Cryptocoin for Swift

A modular framework for Bitcoin, Litecoin, Dogecoin, etc. written in Swift (and a bit of C)

github.com/CryptoCoinSwift

Work in progress. Inspired by CryptoCoinJS.

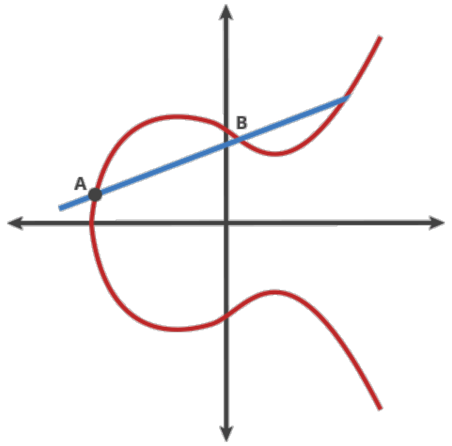
Sjors Provoost / sjors@purpledunes.com / @provoost

August 5th, 2014: Dutch Ethereum & Bitcoin meetup Amsterdam

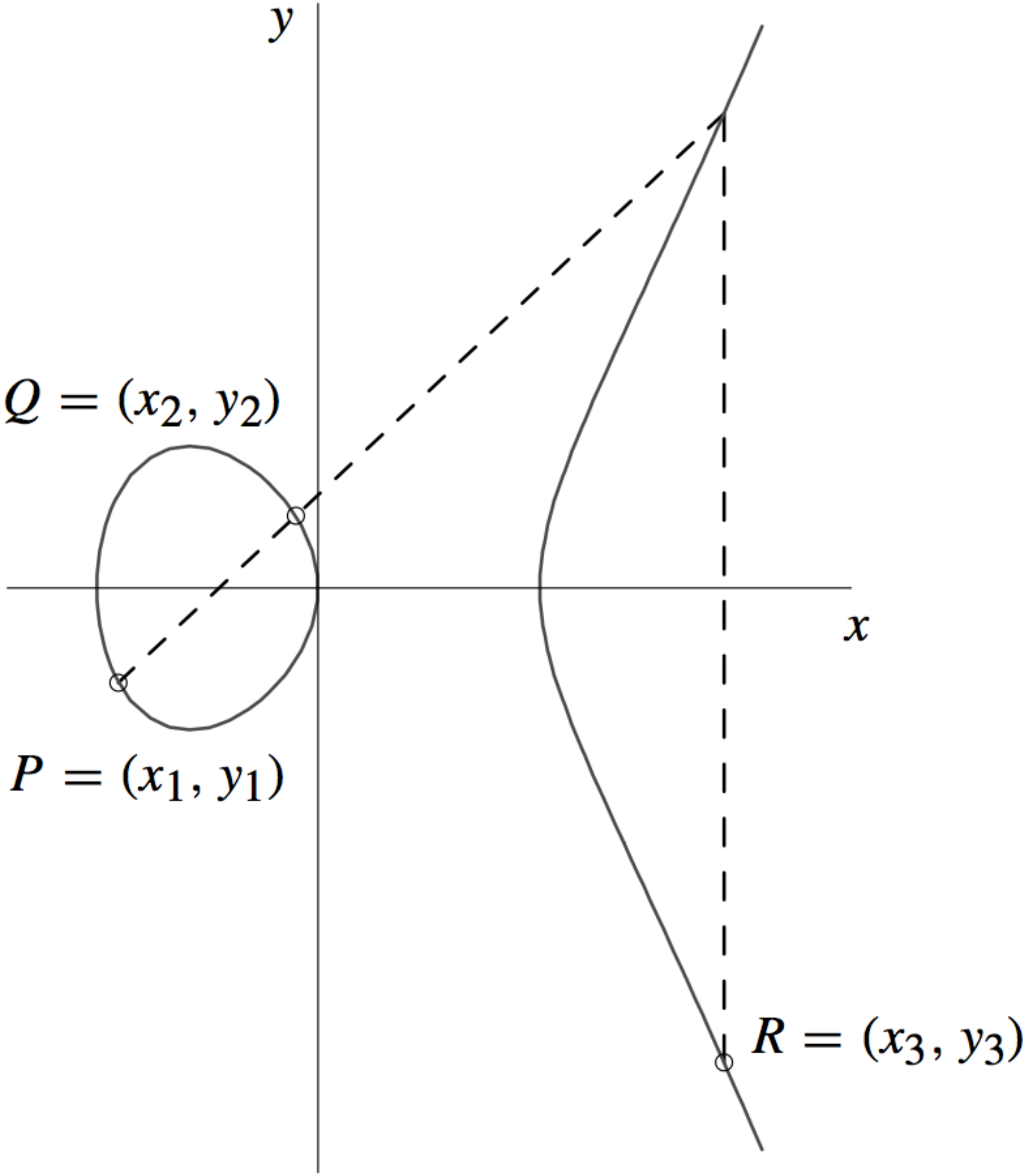
Why build your own framework?

- Native iPhone app
- Learn Swift
- Learn Bitcoin
- Learn elliptic curve cryptography

- **UInt256** 0...115792089237316195423570985008687907853269984665640564039457584007913129639935
- **FFInt** Finite field math: (6 + 6) mod 10 = 2
- **ECPoint** A = (x, y)



- **ECurve** A + B
- **ECKey** General crypto: public key, signature..
- **CoinKey** Crypto currency specific: address, WIF..
- **Bitcoin** Subclass of CoinKey with 0x80 prefix..

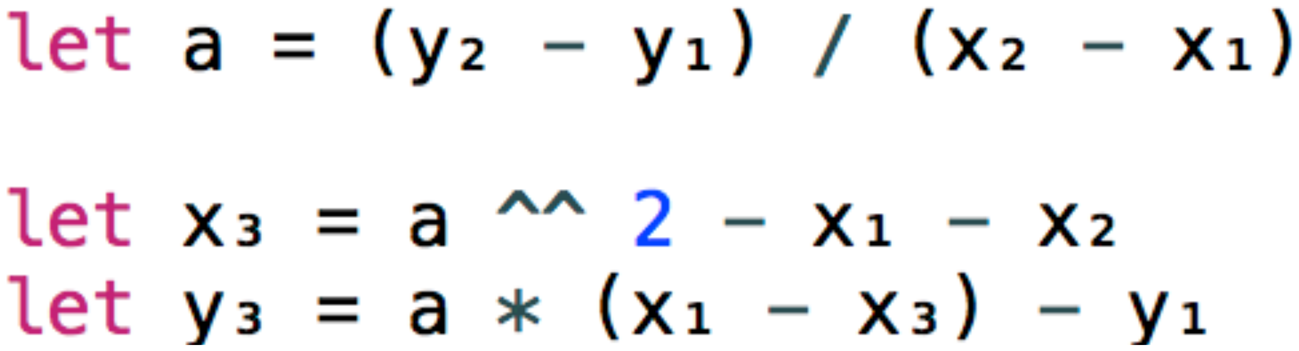


Point addition

The book

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$$
$$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1$$

Swift



```
let a = (y₂ - y₁) / (x₂ - x₁)

let x₃ = a ^^ 2 - x₁ - x₂
let y₃ = a * (x₁ - x₃) - y₁
```

Operator overload fest

```
let Q = d * P // d is big number, P is a point on a curve

// Handles multiplying a big number with point on curve
func * (lhs: UInt256, rhs: ECPoint) -> ECPoint { ...
  let a = (y2 - y1) / (x2 - x1)

// Multiplication modulo (finite field)
func * (lhs: FFInt, rhs: FFInt) -> FFInt {
  let product: (UInt256, UInt256) = lhs.value * rhs.value
  return field.int(product % p.p)
}

// Multiply two 256 bit integers:
func * (lhs: UInt256, rhs: UInt256) -> (UInt256, UInt256) { ... }
```

The Future - What I need

- 4x faster
- ECDSA (signature)
- Generate a bitcoin transaction
- Submit transaction to cloud
- Fetch blockchain data from cloud

The Future - For others

- Altcoins
- Different currencies like Etherium
- Blockchain & network : full client
- Mining? :-)

Cryptocoin Swift

https://github.com/CryptoCoinSwift/CryptoCoinFramework

Sources

Animating curve from CloudFlare Blog:

http://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography

Book: Guide to Elliptic Curve Cryptography - Hankerson, Menezes, Vanstone

Presentation written in Markdown, powered by Deckset: http://decksetapp.com