcal{E}/K)$ satisfying the following due to the ideal $I(\mathcal{E}/K)$ and the multiplicative absorption

property of ideals.

$$(a_1(X) + b_1(X)Y + I(\mathcal{E}/K))(a_1(X) - b_1(X)Y + I(\mathcal{E}/K)) = a_1^2 - b_1^2Y^2 + I(\mathcal{E}/K) \tag{10}$$

$$= a_1^2 - (X^3 + \alpha X + \beta)b_1^2 + I(\mathcal{E}/K) \tag{11}$$

Hence, for any $\frac{a_0(X)+b_0(X)Y+I(\mathcal{E}/K)}{a_1(X)+b_1(X)Y+I(\mathcal{E}/K)} \in K(\mathcal{E})$, we can apply an identity map by multiplying and dividing by $(a_1(X) - b_1(X)Y + I(\mathcal{E}/K))$, yielding another representative whose denominator does not depend on X as follows

$$\frac{a_0(X) + b_0(X)Y + I(\mathcal{E}/K)}{a_1(X) + b_1(X)Y + I(\mathcal{E}/K)} \cdot \frac{a_1(X) - b_1(X)Y + I(\mathcal{E}/K)}{a_1(X) - b_1(X)Y + I(\mathcal{E}/K)} \tag{12}$$

$$= \frac{a_0(X)a_1(X) - a_0(X)b_1(X)Y + a_1(X)b_0(X)Y - b_0(X)b_1(X)Y^2 + I(\mathcal{E}/K)}{(a_1(X))^2 - (b_1(X))^2Y^2 + I(\mathcal{E}/K)} \tag{13}$$

$$= \frac{f_0(X) + f_1(X)Y + I(\mathcal{E}/K)}{f_3(X) + I(\mathcal{E}/K)} \tag{14}$$

where $f_0(X) = a_0(X)a_1(X) - (X^3 + \alpha X + \beta)b_0(X)b_1(X)$, $f_1(X) = a_1(X)b_0(X) - a_0(X)b_1(X)$, and $f_2(X) = (a_1(X))^2 - (X^3 + \alpha X + \beta)(b_2(X))^2$. This requires only $\deg(f_0(X)) + \deg(f_1(X)) + \deg(f_2(X))$ elements of K.

Thus, we write every element of $K(\mathcal{E})$ as $\frac{f_0(X)+f_1(X)Y+I(\mathcal{E}/K)}{f_2(X)+I(\mathcal{E}/K)}$ for some polynomials $f_0(X)$, $f_1(X)$, $f_2(X) \in K[X]$.

2.4 Elliptic curve groups

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 affine curve $\mathcal{E} = v_a(e)$. 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 $K(\mathcal{E}^n) = \mathrm{Frac}(K[X_1, Y_1, \ldots, X_n, Y_n]/I(\mathcal{E}^n/K))$.

The projective variety corresponding to $\mathcal{E}$ is $\mathbb{G}(\mathcal{E}) = v_p(e^*)$, where 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: the point at infinity $O = (0 : 1 : 0)$ is distinguished as the group identity. For any non-identity point $P \in \mathbb{G}(\mathcal{E}) \setminus \{O\}$, it must be that P lies on some unique projective line ℓ which is tangent to $\mathcal{E}$. Bezout's theorem implies ℓ intersects $\mathcal{E}$ at P with multiplicity 2 and at a second point R with multiplicity 1. In this case, the elliptic curve group law states $P + P + R = O$, so $R = -2P$. Then, given points $P, Q \in \mathbb{G}(\mathcal{E})$, there exists a unique projective line ℓ that interpolates P and Q. The line may or may not be vertical, where vertical lines satisfy a trivial relation. If this line is vertical, then the elliptic curve group law declares $P + Q = O$, so $Q = -P$. Otherwise, Bezout's theorem ensures that ℓ intersects $\mathcal{E}$ at a third point R (which may or may not be distinct from P and Q). In this case, the elliptic curve group law declares $P + Q + R = O$, so $R = -(P + Q)$. 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 $(x : y : z)$. Moreover, the only solution to e^* in $\mathbb{P}^2$ with $z = 0$ is exactly O, so if P, Q, and R are non-identity, then these have $z \neq 0$. Thus, 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.

In the following, we take Bézout's theorem for granted (for more details, see Theorem I.7.8 in [Har13]).

Theorem 2.1 (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 $\sum_{i=1}^{s} m_i = n_1 n_2$.*

If $\mathcal{F}_1 = \mathcal{E}$ and $\mathcal{F}_2$ is a line ℓ, then $\sum_i m_i = 3$. The partitions of 3 in $\mathbb{N}$ are $1 + 1 + 1$, $1 + 2$, and 3, so every line intersects $\mathcal{E}$ at 3 points, counting multiplicities. To fully work out the group structure requires identities, inverses, and the group operation as defined for arbitrary pairs.

First, the point at infinity in projective space $O = (0 : 1 : 0) \in \mathbb{P}^2(K)$ is distinguished as the group identity[2]. Next, we find the additive inverse of a non-identity $P \in \mathbb{G}$, i.e. we seek $Q \in \mathbb{G}$ such that $P + Q + O = O$. Then $P = -Q$. By Bezout's theorem, we need a line interpolating P, Q, O, and the multiplicities of these points sum to 3. So we have three cases.

If P has multiplicity 3, then $P = Q = O$, a contradiction. If P has multiplicity 2, then $P = Q$ yet $P + Q + O = P + P = 2P = O$. If $4\alpha^3 + 27\beta^2 \neq 0$, then $\mathcal{E}$ is not singular, a contradiction. So P, Q, and O form a 3-set of intersection points, and each has multiplicity 1. It is easy to check that a line in $\mathbb{P}^2(\overline{K})$ intersects $\mathcal{E}$ at O if and only if the line is vertical. Thus, to invert P, we just reflect $P = (x_P, y_P) \in \mathbb{P}^2$ across the X-axis, setting $-P = (x_P, -y_P)$.

This establishes the group identity and inverse elements. It remains to establish the group operation on arbitrary pairs of points, P, Q. We can assume $P \neq -Q$ as we handled that case already. Given these, to compute R such that $P + Q + R = O$, we find a line ℓ interpolating P and Q and consider Bézout's theorem: the line and $\mathcal{E}$ may intersect at P with multiplicity 3, 2, or 1.

In the first case, ℓ intersects $\mathcal{E}$ at P with multiplicity 3. In this case, $P + P + P = O$ implies $3P = O$, so P is 3-torsion, implying $\mathcal{E}$ is singular. If $4\alpha^3 + 27\beta^2 \neq 0$, then $\mathcal{E}$ is not singular, a contradiction.

In the second case, ℓ intersects $\mathcal{E}$ at P with multiplicity 2. In this case $P + P + Q = O$, and so the line interpolating P and Q is tangent to $\mathcal{E}$ at P, with multiplicity 2. That is to say, the second case occurs when we add a point P to itself, "doubling" it, computing $P + P = 2P = -Q$.

[2] See [Sil09] and [Har13] for details, general constructions, and more.

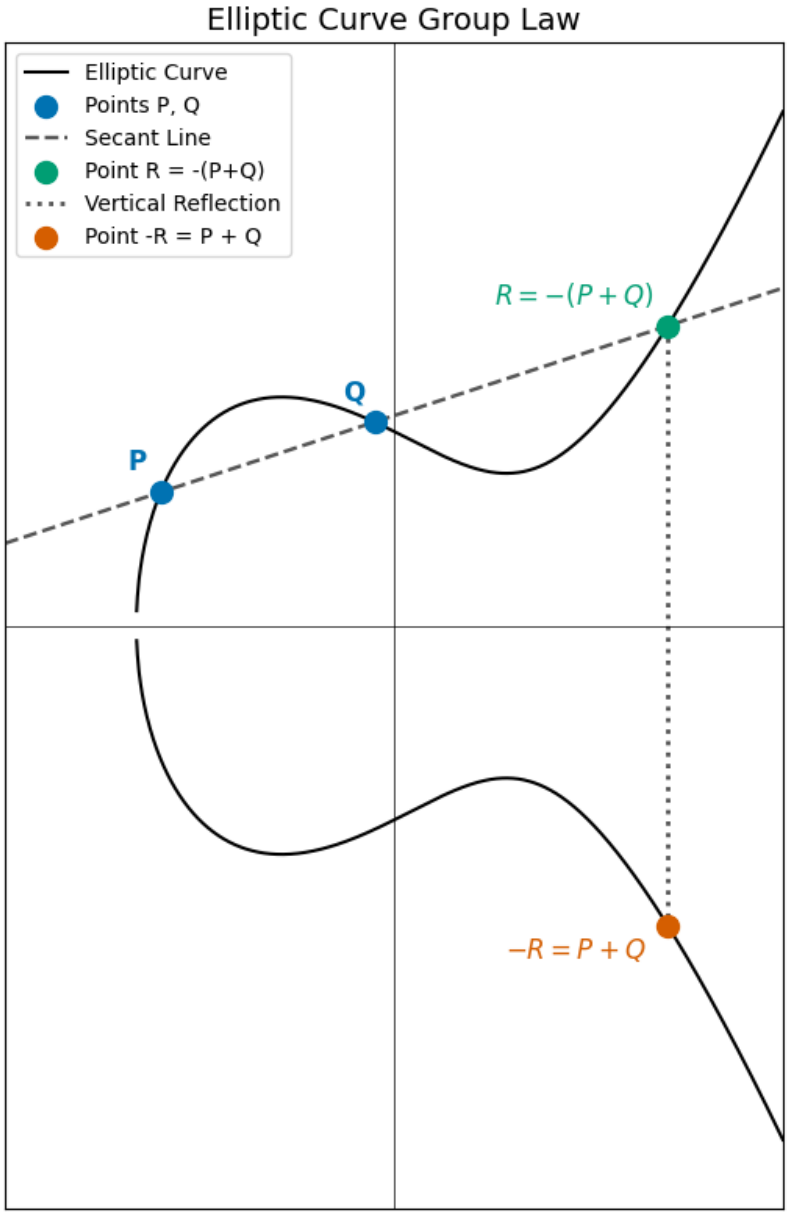

Figure 1: The points P, Q, R are collinear, so their sum is $P + Q + R = O$, where O is the group identity. In particular, $R = -(P + Q)$ and its reflection across the X-axis is $-R = P + Q$.

In the final case, the line ℓ intersects $\mathcal{E}$ at three distinct points P, Q, and R, necessarily with multiplicity 1 each by Bézout's theorem. That is to say, the line ℓ intersects $\mathcal{E}$ transversally at P, Q, R. For these points, $P+Q+R=O$, so $R=-(P+Q)$. Moreover, we can write the coordinates $R=(x_R,y_R)$ with $y_R=\lambda x_R+\mu$ where $(\lambda,\mu)\in\overline{K}^2$ parameterize the line interpolating P and Q.

These rules are sufficient for generating the elliptic curve group $\mathbb{G}(\mathcal{E})$. 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.

2.4.1 Group law and summing points in greater detail

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 A later. As both are not the identity and $P\neq\pm Q$, the line interpolating P and Q is neither vertical nor tangent to $\mathcal{E}$. By Bezout's theorem, the line must intersect $\mathcal{E}$ transversally at P and Q, as well as at a third distinct point R, and these intersection points all have multiplicity 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

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

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 $(x_P,y_P),(x_Q,y_Q)\in\mathcal{E}$; 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 $X^3-\lambda^2X^2+(\alpha-2\lambda\mu)X+(\beta-\mu^2)$ in X. All the roots are accounted for, so the polynomial splits into linear factors.

$$(X-x_P)(X-x_Q)(X-x_R)=X^3-\lambda^2X^2+(\alpha-2\lambda\mu)X+(\beta-\mu^2)\tag{15}$$

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

y_R as rational functions in the coordinates of P and Q:

$$x_R = \lambda^2 - x_P - x_Q \tag{16}$$

$$y_R = \lambda x_R + \mu \tag{17}$$

Recalling λ and μ defined above are rational functions, it is clear that these are both rational in (x_P, x_Q), as promised. Thus, the group law demands the following system of equations are satisfied.

$$\begin{cases} \lambda = & \frac{y_Q - y_P}{x_Q - x_P} \\ \mu = & y_P - \lambda x_P \\ y_P = & \lambda x_P + \mu \\ y_Q = & \lambda x_Q + \mu \\ y_R = & \lambda x_R + \mu \\ y_P^2 = & x_P^3 + \alpha x_P + \beta \\ y_Q^2 = & x_Q^3 + \alpha x_Q + \beta \\ y_R^2 = & x_R^3 + \alpha x_R + \beta \end{cases} \tag{18}$$

2.4.2 Problems with previous approaches

This system of 8 equations with 6 unknowns influences many computations with derivations in the sequel. Indeed, students of the standard calculus sequence may be familiar with using the *Jacobian* of similar systems of equations to determine if a system is smooth at a point. This yields chain-rule equations when differentiating function field elements at P, Q, and R.

This is precisely the complication left unaddressed by Eagen in [Eag22], and tackled by Bassa in [Bas]. We think the work in [Eag22] intended characterize solutions to this system as dependent upon λ and μ, the only free variables, exactly determining solutions to this system. Had this been correct, the resulting verification equations would have reduced Eagen's claims. This would simplify verification by allowing the chain rule to be computed with respect to the line $Y - \lambda X - \mu = 0$.

While solutions for this system of equations in terms of λ and μ are easily obtainable using Cardano's formula, the solutions are not guaranteed to be rational functions in λ and μ. In fact, generally these solutions are not rational, which complicates the application of derivatives beyond the computations in [Eag22].

On the other hand, by picking P and Q, we obtain x_P, y_P, x_Q, y_Q, instead over-determining the solution to this system; as Bassa noted, $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ form a pair on the surface $(P, Q) \in \mathcal{E}^2$. With this, solutions are rational in x_P, y_P, x_Q, y_Q. 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. Thus, we really have five unknown field

elements $(x_P, x_Q, x_R, \lambda, \mu)$ and 3 sign bits (one each to uniquely determine y_P, y_Q, y_R). Above, we pick P and Q, i.e. two field elements and two bits, and this is sufficient to uniquely determine R, λ, and μ. By over-determining our solutions, we risk that with small probability a solution may not exist, however we only slightly over-determine our solutions by 2 bits. For this price, we then obtain rational solutions.

2.5 Divisors over $\mathcal{E}$

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 $\mathrm{Div}_K(\mathcal{E})$. To each $\nu : \mathcal{E} \to \mathbb{Z}$ with finite support, we assign a unique divisor $[\nu] = \sum_{T \in \mathcal{E}} \nu(T)\,[T]$. The *degree* of $[\nu]$ is $\deg([\nu]) = \sum_T \nu(T)$. Let $\mathrm{Div}^0_K(\mathcal{E})$ denote the set of degree-zero divisors. Note $\mathrm{Div}^0_K(\mathcal{E}) \subseteq \mathrm{Div}_K(\mathcal{E})$ 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 $\mathrm{div}(f) = \sum_{T \in \mathcal{E}} \mathrm{ord}_T(f)[T]$. We say these divisors are *principal divisors* because they are related to computing fractional ideals. Define $\mathrm{Prin}(\mathcal{E}) = \{\mathrm{div}(f) \mid f \in K(\mathcal{E})\}$. Note we have the following chain of subgroups: $\mathrm{Prin}(\mathcal{E}) \subseteq \mathrm{Div}^0_K(\mathcal{E}) \subseteq \mathrm{Div}_K(\mathcal{E})$. In the sequel, define the following relation.

$$\mathcal{R}_{\mathrm{div}} = \big\{(f, [\nu]) \mid f \in K(\mathcal{E}), [\nu] \in \mathrm{Div}^0_K(\mathcal{E}), \text{ and } \mathrm{div}(f) = [\nu]\big\} \tag{19}$$

We take the following for granted.

Theorem 1. Let $f \in K(\mathcal{E})$, $v_a(f) \setminus \{O\} = \{P_1, \ldots, P_n\}$, 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 to the additive subgroup $\mathrm{Prin}(\mathcal{E})$.

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 $\mathrm{div}(f_1) + \mathrm{div}(f_2)$.

2.6 Weil's reciprocity

Every $f \in \overline{K}(\mathcal{E})$ induces a map on divisors. Indeed, let $\sum_T \eta(T)[T]$ be any divisor, which we denote with $[\eta]$, such that the support of η is disjoint from the roots and poles of f. Then let $\widehat{f}([\eta]) = \prod_T f(T)^{\eta(T)}$; this is well-defined because the points of $[\eta]$ are disjoint from the roots and poles of f.

Theorem 2 (Weil's Reciprocity Law)**.** Let $f, g \in \overline{K}(\mathcal{E})$, with $\mathrm{div}(f) = \sum_T \nu(T)[T]$ and $\mathrm{div}(g) = \sum_T \eta(T)[T]$. Then $\prod_{T \in \mathbb{G}} f(T)^{\eta(T)} = \prod_{S \in \mathbb{G}} g(S)^{\nu(S)}$.

In particular, if g has a degree 1 polynomial representing its numerator $aX + bY + c + (e)$ and a non-zero constant representing its denominator, say $g = \frac{aX+bY+c}{d}$ for some constants $(a, b, c, d) \in K^4$, then we can re-scale g so that the denominator is $1 + (e)$, say $g = \ell = \frac{\frac{a}{d}X + \frac{b}{d}Y + \frac{c}{d} + (e)}{1+(e)}$ without losing any generality; for these, we abuse notation and write $g = \ell = \frac{a}{d}X + \frac{b}{d}Y + \frac{c}{d}$. Then $\widehat{g}(\nu) = \prod_T (ax_T + by_T + c)^{\nu(T)}$.

2.7 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.

Given any derivation $d : L \to L$ over K, there is a natural map $\delta : L \to L$ 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.

2.8 Schwarz-Zippel

We take the following for granted.

Lemma 2.1. 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 $\frac{d}{|S|}$.

3 Method

Suppose that a prover and verifier wish to implement a proving scheme in an elliptic curve group setting. In most cases, verification requires satisfying one or more linear constraints coming from the group law of $\mathbb{G}$. That is to say, a proof includes some non-identity group elements $P_1, \ldots, P_n \in \mathbb{G} \setminus \{O\}$ and integer weights $m_1, \ldots, m_n \in \mathbb{Z}$, then the verification procedure requires checking the $\sum_{i=1}^{n} m_i P_i = O$.

The prover and verifier recall that every principal divisor $\sum_{i=1}^{n} m_i[P_i] - (\sum_{i=1}^{n} m_i)[O]$ corresponds to a decomposition of O, so they can merely check whether some divisor is principal. The map *div* is a surjective group homomorphism. Therefore, there exists an element $f \in K(\mathcal{E})$ such that $\operatorname{div}(f) = \sum_i m_i[P_i] - (\sum_i m_i)\,[O]$, which can be used as a witness of principality.

The prover and verifier also recall Weil's reciprocity provides $\widehat{f}(\operatorname{div}(g)) = \widehat{g}(\operatorname{div}(f))$ for every pair $f, g \in K(\mathcal{E}) \subseteq \overline{K}(\mathcal{E})$. For a fixed f, Weil's reciprocity implies an equality of rational functions in the coefficients of g and the coordinates of the points of $\operatorname{div}(g)$; clearing denominators, we obtain a polynomial expression. Evaluating this polynomial at a given $(g, \operatorname{div}(g))$ always yields equality.

Thus, if $f \in K(\mathcal{E})$ and some random $(g, \operatorname{div}(g))$ is a root of the polynomial obtained by rationalizing $\widehat{f}(\operatorname{div}(g)) - \widehat{g}(\operatorname{div}(f))$, then the Schwarz-Zippel Lemma implies that the polynomial is the zero polynomial exactly, except with a probability inversely proportional to the cardinality of the set from which we sample g.

Hence, checking whether $\widehat{f}(\operatorname{div}(g)) = \widehat{g}([\nu])$ for a random g is sufficient to determine if $\operatorname{div}(f) = [\nu]$, except perhaps with a probability given by the Schwarz-Zippel Lemma. Thus, the prover and verifier can conclude that $\sum_{T \in \mathbb{G}} \nu(T)T = O$, except with that probability.

To make the checks as efficient as possible, we sample g with degree 1, and we perform computations under logarithmic derivatives, drastically reducing computational costs.

3.1 Description

Let $H : \{0,1\}^* \to (\mathbb{G} \setminus \{O\})^2$ be a cryptographic hash function. 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 $\texttt{params}$.
- $\texttt{Prove}(\texttt{params}, [\nu]) \to \pi$. The prover does the following:

1. Compute $\texttt{out} \leftarrow \texttt{MakeWitness}(\texttt{params}, [\nu])$ with Algorithm 1; if $\texttt{out} = \perp$, output $\perp$ and terminate.
2. Compute $(P, Q) \leftarrow H(\texttt{params}, f, [\nu])$. If $P = \pm Q$, output $\perp$ and terminate.
3. Compute $R = -(P + Q)$. If $P = \pm R$ or $Q = \pm R$ or $\{P, Q, R\} \cap \{P_i\}_{i=1}^{n} \neq \emptyset$, output $\perp$ and terminate.
4. Parse the following, if possible; otherwise, output $\perp$ and terminate.
   - $f \leftarrow \texttt{out}$,
   - $\frac{f_1(X)+Yf_2(X)+(e)}{f_3(X)+(e)} \leftarrow f$,
   - $\sum_{i=1}^{n} m_i[P_i] \leftarrow [\nu]$,
   - $(x_P, y_P) \leftarrow P$,
   - $(x_Q, y_Q) \leftarrow Q$, and
   - $(x_R, y_R) \leftarrow R$.
5. If $y_T = 0$ for any $T \in \{P, Q, R\}$, output $\perp$ and terminate.
6. Compute $y'_T = \frac{3x_T^2+\alpha}{2y_T}$ for each $T \in \{P, Q, R\}$.
7. If $y'_T = \lambda$ or $f(T) = 0$ or $f_3(x_T) = 0$ for any $T \in \{P, Q, R\}$, output $\perp$ and terminate.
8. Compute the following:

$$\Delta X = x_Q - x_P \tag{20}$$
$$\Delta Y = y_Q - y_P \tag{21}$$
$$\lambda = \frac{\Delta Y}{\Delta X} \tag{22}$$
$$\mu = y_P - \lambda x_P \tag{23}$$
$$f_X = \frac{(f'_1 + Yf'_2)f_3 - f \cdot f'_3 + (e)}{f_3^2 + (e)} \tag{24}$$
$$f_Y = \frac{f_2 + (e)}{f_3 + (e)} \tag{25}$$

9. Compute the following:

$$A_i = (y_i - \lambda x_i - \mu)^{-1} \text{ for each } 1 \leq i \leq n \tag{26}$$
$$B_1 = \frac{2y_P f_X(P) + (3x_P^2 + \alpha)f_Y(P)}{2y_P f(P)(3x_P^2 + \alpha - 2\lambda y_P)} \tag{27}$$
$$B_2 = -\frac{2\lambda y_P(2y_R f_X(R) + (3x_R^2 + \alpha)f_Y(R))(3x_P^2 + \alpha - 2y_P(\lambda + \Delta X))}{2y_P y_R f(R)(3x_P^2 + \alpha - 2y_P\lambda)\Delta X} \tag{28}$$
$$B_3 = \frac{2y_Q f_X(Q) + (3x_Q^2 + \alpha)f_Y(Q)}{2y_Q f(Q)(3x_Q^2 + \alpha - 2\lambda y_Q)} \tag{29}$$
$$B_4 = \frac{2\lambda y_Q(2y_R f_X(R) + (3x_R^2 + \alpha)f_Y(R))(3x_Q^2 + \alpha - 2y_Q(\lambda + \Delta X))}{2y_Q y_R f(R)(3x_Q^2 + \alpha - 2\lambda y_Q)\Delta X} \tag{30}$$

10. Output $\pi = (f, A_1, \ldots, A_n, B_1, B_2, B_3, B_4)$.

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

1. Parse $\sum_{i=1}^{n} m_i P_i \leftarrow [\nu]$ and $(f, A_1, \ldots, A_n, B_1, B_2, B_3, B_4) \leftarrow \pi$, and $\frac{f_1(X)+Yf_2(X)+(e)}{f_3(X)+(e)} \leftarrow f$ if possible; if not, output 0 and terminate.
2. Compute challenge points $(P, Q) \leftarrow H(\texttt{params}, f, [\nu])$. If $P = \pm Q$, output 0 and terminate.
3. Compute $R = -(P + Q)$. If $P = \pm R$ or $Q = \pm R$ or $\{P, Q, R\} \cap \{P_i\}_{i=1}^{n} \neq \emptyset$, output $\perp$ and terminate.
4. Carry out step 8 from `Prove`.
5. Compute the following.

$$b_1 = 2y_P f(P)(3x_P^2 + \alpha - 2\lambda y_P) \tag{31}$$
$$c_1 = 2y_P f_X(P) + (3x_P^2 + \alpha) f_Y(P) \tag{32}$$
$$b_2 = 2y_P y_R f(R)(3x_P^2 + \alpha - 2y_P\lambda)\Delta X \tag{33}$$
$$c_2 = -2\lambda y_P(2y_R f_X(R) + (3x_R^2 + \alpha) f_Y(R))(3x_P^2 + \alpha - 2y_P(\lambda + \Delta X)) \tag{34}$$
$$b_3 = 2y_Q f(Q)(3x_Q^2 + \alpha - 2\lambda y_Q) \tag{35}$$
$$c_3 = 2y_Q f_X(Q) + (3x_Q^2 + \alpha) f_Y(Q) \tag{36}$$
$$b_4 = \big(2y_Q y_R f(R)(3x_Q^2 + \alpha - 2\lambda y_Q)\Delta X\big) \tag{37}$$
$$c_4 = 2\lambda y_Q(2y_R f_X(R) + (3x_R^2 + \alpha) f_Y(R))(3x_Q^2 + \alpha - 2y_Q(\lambda + \Delta X)) \tag{38}$$

6. If any $1 \leq i \leq n$ has $(y_i - \lambda x_i - \mu)A_i \neq 1$, output 0 and terminate.
7. If any $1 \leq i \leq 4$ has $b_i B_i \neq c_i$, output 0 and terminate.
8. If $B_1 + B_2 + B_3 + B_4 \neq \sum_i m_i A_i$, output 0 and terminate.
9. Otherwise, output 1 and terminate.

`MakeWitness`$([P_i]) \to$ `out` $\in \{\perp\} \cup K(\mathcal{E})$

If no points were provided, output $\perp$ and terminate.
If the input points do not sum to the identity O, or any input point is O,
  output $\perp$ and terminate.
Initialize an empty queue `queue` $= \emptyset$.
Parse the point set into pairs. For each pair (P_{2j}, P_{2j+1}):
  (If the number of points is odd, set $P_{2j+1} \leftarrow O$ for the final pair.)
  Compute:
  $T_j = P_{2j} + P_{2j+1} \in \mathcal{E}$,
  ℓ_j, the interpolation line divisor: $\ell_j =$ `line`$(P_{2j}, P_{2j+1}) \in K(\mathcal{E})$
  Queue $(2, T_j, \ell_j)$ into `queue`.
While `len`(`queue`) > 1:
  Initialize empty queue `nextQueue` $= \emptyset$.
  If `len`(`queue`) $\equiv 1 \mod 2$:
  Pop the last element from `queue` and push it onto `nextQueue`.
  Let $N = \lfloor$`len`(`queue`)$/2 \rfloor$.
  For $1 \le j \le N$:
  Pop (m_a, T_a, f_a) and (m_b, T_b, f_b) from `queue`.
  Compute:
  $T_j = T_a + T_b$,
  $\widehat{\ell_j} =$ `line`(T_a, T_b),
  numerator $= f_a \cdot f_b \cdot \widehat{\ell_j} \mod e(X, Y)$,
  denominator $=$ `line`$(T_a, -T_a) \cdot$ `line`$(T_b, -T_b)$,
  Set $g :=$ numerator/denominator.
  Queue $(m_a + m_b, T_j, g)$ into `nextQueue`.
  Set `queue` $\leftarrow$ `nextQueue`.
Pop $(m, T, f) \leftarrow$ `queue`.
If $T \neq O$, output $\perp$ and terminate.
Output f.

Algorithm 1: Constructing a rational function whose divisor interpolates a list of elliptic curve points.

The runtime of `MakeWitness` is $O(n^2)$.

Usage of this scheme is somewhat self-evident; for example, to prove that a scalar multiplication has been performed correctly, say $P = aG$ for some $G \in \mathbb{G} \setminus \{O\}$ and some $1 \le a < \text{ord}(\mathbb{G})$, the prover runs `MakeWitness` with $[\nu] = a[G] + [-P]$.

3.2 Security properties

It should be clear that, if $[\nu]$ is a nontrivial decomposition of O into a sum of non-identity points, then `MakePairs` certainly does not output $\perp$.

Lemma 3.1. Let $\{P_1, \ldots, P_n\}$ be a subset of $\mathbb{G}$. Let `Collinear`$(\mathbb{G})$ be the set of 3-sets of points, say $\{P, Q, R\} \subseteq \mathbb{G}$, which are distinct, on $\mathcal{E}$, and collinear. Then $\mathbb{P}\left[\texttt{Collinear} \cap \{P_1, \ldots, P_n\} \neq \emptyset\right] = \frac{1}{6}\left(|\mathbb{G}| - 1\right)\left(|\mathbb{G}| - 2\right)$.

Proof. There are $|\mathbb{G}| - n$ choices of P which do not collide with any P_i. After selecting P, there are $|\mathbb{G}| - (n+1)$ choices of Q which do not collide with any P_i. □

We also have the following.

Lemma 3.2. If $[\nu]$ is a nontrivial decomposition of O into a sum of non-identity points, then `MakeWitness` runs in time $t = O()$, succeeds with certainty, and outputs f such that $\operatorname{div}(f) = [\nu] - \deg([\nu])[O]$.

We omit the proof for this claim for now; see upcoming work.

Theorem 3. Let $\mathbb{G}$ have prime order $q > 3$. Π is a complete proving system for the relation $\mathcal{R}$ from Equation 19 with the completeness error $\kappa \leq \frac{2n+8}{|\mathbb{G}|} \in O(1/|\mathbb{G}|)$.

Proof. Presume that $f \in K(\mathcal{E})$, $[\nu] = \operatorname{div}(f)$, and the prover honestly executes `Prove`. Then the following occurs.

(a) The prover computes $\texttt{out} \leftarrow \texttt{MakeWitness}(\texttt{params}, [\nu])$. Following Lemma 3.2, this algorithm does not fail and $\texttt{out} = f \in K(\mathcal{E})$, where $f = \frac{f_1(X)+Yf_2(X)+(e(X,Y))}{f_3(X)+(e(X,Y))}$.

(b) The prover computes $(P, Q) \leftarrow H(\texttt{params}, f, [\nu])$ and fails if $P = \pm Q$. Under the random oracle model, H has output uniformly sampled from $(\mathbb{G} \setminus \{O\})^2$. There are $|\mathbb{G}| - 1$ pairs (P, P) with $P \neq O$. There are $\frac{1}{2}(|\mathbb{G}| - 1)$ unordered pairs $(P, -P)$ with $P \neq O$. Since the order of $\mathbb{G}$ is q, every non-identity element has order $q > 3$, so no group element P satisfies $P + P = O$. Hence, we have $\frac{3}{2}(|\mathbb{G}| - 1)$ pairs which may cause the prover to output $\perp$, out of $(|\mathbb{G}| - 1)^2$. In particular, the prover fails with probability $\frac{3}{2(|\mathbb{G}|-1)}$.

(c) The prover computes $R = -(P + Q)$. Note the constraints that $P, Q \neq O$ and $P \neq \pm Q$ admit an interpolating line ℓ which is not vertical and not tangent to $\mathcal{E}$, so $R \neq \pm P$ and $R \neq \pm Q$.

Since (P, Q) are sampled from a random oracle, this line ℓ is sampled uniformly from the set of all lines in $K[X, Y]/I(\mathcal{E}/K)$ which are vertical and not tangent to $\mathcal{E}$. Thus, the set $\{P, Q, R\}$ is sampled uniformly from the set of all 3-sets of collinear non-identity points on $\mathcal{E}$. Each such triple is uniquely determined by P and Q. By Lemma 3.1, there are $\frac{1}{6}(|\mathbb{G}| - 1)(|\mathbb{G}| - 2)$ such triples. On the other hand, there are $|\mathbb{G} \setminus \{O, P_1, \ldots, P_n\}| = |\mathbb{G}| - n - 1$ group elements which miss the divisor points, so $\frac{1}{6}(|\mathbb{G}| - n - 1)(|\mathbb{G}| - n - 2)$ triples do not collide with the points $\{P_1, \ldots, P_n\}$. Hence, the probability of failure here is $\mathbb{P}[\{P, Q, R\} \cap \{P_1, \ldots, P_n\} \neq \emptyset] = 1 - \left(1 - \frac{n}{|\mathbb{G}|-1}\right)\left(1 - \frac{n}{|\mathbb{G}|-2}\right) = O\left(\frac{n^2}{|\mathbb{G}|^2}\right)$. This is asymptotically $O(|\mathbb{G}|^{-2})$, but concrete choices of n and $\mathbb{G}$ may lead to insecurity.

(d) As the prover is honest, parsing in this step succeeds with certainty.

(e) Unless the prover already failed, we have that $\{P, Q, R\}$ misses the roots and poles of f, so $f(T)$ and $f_3(x_T)$ are both non-zero and well-defined for each $T \in \{P, Q, R\}$. Also, at most one point on $\mathcal{E}$ has a zero y-coordinate. There are $|\mathbb{G}| - 2$ lines ℓ which interpolate the point with a zero y-coordinate, out of $\frac{1}{6}(|\mathbb{G}| - 1)(|\mathbb{G}| - 2)$. Hence, there is a probability of $\frac{6}{|\mathbb{G}|-1}$ that the prover outputs $\perp$ in this step.

(f) As the prover is honest, computations in this step succeed with certainty.

(g) If the prover did not fail before this point, then $\{P, Q, R\} \cap \{P_i\}_{i=1}^{n} = \emptyset$, where these P_i exhaust the roots and poles of f. Hence, $f(P)$, $f(Q)$, $f(R)$, are each non-zero and well-defined, so the prover succeeds in this step with certainty.

(h) If the prover has not failed before this step, then all the computations in this step succeed with certainty.

(i) The prover outputs $\pi = (f, A_1, \ldots, A_n, B_1, B_2, B_3, B_4)$.

If the prover does not fail, then during verification the following occurs.

(a) Parsing succeeds with certainty here, so the verifier does not output 0 in this step.

(b) Since the prover is honest and did not fail, the witness is a valid function field element, and the verifier obtains (P, Q) with $P \neq \pm Q$ without outputting 0 here.

(c) The verifier computes $R = -(P + Q)$ and checks for a collision. Since the prover is honest and did not fail, the points P, Q, R do not collide with the points $\{P_i\}$, so the verifier does not output 0 in this step.

(d) If the verifier has not yet terminated, then all the computations in this step succeed with certainty and we obtain the following.

$$b_1 = 2y_P f(P)(3x_P^2 + \alpha - 2\lambda y_P) \tag{39}$$
$$c_1 = 2y_P f_X(P) + (3x_P^2 + \alpha) f_Y(P) \tag{40}$$
$$b_2 = 2y_P y_R f(R)(3x_P^2 + \alpha - 2y_P\lambda)\Delta X \tag{41}$$
$$c_2 = -2\lambda y_P(2y_R f_X(R) + (3x_R^2 + \alpha) f_Y(R))(3x_P^2 + \alpha - 2y_P(\lambda + \Delta X)) \tag{42}$$
$$b_3 = 2y_Q f(Q)(3x_Q^2 + \alpha - 2\lambda y_Q) \tag{43}$$
$$c_3 = 2y_Q f_X(Q) + (3x_Q^2 + \alpha) f_Y(Q) \tag{44}$$
$$b_4 = \big(2y_Q y_R f(R)(3x_Q^2 + \alpha - 2\lambda y_Q)\Delta X\big) \tag{45}$$
$$c_4 = 2\lambda y_Q(2y_R f_X(R) + (3x_R^2 + \alpha) f_Y(R))(3x_Q^2 + \alpha - 2y_Q(\lambda + \Delta X)) \tag{46}$$

(e) Each $A_i = (y_i - \lambda x_i - \mu)$ since the prover was honest, so $(y_i - \lambda x_i - \mu)A_i = 1$ with certainty, and the verifier does not output 0 in this step.

(f) We have the following, also since the prover was honest.

$$B_1 = \frac{2y_P f_X(P) + (3x_P^2 + \alpha) f_Y(P)}{2y_P f(P)(3x_P^2 + \alpha - 2\lambda y_P)} = \frac{c_1}{b_1} \tag{47}$$

$$B_2 = -\frac{2\lambda y_P (2y_R f_X(R) + (3x_R^2 + \alpha) f_Y(R))(3x_P^2 + \alpha - 2y_P(\lambda + \Delta X))}{2y_P y_R f(R)(3x_P^2 + \alpha - 2\lambda y_P)\Delta X} = \frac{c_2}{b_2} \tag{48}$$

$$B_3 = \frac{2y_Q f_X(Q) + (3x_Q^2 + \alpha) f_Y(Q)}{2y_Q f(Q)(3x_Q^2 + \alpha - 2\lambda y_Q)} = \frac{c_3}{b_3} \tag{49}$$

$$B_4 = \frac{2\lambda y_Q (2y_R f_X(R) + (3x_R^2 + \alpha) f_Y(R))(3x_Q^2 + \alpha - 2y_Q(\lambda + \Delta X))}{2y_Q y_R f(R)(3x_Q^2 + \alpha - 2\lambda y_Q)\Delta X} = \frac{c_4}{b_4} \tag{50}$$

Thus, each $b_i B_i = c_i$, so the verifier does not output 0 in this step.

(g) In Equation 111 of the Appendix, we show $\delta(\widehat{f}(X_P, Y_P, X_Q, Y_Q)) \mid_{P,Q} = B_1 + B_2 + B_3 + B_4$. We also have $\widehat{f}(x_P, y_P, x_Q, y_Q) = f(P)f(Q)f(R)$, and by Weil's 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.

Let $\rho_b = \frac{3}{2|\mathbb{G}|-2}$ be the probability of failure in step b. Let $\rho_c = 1 - \left(1 - \frac{n}{|\mathbb{G}|-1}\right)\left(1 - \frac{n}{|\mathbb{G}|-2}\right)$ be the probability of failure in step c. Let $\rho_e = \frac{6}{|\mathbb{G}|-1}$ be the probability of failure in step e. Then, by the law of total probability, the completeness error of the protocol is $\rho_b + (1-\rho_b)(\rho_c + (1-\rho_c)\rho_e)$. This simplifies as follows:

$$\begin{aligned}
\kappa &= \frac{3(5\,|\mathbb{G}| - 11)}{2(|\mathbb{G}| - 1)^2} + \frac{n(|\mathbb{G}| - 7)(2\,|\mathbb{G}| - 5)(2\,|\mathbb{G}| - (n+3))}{2(|\mathbb{G}| - 2)(|\mathbb{G}| - 1)^3} \\
&\approx \frac{15\,|\mathbb{G}|}{2\,|\mathbb{G}|^2} + \frac{4n\,|\mathbb{G}|^3}{2\,|\mathbb{G}|^4} \\
&= \frac{4n + 15}{2\,|\mathbb{G}|} \\
&\leq \frac{2n + 8}{|\mathbb{G}|}
\end{aligned}$$

□

Theorem 4. Π is a sound proving system for the relation

$$\mathcal{R} = \{(f, \mathrm{div}(f)) \mid f \in K(\mathcal{E}) \text{ and } \mathrm{div}(f) \in Prin(\mathbb{G})\}.$$

Proof. Assume $\texttt{Verify}(\texttt{params}, \sum_{i=1}^{n} m_i[P_i], \pi) = 1$. Then $\pi = (f, A_1, \ldots, A_n, B_1, B_2, B_3, B_4)$ for some $f \in K(\mathcal{E})$ and some $A_1, \ldots, A_n, B_1, B_2, B_3, B_4 \in K$, otherwise the verifier would output 0 in the first step, a contradiction. Moreover, the verifier computes challenge points $(P, Q) \leftarrow H(\texttt{params}, f, \sum_{i=1}^{n} m_i[P_i])$ such that $P \neq \pm Q$, 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, R, and (b_i, c_i) for $i = 1$, 2, 3, and 4, yielding the following

$$\Delta X = x_Q - x_P \tag{51}$$
$$\Delta Y = y_Q - y_P \tag{52}$$
$$\lambda = \frac{\Delta Y}{\Delta X} \tag{53}$$
$$\mu = y_P - \lambda x_P \tag{54}$$
$$b_1 = 2y_P f(P)(3x_P^2 + \alpha - 2\lambda y_P) \tag{55}$$
$$c_1 = 2y_P f_X(P) + (3x_P^2 + \alpha) f_Y(P) \tag{56}$$
$$b_2 = 2y_P y_R f(R)(3x_P^2 + \alpha - 2y_P\lambda)\Delta X \tag{57}$$
$$c_2 = -2\lambda y_P(2y_R f_X(R) + (3x_R^2 + \alpha) f_Y(R))(3x_P^2 + \alpha - 2y_P(\lambda + \Delta X)) \tag{58}$$
$$b_3 = 2y_Q f(Q)(3x_Q^2 + \alpha - 2\lambda y_Q) \tag{59}$$
$$c_3 = 2y_Q f_X(Q) + (3x_Q^2 + \alpha) f_Y(Q) \tag{60}$$
$$b_4 = \left(2y_Q y_R f(R)(3x_Q^2 + \alpha - 2\lambda y_Q)\Delta X\right) \tag{61}$$
$$c_4 = 2\lambda y_Q(2y_R f_X(R) + (3x_R^2 + \alpha) f_Y(R))(3x_Q^2 + \alpha - 2y_Q(\lambda + \Delta X)) \tag{62}$$

At this point, the verifier finds $(y_i - \lambda x_i - \mu)A_i = 1$ for each $i = 1, 2, \ldots, n$, otherwise the verifier would output 0 after this check. Thus, the verifier is assured each $A_i = (y_i - \lambda x_i - \mu)^{-1}$ with certainty. Then the verifier finds that $b_i B_i = c_i$ for each $i = 1$, 2, 3, and 4, otherwise the verifier would output 0 after this check. Thus, the verifier is assured that each of these satisfy $B_i = c_i / b_i$ with certainty. Lastly, the verifier finds that $B_1 + B_2 + B_3 + B_4 = \sum_i m_i A_i$. Hence, the verifier is assured that the following is satisfied with certainty.

$$\frac{c_1}{b_1} + \frac{c_2}{b_2} + \frac{c_3}{b_3} + \frac{c_4}{b_4} = \sum_i \frac{m_i}{y_i - \lambda x_i \mu}$$

Moreover, since the verifier computed these c_i, b_i as above, the verifier concludes the following.

$$\frac{2y_P f_X(P) + (3x_P^2 + \alpha) f_Y(P)}{2y_P f(P)(3x_P^2 + \alpha - 2\lambda y_P)} - \frac{2\lambda y_P(2y_R f_X(R) + (3x_R^2 + \alpha) f_Y(R))(3x_P^2 + \alpha - 2y_P(\lambda + \Delta X))}{2y_P y_R f(R)(3x_P^2 + \alpha - 2y_P\lambda)\Delta X}$$
$$+ \frac{2y_Q f_X(Q) + (3x_Q^2 + \alpha) f_Y(Q)}{2y_Q f(Q)(3x_Q^2 + \alpha - 2\lambda y_Q)} + \frac{2\lambda y_Q(2y_R f_X(R) + (3x_R^2 + \alpha) f_Y(R))(3x_Q^2 + \alpha - 2y_Q(\lambda + \Delta X))}{2y_Q y_R f(R)(3x_Q^2 + \alpha - 2\lambda y_Q)\Delta X}$$
$$= \sum_i \frac{m_i}{y_i - \lambda x_i \mu}$$

By Equations 93 and 94 in the Appendix, the verifier knows the following.

$$\frac{2y_P f_X(P) + (3x_P^2 + \alpha) f_Y(P)}{2y_P f(P)(3x_P^2 + \alpha - 2\lambda y_P)} = \delta(f(X,Y)) \mid_{(X,Y)=(x_P,y_P)}$$

$$\frac{2y_Q f_X(Q) + (3x_Q^2 + \alpha) f_Y(Q)}{2y_Q f(Q)(3x_Q^2 + \alpha - 2\lambda y_Q)} = \delta(f(X,Y)) \mid_{(X,Y)=(x_Q,y_Q)}$$

Also, the verifier knows the derivation d (and therefore the logarithmic derivative δ) on $K(\mathcal{E})$ over K extends to the function fields for the product variety $K(\mathcal{E}^2)$. Thus, $\delta(f(X_R, Y_R)) \mid_{(P,Q)}$ is as follows due to Equation 97.

$$\frac{2\lambda(f_X(R) + y'_R f_Y(R))}{(x_Q - x_P) f(R)} \left(\frac{y'_Q - \lambda - (x_Q - x_P)}{y'_Q - \lambda} - \frac{y'_P - \lambda + (x_Q - x_P)}{y'_P - \lambda} \right) = \delta\big(f(X,Y)\big) \mid_{(P,Q)} \quad (63)$$

where in this case, δ is the logarithmic derivative on on $K(\mathcal{E}^2)$. Since δ is a group homomorphism from the multiplicative subgroup of $K(\mathcal{E}^2)$ to the additive group of principal divisors, the verifier knows the sum of these three terms is certainly as follows.

$$\delta\big(f(P)f(Q)f(R)\big) = \delta\big(\widehat{f}(X_P, Y_P, X_Q, Y_Q)\big) \mid_{(X_P,Y_P,X_Q,Y_Q)=(x_P,y_P,x_Q,y_Q)}$$

Hence, the following holds.

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

Of course, the verifier also knows that, for the line ℓ interpolating P and Q, the equality $\sum_i \frac{m_i}{y_i - \lambda x_i \mu} = \sum_i \delta\big(\ell(X_i, Y_i)\big) \mid_{(X_i,Y_i)=(x_i,y_i)}$ 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.

$$\delta\big(\widehat{f}(X_P, Y_P, X_Q, Y_Q)\big) \mid_{(X_P,Y_P,X_Q,Y_Q)=(x_P,y_P,x_Q,y_Q)} = \delta\left(\prod_{i=1}^{n} \ell(X_i, Y_i)^{m_i} \right) \mid_{\{(X_i,Y_i)\}_i = \{(x_i,y_i)\}_i} \quad (64)$$

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). Hence, by computing the difference of both sides of Equation 64 and clearing denominators, we obtain a polynomial expression. In particular, the roots of the rational function are roots of the polynomial.

The common denominator is exactly $f_3(X_P) f_3(X_Q) f_3(X_R)$ as follows, where we have omitted the term $I(\mathcal{E}/K)$ in the numerator and denominator of f, with the understanding that both terms

are cosets modulo $I(\mathcal{E}/K)$.

$$f(P)f(Q)f(R) = \prod_i \ell(X_i, Y_i)^{m_i} \tag{65}$$

$$\frac{f_1(X_P) + Y_P f_2(X_P)}{f_3(X_P)} \cdot \frac{f_1(X_Q) + Y_Q f_2(X_Q)}{f_3(X_Q)} \cdot \frac{f_1(X_R) + Y_R f_2(X_R)}{f_3(X_R)} = \prod_i \ell(X_i, Y_i)^{m_i} \tag{66}$$

Given a fixed f and (P_i, m_i) pairs, the point pairs (P, Q) satisfying Equation 67 have coefficients in the vanishing set of the following polynomial.

$$\prod_{T \in \{P,Q,R\}} \big(f_1(X_T) + f_2(X_T)Y_T\big) - f_3(X_P)f_3(X_Q)f_3(X_R) \prod_i \ell(X_i, Y_i)^{m_i} \tag{67}$$

We note that it may be the case that $\mathrm{div}(f) \neq \sum_i m_i[P_i]$. In particular, $f_1(X) + Y f_2(X)$ and $f_3(X)$ both have some degree n', which may not match n. The Schwarz-Zippel lemma states that, if $g \in K[\{Z_i\}_i]$, S 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 $\frac{n}{|S|}$. In this case, the polynomial ring is $K[X_P, Y_P, X_Q, Y_Q]$, and the polynomial is as in Equation 67. The sample set size is $|K| \times (|K| - 1) \times \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$. Indeed, because one element each from K is used to uniquely determine the distinct values of x_P and x_Q, and these in term determine, up to sign, the corresponding y_P and y_Q. This allows us to encode the field elements y_P and y_Q with 2 bits after x_P and x_Q are known. Hence, $|S| = 4\,|G|\,(|G| - 1)$.

As $\deg(f) = n$, the probability that the polynomial of Equation 67 is killed by some (P, Q) in the event that $\mathrm{div}(f) \neq \sum_i m_i[P_i]$ is $\frac{n}{4|G|(|G|-1)}$. The probability the polynomial of Equation 67 evaluates to zero when evaluated at the random (P, Q), conditioned on the event that the polynomial of Equation 67 is not the zero polynomial, is at most $\frac{n}{4|G|(|\mathbb{G}|-1)}$.

Now we claim that if the polynomial in Equation 67 is killed by some P, Q such that the interpolating line is non-vertical and non-tangent, then every $g \in K(\mathcal{E})$ satisfies the following with respect to f and the point-multiplicity pairs (P_i, m_i), except with probability at most $\frac{n}{4|\mathbb{G}|(|\mathbb{G}|-1)}$.

$$\prod_{j=1}^{m} f(w_j, z_j)^{\eta(w_j, z_j)} = \prod_{i=1}^{n} g(x_i, y_i)^{m_i} = \widehat{f}(\mathrm{div}(g)) \tag{68}$$

where $\mathrm{div}(g) = \sum_j \eta_j[S_j]$ and each $S_j = (w_j, z_j) \in K^2$. That is to say, we claim it is sufficient to only check f against one line g, except with except with probability at most $\frac{n}{4|\mathbb{G}|(|\mathbb{G}|-1)}$. To see why, consider that Weil's reciprocity is defined over the algebraic closure, in which any $g \in K(\mathcal{E})$ splits into linear factors, and div is a homomorphism. □

3.3 Computational refinements

In the generic divisors framework we present, 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 $(\mathbb{G} \setminus \{O\})^2$, which can model a PRNG (really, a cryptographically secure pseudo-random *elliptic curve point* generator, but this CSPRECPG may be modeled with a PRNG and a hash-to-point method), but care must be taken to avoid vulnerabilities with the weak Fiat-Shamir transform [NHB24, BPW16, DMWG23]: all public parameters must be included in the hash input. To avoid vulnerabilities like the so-called Frozen Heart attack [Jon, Mil], the hash digest must have been generated from the commitments, previous randomness, user tags, counters, etc.

With this in mind, we present a standard improvement utilizing 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.

4 Applications

In this section, we highlight a number of practical use cases. By utilizing divisors, these applications benefit from a marked computational improvement to the verifier's checks, however they also enjoy the zero-knowledge property from their underlying zero-knowledge proof.

In this next section, we use $\star$ to denote the entrywise vector product, alternately referred to as 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 will be used to ensure 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 to denote 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.

4.1 Schnorr identification

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 2 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 the divisors approach is ever zero-knowledge, the Schnorr protocol enjoys this feature, and hence, this protocol can represent 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: $s = (s_0, \ldots, s_k) \in K^n$ such that $a = s_0 + s_1 2 + \cdots + s_k 2^k$	
Prover	**Verifier**
Sample $(r_0, \ldots, r_k) \xleftarrow{\$} K$	
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 $z = s + cr$	
Set $\nu = \sum_{i=0}^{k} z_i \cdot (B_i) - (P) - c \cdot (R)$	
$\pi \leftarrow \texttt{Prove}(\nu)$	$\xrightarrow{\pi}$
	Recompute ν
	Check: $\texttt{Verify}(\nu, \pi) = 1$

Algorithm 2: A computational improvement to the verification step in the Schnorr protocol.

In Algorithm 2, 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 with the exhibited communication of the proof π. Then the verifier's check in the above portion - which traditionally requires $k + 2$ 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.

Theorem 5. Algorithm 2 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. There will be a nominal change to the completeness and soundness error, due to the arguments presented in Section 3.2 (specifically Theorems 3 and 4), but this alteration is negligible in $|\mathbb{G}|$, so does not cause problems. □

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 will therefore be heavily reduced at the cost of a negligible 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 will be verified a great many times, so it behooves the protocol designers to offload the verifier's 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.

Mention [CS25] somewhere...

4.2 Bulletproofs range proofs

We point out that these ideas may 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 number $v \in K$, for $\text{char}(K) = p$ large enough compared to n, is constrained to some interval $[0,\, 2^n - 1]$. Specifically, it provides a proof of the following relation:

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

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

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

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 setup protocol for Bulletproofs in Algorithm 3, which serves as a precomputation step to the main proof generation and verification protocols in Algorithm 4.

Bulletproofs Setup Protocol		
Public: $g, h \in \mathbb{G}, \quad G, H \in \mathbb{G}^n$ **Private**: $v \in K$		
PROVER		**VERIFIER**
Compute $a_L \in \{0,\, 1\}^n$ s.t. $\langle a_L,\, \vec{2}\rangle = v$ Set $a_R = a_L - \mathbb{1}$ 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 $y, z \xleftarrow{\$} K^\times$

Algorithm 3: The setup algorithm for Bulletproofs.

Bulletproofs Proof Protocol		
Public: $g, h \in \mathbb{G}, \quad G, H \in \mathbb{G}^n$ **Private**: $v \in K^n$		
PROVER		**VERIFIER**
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 $x_0 \xleftarrow{\$} K^*$
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 $v_1 = (A) + x_0(S) - (z - L)(G)$ $\quad +(z\vec{y} + z^2\vec{2} - R)(H') - \mu(h)$ 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 $\pi_2 \leftarrow \texttt{Prove}(v_2)$	$\xrightarrow{\pi_1, \pi_2}$	
		Recompute ν_1 and ν_2 Check $\texttt{Verify}(v_1,\, \pi_1) = 1$ Check $\texttt{Verify}(v_2,\, \pi_2) = 1$ Check $t \stackrel{?}{=} \langle L,\, R\rangle$

Algorithm 4: The proof generation and verification procedure for Bulletproofs.

Theorem 6. The Bulletproofs range proof presented in Algorithm 4 enjoys completeness, witness-

extended emulation (a more robust notion of soundness), and is zero-knowledge.

For a proof of this claim, 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, so can be performed efficiently - this is the last line on the verifier's side of the protocol. 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, these $6n+11$ point multiplications can be reduced to divisor additions, which may be verified much more efficiently than computing the elliptic curve multiplications.

A 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.

A.1 Logarithmic derivative of function field elements at the zeroes of a line

In this section, we formalize taking the logarithmic derivative of $f(P)f(Q)f(R)$ where $P+Q+R = O$ in the group law. Indeed, for any $T \in \mathcal{E}$, $f(T)$ is just an element of K, and so $dT = 0$. However, if P, Q, R are handled as variable points on the curve $\mathcal{E}$, say by introducing indeterminates X_P, Y_P, X_Q, Y_Q, X_R, and Y_R, then a derivation on $K(\mathcal{E})$ extends naturally to a derivation on $K(\mathcal{E}^3)$. However, given P and Q, there is a unique R satisfying $P + Q + R = O$, so we only need indeterminates X_P, Y_P, X_Q, and Y_Q.

Note that although these are all indeterminates over K, some Y is not indeterminate over $K[X]$. Indeed, Y is uniquely determined by X up to sign. Thus, we could instead use indeterminates X_P, S_P, X_Q, S_Q for indeterminates S_P and S_Q acting as variables for sign bits. However, this can significantly complicate the following formulae. Thus, despite that each Y is almost uniquely determined by each X, we still have four degrees of freedom in the choice of P and Q by using X_P, Y_P, X_Q, and Y_Q, and we instead treat X_R and Y_R as functions of P and Q.

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 $Q = (X_Q, Y_Q)$.

$$\Delta X = X_Q - X_P \tag{69}$$

$$\Delta Y = Y_Q - Y_P \tag{70}$$

$$\Lambda = \frac{\Delta Y}{\Delta X} \tag{71}$$

$$\begin{aligned} d\Lambda =& \frac{\Delta X d\Delta Y - \Delta Y d\Delta X}{(\Delta X)^2} && (72)\\ =& \frac{d\Delta Y - \Lambda d\Delta X}{\Delta X} && (73)\\ =& \frac{(dY_Q - \Lambda dX_Q) - (dY_P - \Lambda dX_P)}{\Delta X} && (74)\\ =& \frac{\left(Y'_Q - \Lambda\right) dX_Q - \left(Y'_P - \Lambda\right) dX_P}{\Delta X} && (75) \end{aligned}$$

$$M = Y_P - \Lambda X_P \tag{76}$$

$$\begin{aligned} dM =& dY_P - \Lambda dX_P - X_P d\Lambda && (77)\\ =& \left(Y'_P - \Lambda\right) dX_P - X_P d\Lambda && (78)\\ =& \left(Y'_P - \Lambda\right) dX_P - X_P \frac{\left(Y'_Q - \Lambda\right) dX_Q - \left(Y'_P - \Lambda\right) dX_P}{\Delta X} && (79)\\ =& \left(Y'_P - \Lambda\right) dX_P - \frac{X_P}{\Delta X}\left(Y'_Q - \Lambda\right) dX_Q + \frac{X_P}{\Delta X}\left(Y'_P - \Lambda\right) dX_P && (80)\\ =& \left(1 + \frac{X_P}{\Delta X}\right)\left(Y'_P - \Lambda\right) dX_P - \frac{X_P}{\Delta X}\left(Y'_Q - \Lambda\right) dX_Q && (81)\\ =& -\frac{X_P}{\Delta X}\left(Y'_Q - \Lambda\right) dX_Q + \frac{X_Q}{\Delta X}\left(Y'_P - \Lambda\right) dX_P && (82) \end{aligned}$$

$$\begin{aligned} X_R =& \Lambda^2 - X_P - X_Q && (83)\\ Y_R^2 =& X_R^3 + \alpha X_R + \beta && (84)\\ R =& (X_R, Y_R) && (85) \end{aligned}$$

$$\begin{aligned} dX_R =& 2\Lambda d\Lambda - dX_P - dX_Q && (86)\\ =& \frac{2\Lambda}{\Delta X}\Big(\left(Y'_Q - \Lambda\right) dX_Q - \left(Y'_P - \Lambda\right) dX_P\Big) - dX_P - dX_Q && (87)\\ =& \frac{2\Lambda}{\Delta X}\left(Y'_Q - \Lambda - \Delta X\right) dX_Q - \frac{2\Lambda}{\Delta X}\left(Y'_P - \Lambda + \Delta X\right) dX_P && (88) \end{aligned}$$

$$dY_R = \frac{dY_R}{dX_R} dX_R \tag{89}$$
$$= \frac{3X_R^2 + \alpha}{2Y_R} dX_R \tag{90}$$
$$= Y_R' dX_R \tag{91}$$

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.

$$df(P) = f_X(P) dX_P + f_Y(P) dY_P \tag{92}$$
$$= (f_X(P) + f_Y(P) Y_P') dX_P \tag{93}$$
$$df(Q) = (f_X(Q) + f_Y(Q) Y_Q') dX_Q \tag{94}$$

$$df(R) = f_X(R) dX_R + f_Y(R) dY_R \tag{95}$$
$$= (f_X(R) + Y_R' f_Y(R)) dX_R \tag{96}$$
$$df(R) = \frac{2\Lambda(f_X(R) + Y_R' f_Y(R))}{\Delta X} \Big(\left(Y_Q' - \Lambda - \Delta X\right) dX_Q - \left(Y_P' - \Lambda + \Delta X\right) dX_P \Big) \tag{97}$$

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

$$\delta(\widehat{f}) = \frac{df(P)}{f(P)} + \frac{df(Q)}{f(Q)} + \frac{df(R)}{f(R)} \tag{98}$$
$$= \frac{df(P)}{f(P)} - \frac{2\Lambda(f_X(R) + Y_R' f_Y(R))(Y_P' - \Lambda - \Delta X)}{f(R)\Delta X} dX_P \tag{99}$$
$$+ \frac{df(Q)}{f(Q)} + \frac{2\Lambda(f_X(R) + Y_R' f_Y(R))(Y_Q' - \Lambda - \Delta X)}{f(R)\Delta X} dX_Q \tag{100}$$
$$= \left(\frac{f_X(P) + Y_P' f_Y(P)}{f(P)} - \frac{2\Lambda(f_X(R) + Y_R' f_Y(R))(Y_P' - \Lambda - \Delta X)}{f(R)\Delta X} \right) dX_P \tag{101}$$
$$+ \left(\frac{f_X(Q) + Y_Q' f_X(Q)}{f(Q)} + \frac{2\Lambda(f_X(R) + Y_R' f_Y(R))(Y_Q' - \Lambda - \Delta X)}{f(R)\Delta X} \right) dX_Q \tag{102}$$

We are particularly interested in the case that $d(Y - \lambda^* X - \mu^*) = 1$ for some fixed $(\lambda^*, \mu^*) \in K^2$.

$$1 = dY - \lambda dX \tag{103}$$
$$1 = (Y' - \lambda)dX \tag{104}$$
$$dX = (Y' - \lambda)^{-1} \tag{105}$$

Note then that $d\Lambda = dM = 0$, and $\delta(\widehat{f})$ becomes the following.

$$\delta(\widehat{f}) = \left(\frac{f_X(P) + Y'_P f_Y(P)}{f(P)} - \frac{2\Lambda(f_X(R) + Y'_R f_Y(R))(Y'_P - \Lambda - \Delta X)}{f(R)\Delta X}\right) dX_P \tag{106}$$
$$+ \left(\frac{f_X(Q) + Y'_Q f_X(Q)}{f(Q)} + \frac{2\Lambda(f_X(R) + Y'_R f_Y(R))(Y'_Q - \Lambda - \Delta X)}{f(R)\Delta X}\right) dX_Q \tag{107}$$
$$= \left(\frac{f_X(P) + Y'_P f_Y(P)}{f(P)} - \frac{2\Lambda(f_X(R) + Y'_R f_Y(R))(Y'_P - \Lambda - \Delta X)}{f(R)\Delta X}\right) (Y'_P - \lambda^*)^{-1} \tag{108}$$
$$+ \left(\frac{f_X(Q) + Y'_Q f_X(Q)}{f(Q)} + \frac{2\Lambda(f_X(R) + Y'_R f_Y(R))(Y'_Q - \Lambda - \Delta X)}{f(R)\Delta X}\right) (Y'_Q - \lambda^*)^{-1} \tag{109}$$

This is the left-hand side of the verification equation. We can now evaluate at any $(X_P, Y_P, X_Q, Y_Q) = (x_P, y_P, x_Q, y_Q)$, 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.

$$\delta(\widehat{f}) \mid_{P,Q} = \frac{f_X(P) + y'_P f_Y(P)}{f(P)(y'_P - \lambda^*)} - \frac{2\lambda^*(f_X(R) + y'_R f_Y(R))(y'_P - \lambda - \Delta X)}{f(R)\Delta X(y'_P - \lambda)} \tag{110}$$
$$+ \frac{f_X(Q) + y'_Q f_X(Q)}{f(Q)(y'_Q - \lambda)} + \frac{2\lambda^*(f_X(R) + y'_R f_Y(R))(y'_Q - \lambda - \Delta X)}{f(R)\Delta X(y'_Q - \lambda)} \tag{111}$$

A.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 is as in Equation (71) and M is as in Equation (76). 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 $Q = (x_Q, y_Q)$.

$$\underline{\ell} = \prod_{i=1}^{n} \widetilde{\ell}_i(T_i)^{\nu_i} \tag{112}$$

$$= \prod_{i=1}^{n} (y_i - \Lambda x_i - M)^{\nu_i} \tag{113}$$

$$\delta(\underline{\ell}) = \sum_{i=1}^{n} \nu_i \frac{d\,(y_i - \Lambda x_i - M)}{y_i - \Lambda x_i - M} \tag{114}$$

$$= \sum_{i} \nu_i \frac{dy_i - \Lambda dx_i - x_i d\Lambda - dM}{y_i - \Lambda x_i - M} \tag{115}$$

$$= \sum_{i} \nu_i \frac{(y_i' - \Lambda) dx_i - x_i d\Lambda - dM}{y_i - \Lambda x_i - M} \tag{116}$$

$$= \sum_{i} \nu_i \frac{1 - x_i d\Lambda - dM}{y_i - \Lambda x_i - M} \tag{117}$$

$$= \sum_{i} \nu_i (y_i - \Lambda x_i - M)^{-1} \tag{118}$$

where we refer to Equations (71), (76), (75), and (82) 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.

References

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

[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.

[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`.

[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.

[CS25] Elizabeth Crites and Alistair Stewart. A Plausible Attack on the Adaptive Security of Threshold Schnorr Signatures. Cryptology ePrint Archive, Paper 2025/1001, 2025.

[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.

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

[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.

[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.

[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.

[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++. 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.

[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.