Optimizing Halo and Constructing Graphs of Elliptic Curves

Daira Hopwood
<daira@electriccoin.co>
@feministPLT
@daira on Discord
Slides at https://github.com/daira/ecgraphs

Outline

- Background
  - What is a graph of elliptic curves? Why are they useful?
  - Constructing cycles – the problem
  - CM curves to the rescue
- Detail of curve construction
  - Automorphisms
  - The CM norm equation
  - 2-adicity: why we need it and how to get it
  - Cycles and chains
  - Speeding up searches
  - Pairings, half-pairing cycles, and inverted lollipops
  - Live coding in Sagemath
- Halo optimizations
  - Understanding Halo/Sonic/Bulletproofs arithmetization
  - Scalar multiplication in circuits
  - Optimized scalar multiplication
  - Really optimized scalar multiplication
  - Fiat–Shamir and duplex sponges
  - Algebraic hashes (Rescue).

What is a graph of elliptic curves?

- What is a directed graph?
  - Set of vertices, set of directed edges
  - Our vertices will be primes p, q, …
  - Our edges will be elliptic curves $E_{p \to q}$.
- What is the curve $E_{p \to q}$?
  - Points have coordinates in the *field of definition* $\mathbb{F}_p$.
  - There are n points forming a group, with a prime subgroup of order q dividing n.
  - The *scalar field* is $\mathbb{F}_q$.
  - We'll assume that a proof system using $E_{p \to q}$ efficiently supports circuits with arithmetic over $\mathbb{F}_q$.

Examples

- Diagram here

Motivation for cycles

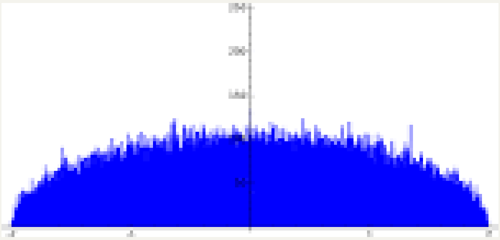
- Wrong-field arithmetic has an overhead of ~1000 times.
  - This is using the sum of residues method for reduction. Zcash #4093
- Doesn’t Plookup solve this?
  - No; Plookup helps but wrong-field arithmetic probably still has an overhead of 10-20 times.
  - Plookup works best for larger circuits, where the cost of tables can be amortized. For Halo we want to minimize the cost of the recursion subcircuit.
  - Nothing here conflicts with using plookup for other things in the same proof system.

The Tweedle cycle

- The Halo paper gives a pair of curves:
  - $E_{p \to q} : y^2 = x^3 + 5$ is called Tweedledum.
  - $E_{q \to p} : y^2 = x^3 + 5$ is called Tweedledee.
  - $p = 2^{254} + \text{0x38AA1276C3F59B9A14064E200000001}$
  - $q = 2^{254} + \text{0x38AA127696286C9842CAFD400000001}$
  - Both have 126-bit Pollard rho security, maximal embedding degree.
  - They have cubic endomorphisms (we'll explain what that means).
  - $\gcd(p - 1, 5) = \gcd(q - 1, 5) = 1$
- We're going to explain the construction that found them, and some generalizations of it.

Constructing cycles – the problem

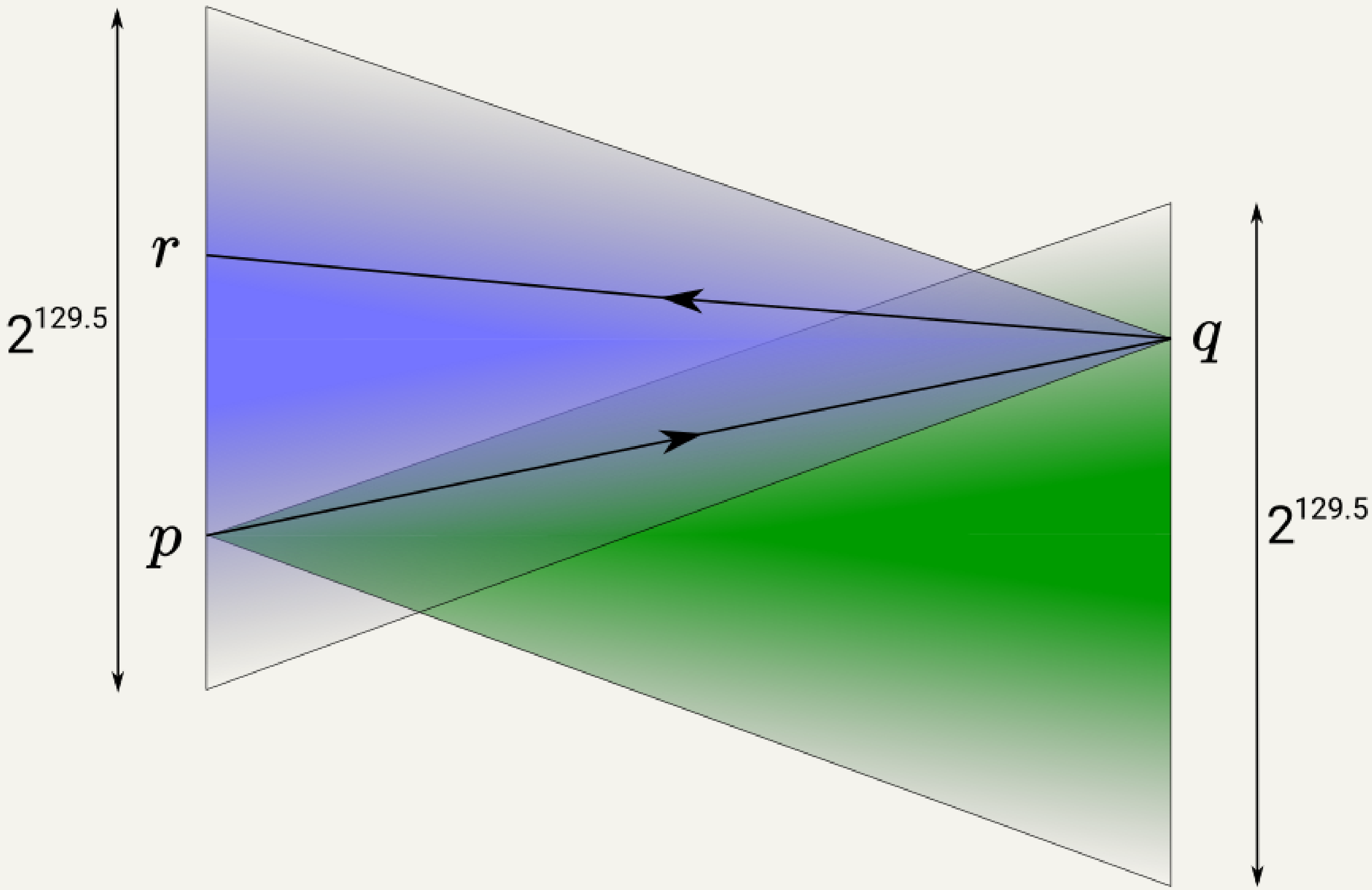
- By the Hasse bound, the order of an elliptic curve over $\mathbb{F}_p$ lies in a range of size ~$4\sqrt{p}$. For Tweedle this is ~$2^{129.5}$.
- The Sato–Tate conjecture concerns the distribution of the order in this range. We don't need to go into detail, but here's a picture:



- That is, the order n could be anywhere in the range.
- And (if n is a prime q) when we construct a curve $E_{q \to r}$ it could also have order anywhere in its Hasse range.
- So, it's exceptionally unlikely that $E_{q \to r}$ has order p.

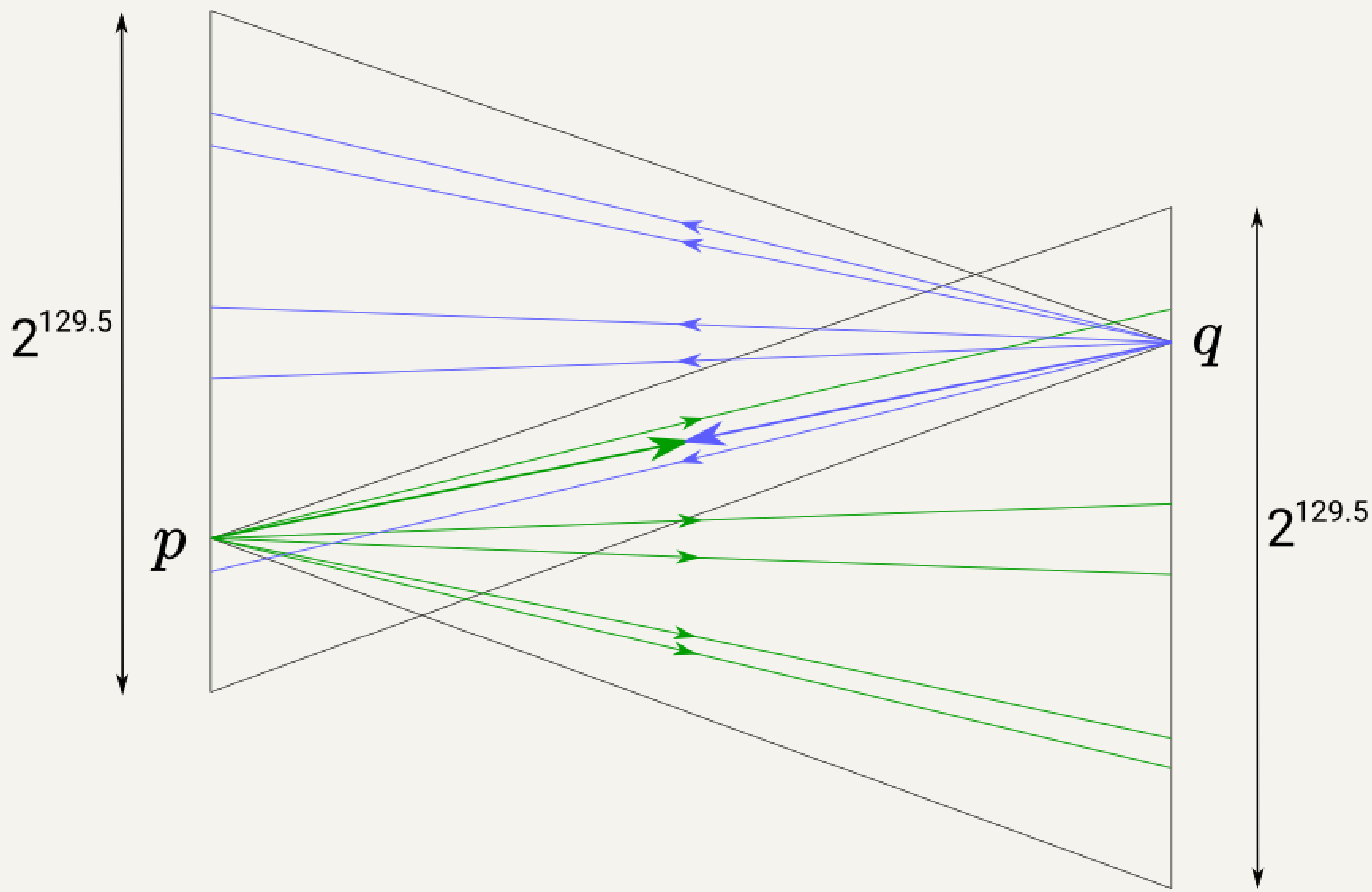
Constructing cycles – the problem

- By the Hasse bound, the order of an elliptic curve over $\mathbb{F}_p$ lies in a range of size ~$4\sqrt{p}$. For Tweedle this is ~$2^{129.5}$.



Constructing cycles – the solution

- Suppose we were able to restrict the orders to a small number of possibilities, one of which was guaranteed to form a cycle...



CM curves to the rescue

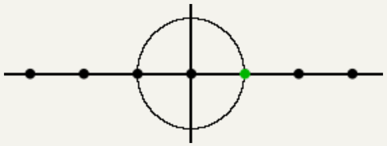
- Katherine Stange and Joe Silverman noticed that CM curves have precisely that property [SS2011].
- What is a CM curve?
  - This is not intended to be a fully precise definition, just to give intuition and make the concept less mysterious.
- All curves have an “endomorphism ring”. An endomorphism is a group homomorphism (meaning that it preserves the group structure) from the curve group to itself.
  - Why is it a ring? Because you can compose and “add” endomorphisms, and there is an identity endomorphism.
- An example of an endomorphism in an elliptic curve group is scalar multiplication by a constant integer. We can think of endomorphisms as being generalized scalars.

Complex multiplication

- All endomorphisms of an elliptic curve *over a finite field* are equivalent to ordinary scalar multiplication by an integer.
- But an elliptic curve over $\mathbb{F}_p$ is a reduction of a curve with the same equation over the complex numbers $\mathbb{C}$.
- "Complex multiplication" refers to scalar-multiplying points in the curve over $\mathbb{C}$ by complex numbers.
  - E.g. consider $E : y^2 = x^3 + ax$ and let $[i]\,(x, y) = (-x, iy)$. Then $[i^2]\,(x, y) = [-1]\,(x, y) = (x, -y)$, which is the same as applying the $[i]$ map twice.
  - So scalars are numbers in a complex lattice such as $\mathbb{Z}[i]$ or $\mathbb{Z}[\sqrt[3]{1}]$.
- There are only 13 elliptic curves <u>over $\mathbb{C}$</u> with complex multiplication, up to isomorphism. So these are a very small fraction of all curves.
- Are yous still with me? If not then don't worry because it all gets simpler again when we map back to $\mathbb{F}_p$.

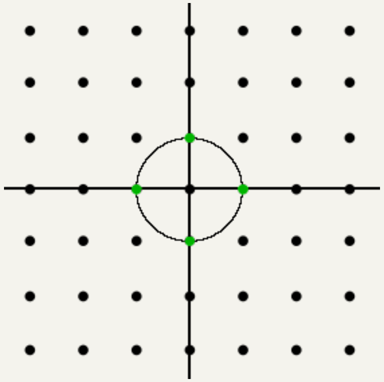
Structure of the endomorphism ring

- We said that a generalized scalar can be a number in a complex lattice.
- An elliptic curve over $\mathbb{C}$ has CM if that lattice has more than one dimension (i.e. it's bigger than $\mathbb{Z}$). The lattice structure depends on something called the curve's j-invariant.
- It's kinda like this (an integer lattice, $j \notin \{0, 1728\}$):



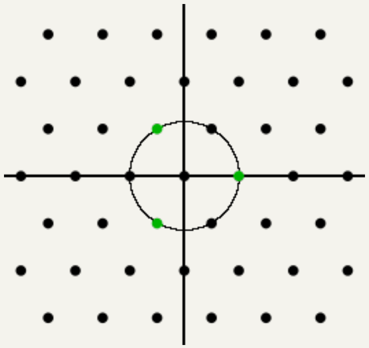
- Or like this (a quadratic lattice $\mathbb{Z}[i]$, $j = 1728$):

  The example we saw on the previous slide is of this case.



- Or like this (a hexagonal lattice $\mathbb{Z}[\sqrt[3]{1}]$, $j = 0$):

  The hexagonal case turns out to be really nice for cryptography. This is the case that secp256k1, used in Bitcoin, falls into.



But what does it all mean?

- A consequence of … is that a CM curve has only a small number of possible orders.
- Which orders it can have is dependent on its j-invariant.
- In particular, curves with equation $y^2 = x^3 + b$ (i.e. with no x or x^2 terms) have j-invariant 0.
- These curves are interesting because, over Fp,
  - they have only 6 possible orders;
  - they have efficiently computable endomorphisms;
  - it’s easy to solve the CM norm equation (next slide);
- Some of the possible orders are likely to form cycles.

The CM norm equation

- $|D|V^2 = 4p - T^2$
- $|D|$ is the (absolute) discriminant
- p is the field size
- V and T are integers
- V and T determine the trace t, where $q = p + 1 - t$.
- In fact $\pm T$ are two of the possible traces.
- The Tweedle curves were found using this construction:
  - set $|D| = 3$, pick V and T, find $p = ¼\,(|D|V^2 + T^2)$
  - Later we'll see other constructions that work in other situations.
- The reason for this approach is that we can choose V and T so that *both* curves in the cycle have high 2-adicity.

The CM norm equation

- The norm equation helps to explain why CM curves form cycles.
- I’ll just go through the $\pm T$ case. For $j = 0$ there are other cases which are very similar.
-

2-adicity: why we need it and how to get it

- Protocols that use Lagrange basis, or that need to efficiently multiply polynomials, benefit from $\mathbb{F}_p^*$ having a “large enough” multiplicative subgroup of size b^k. The simplest option is b = 2.
- In other words, we need $p \equiv 1 \pmod{2^k}$.
- We can freely choose V and T. So let’s choose ½(V-1) and ½(T-1) to be multiples of $2^{k/2}$.

Cycles and chains

Speeding up searches

Pairings and half-pairing cycles

Inverted lollipops

Live coding in Sagemath

Halo/Sonic/Bulletproofs arithmetization

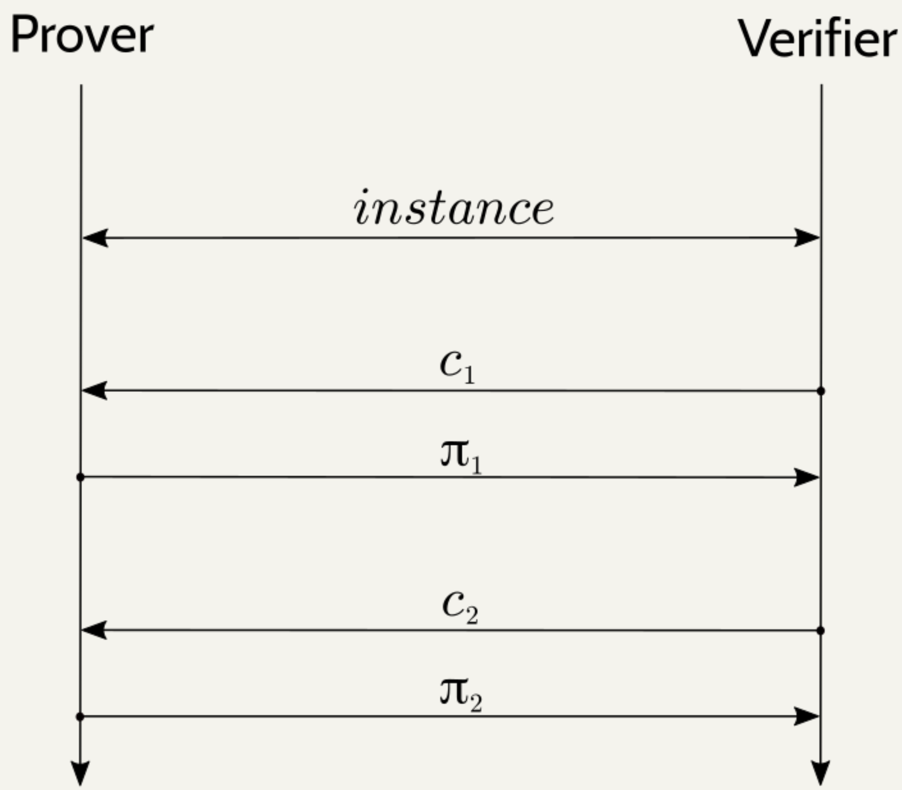
Scalar multiplication in circuits

Optimized scalar multiplication

Really optimized scalar multiplication

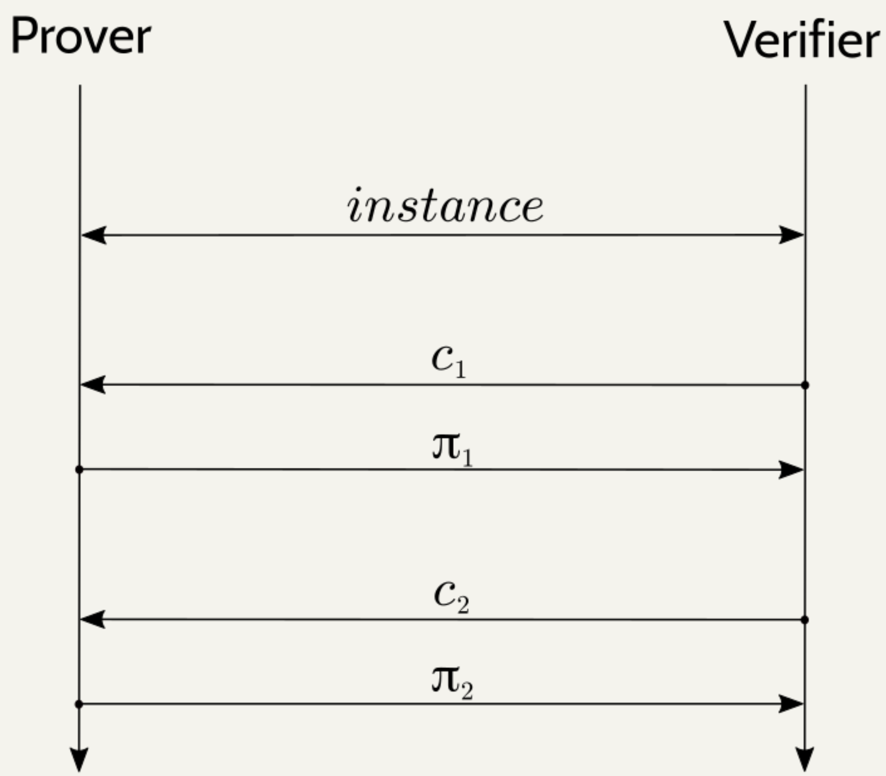
Fiat–Shamir and duplex sponges

- The Fiat–Shamir construction takes an interactive public-coin protocol, …



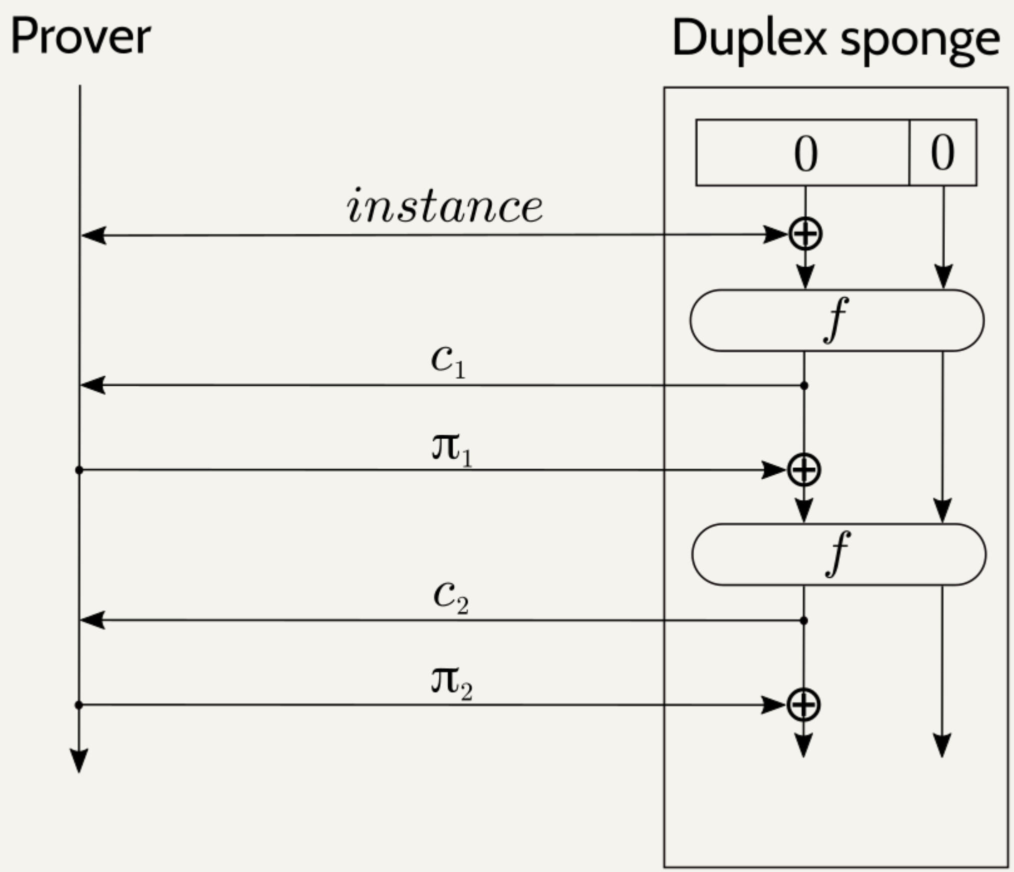
Fiat–Shamir and duplex sponges

- The Fiat–Shamir construction takes an interactive public-coin protocol, ...



Fiat–Shamir and duplex sponges

- The Fiat–Shamir construction takes an interactive public-coin protocol, and replaces the verifier with a hash function.



- Using a duplex sponge basically halves the number of f evaluations relative to other hash constructions.

Fiat–Shamir optimizations

- Compress the absorbed inputs.
  - there’s a way of probabilistically compressing two curve points to three field elements that is much less expensive than standard point compression (see accompanying notes).
- Pick a “rate” for the duplex sponge that is just large enough that we only need one f evaluation per round.
  - For the inner product argument we need $\lg(N)$ rounds, each of which absorbs two curve points and squeezes out one challenge.
-

Algebraic hashes

- To instantiate f in the duplex sponge, we need a permutation that is efficient in the circuit.
- Rescue is a permutation designed to be efficient in circuits over a prime field.
  - We're also considering Poseidon for the next version.
- Optimizations that we can use with either Rescue or Poseidon:
  - Use addition in the field for ⊕, rather than XOR.
  - Express the input in as few field elements as possible.
  - Set the "rate" to be the number of field elements we need to absorb in each round.
  - Choose curves with gcd(p−1, 5) = 1 so that $x \mapsto x^5$ is a permutation.

Questions

Scalable Privacy

Daira Hopwood
<daira@electriccoin.co>
@feministPLT
@daira on chat.zcashcommunity.com
Slides at https://github.com/daira/zcon