GENERATION OF ELLIPTIC CURVES FOR CIRCUIT USE

Barry WhiteHat, Jordi Baylina
and Marta Bellés

MOTIVATION

MOTIVATION

Elliptic Curve E

SNARK

MOTIVATION

Elliptic Curve E

SNARK

EdDSA

MOTIVATION

Elliptic Curve E

SNARK

Elliptic Curve E'

EdDSA

MOTIVATION

BN128

SNARK

Elliptic Curve E'

EdDSA

MOTIVATION

BN128

SNARK

???

EdDSA

PLAN

PLAN

E pairing friendly elliptic curve
of order p

PLAN

BN128 $y^2 = x^3+3$

21888242871839275222246405745257275088548364400416034343698204186575808495617

PLAN

BN128 $y^2 = x^3+3$

21888242871839275222246405745257275088548364400416034343698204186575808495617

`findCurve`

E' defined over $\mathbb{F}_p$

PLAN

BN128 $y^2 = x^3+3$

21888242871839275222246405745257275088548364400416034343698204186575808495617

findCurve

Baby JubJub

PLAN

BN128 $y^2 = x^3+3$

21888242871839275222246405745257275088548364400416034343698204186575808495617

↓

`findCurve`

↓

Baby JubJub

↓

`verify` Check it is secure

PLAN

BN128 $y^2 = x^3+3$

21888242871839275222246405745257275088548364400416034343698204186575808495617

findCurve

Baby JubJub

verify

findCurve

findCurve

WHAT DO WE MEAN?

- Finite field (prime p)
- Equation
- Order
- Generator
- Base point

findCurve

WHAT DO WE MEAN?

- Finite field (prime p)
- Equation
- Order
- Generator
- Base point

WHAT DO WE WANT?

1. Twisted Edwards / Montgomery
2. Deterministically generated
3. Safety criteria is satisfied `verify`

findCurve

1. TWISTED EDWARDS/MONTGOMERY

Twisted Edwards $ax^2+y^2=1+dx^2y^2$

ONE formula for addition of points

Birational equivalence

Montgomery $By^2=x^3+Ax^2+x$

findCurve

2. DETERMINISTIC GENERATION

RFC 7748: Elliptic Curves for Security (2016)
S. Turner, M. Hamburg and A. Langley

Montgomery $By^2=x^3+Ax^2+x$

- B = 1
- Smallest A>2 such that A-2 is divisible by 4

→ Twisted Edwards

findCurve

2. DETERMINISTIC GENERATION

https://github.com/barryWhiteHat/baby_jubjub/blob/master/findCurve.sage

findCurve

2. DETERMINISTIC GENERATION

```
def findCurve(prime, curveCofactor, twistCofactor, _A):

        F = GF(prime)
        A = _A

        while A < _A + 100000:

                if (A-2.) % 4 != 0:
                        A+=1.
                        continue
        try:
                E = EllipticCurve(F, [0, A, 0, 1, 0])
        except:
                A+=1.
                continue

        groupOrder = E.order()

        if (groupOrder % curveCofactor != 0
           or not is_prime(groupOrder // curveCofactor)):
                A+=1
                continue

        twistOrder = 2*(prime+1)-groupOrder

        if (twistOrder % twistCofactor != 0
           or not is_prime(twistOrder // twistCofactor)):
                A+=1
                continue

        return A, E
```

verify

verify

3. SAFETY CRITERIA

SafeCurves: Daniel J. Bernstein and Tanja Lange

Curve parameters	ECDLP security	ECC security

https://github.com/barrywhitehat/baby_jubjub/blob/master/verify.sage