HS18

DISTRIBUTED SYSTEMS ADVANCED & BLOCKCHAIN

Brief Introduction to Elliptic Curve Cryptography

T. Bocek

thomas.bocek@hsr.ch

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Distributed Systems - In the *News*

- **08.11.2018: The New Illustrated TLS Connection**
  - TLS handshake discussed in last lecture
- **Ethereum Whisper: no structured broadcast**
  - https://github.com/ethereum/go-ethereum/blob/b16cc501a88eb7c36cfd8ee9be1c0c985ef4a7d9/whisper/whisperv6/peer.go#L197
- **09.11.2018: IETF 103: Vom guten und bösen Verkehrsmanagement**
  - Last lecture: everything is encrypted
  - Already outdated information: spin bit, explanaition
  - Mozilla, Google, Facebook: not implement
  - Microsoft, Apple: will implement
  - Name will be HTTP/3
- **08.11.2018: Sprint Is Throttling Microsoft's Skype Service, Study Finds**
  - "In the case of a video call, which is what we were testing, the video quality would be much poorer -- poorer than what the network supports,"
- **09.11.2018: Fraudulent South Korean Exchange Pure Bit Nabs $2.8M in ICO Exit Scam**
  - South Korean ICO
  - Exit scam: 13,500 ETH ~2.8m USD
  - "the community admin quickly kicked out every participant in their Kakao chat thread, before sending a final message of "Thank You.""

DSA Challenge Task Milestone Results

- **Practical work: key element of this lecture**
  - Learning distributed systems “the hard way”
  - Lecture gives an overview, students should be able to figure out the details (practical work)
  - Its not worth to learn e.g., Solidity if you don’t need it, as it can look completely different in 2 years. Concept matters!
  - No questions asked in exam to write Solidity code (you may be able to read though)

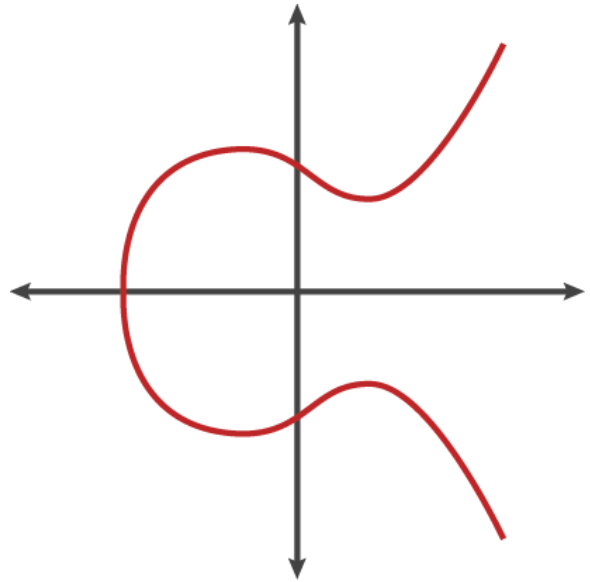
- **Milestone 1**

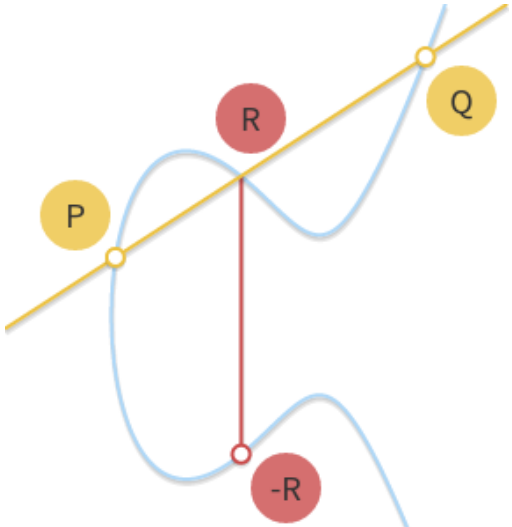
Joel Egger	Pekent Durmaz	Marcel Stocker	???
Fabian Meier	Gabriel Vlasek	Jan Winter	pass*
Ryan Horiguchi	Janik Schlatter		no CT
Yannick Zwicker	Pirmin Wenk	Matthias Baumann	pass
Remo Pfister	Keerthikan Thurairatnam		pass
Matthias Dunkel	Raffael Vögeli	Sandro Blatter	pass
Quoc Tin Tran	Marcel Keller		???
Fabio Rossetto	Dumeni Vincenz	Alesandro Bonomo	pass*
Ben Daniels	Micha Schena	Zarko Dragojevic	pass*
Andi Hörler	Jonas Kugler		no CT
Tobias Saladin	Pascal Hürlimann	Martin Odermatt	pass
Rolf Furrer	Stefano Kals	Lukas Laetsch	pass*
Demian Thoma	Lukas Röllin	Patrick Bächli	pass
Elias Brunner	Alexander van Schie	Pascal Bertschi	pass*
Dominik Kessler	Cyrill Hänni		pass*
Kyburz Cyril	Kramer Roman		pass
Wehli Cedric			no CT

  - * we had issues with installation, talk to us!

Elliptic Curve Cryptography

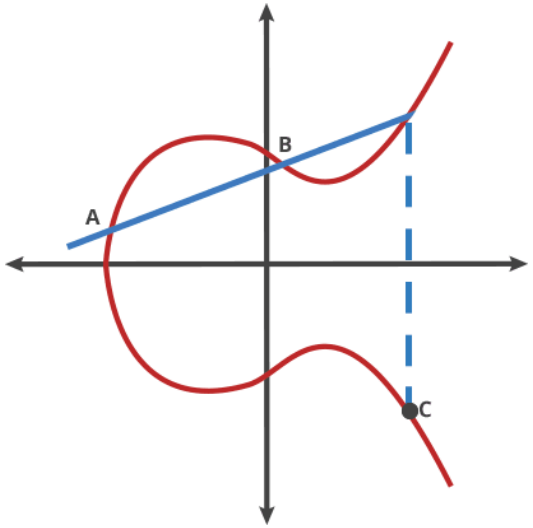
- **Elliptic Curve Cryptography from a practical point of view**
- **Set of points described by the equation:**
  - $y^2=x^3+ax+b$
  - No singular curves: $4a^3+27b^2\neq 0$
- **If a line intersects two points in the curve and not tangent it will always intersect a third**
- **Three aligned, non-zero points P, Q and R, their sum is: P+Q+R=0**
  - P+Q=-R
  - Interactive
- **P+P, tangent through P**
  - multiplication: nP = P+P+P+P+P

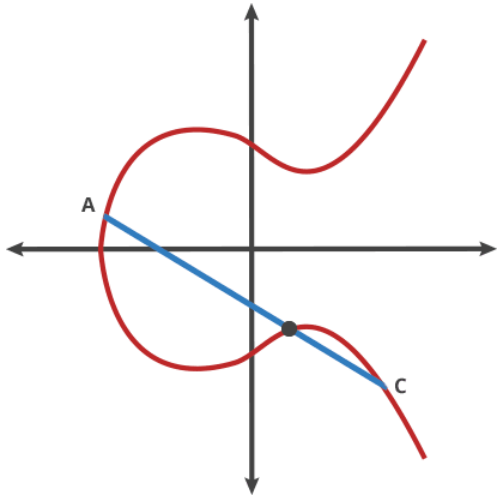


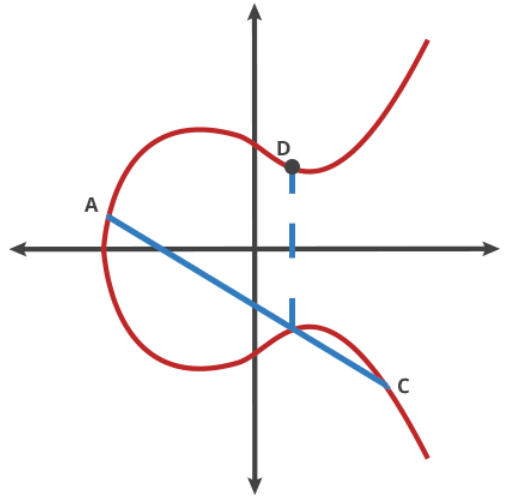


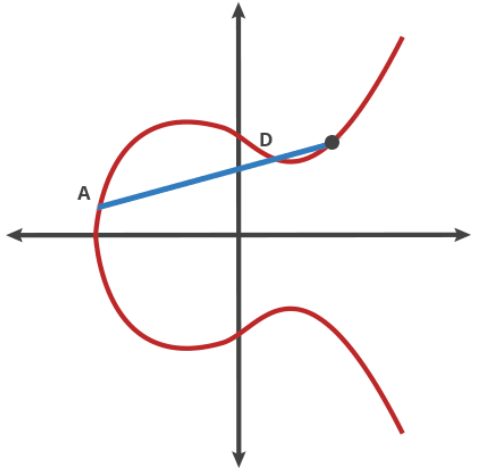
Elliptic Curve Cryptography: Multiplication

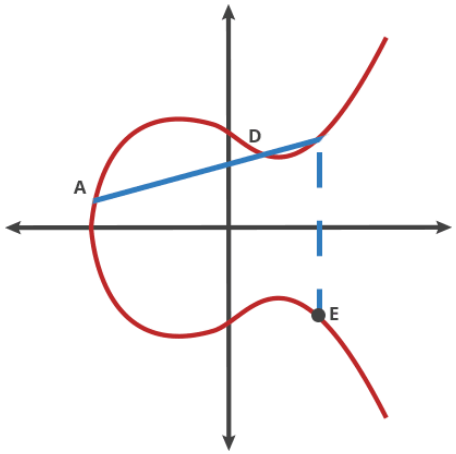
- **A + A = B**
- **A + B = C (P + Q = R)**
- **A + C = D**
- **A + D = E**
- **Point A added n times = final point**
  - Knowing A and the final point, finding n is very hard (finite fields)
  - Trapdoor function, following the dotted lines easy, reverting it very hard







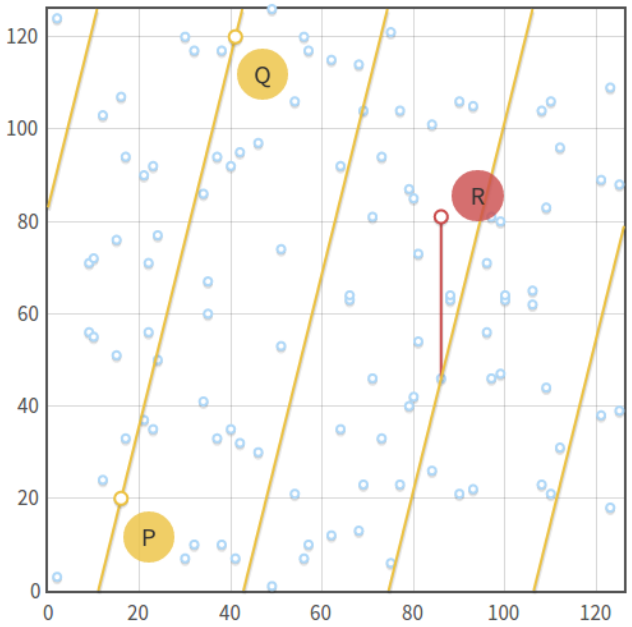




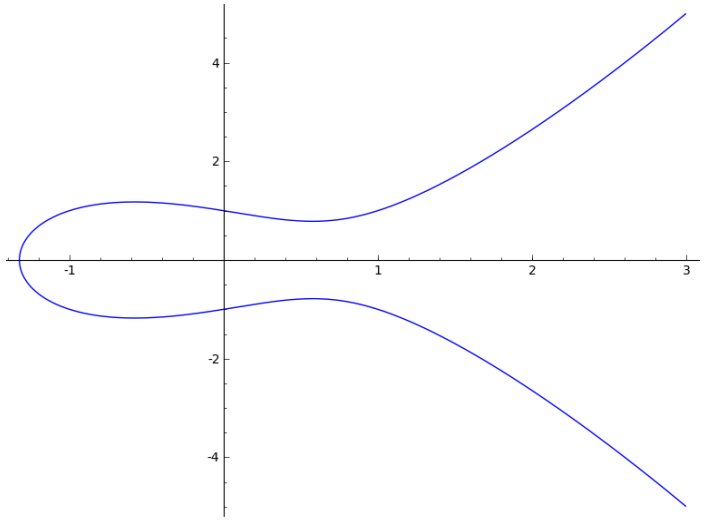
source

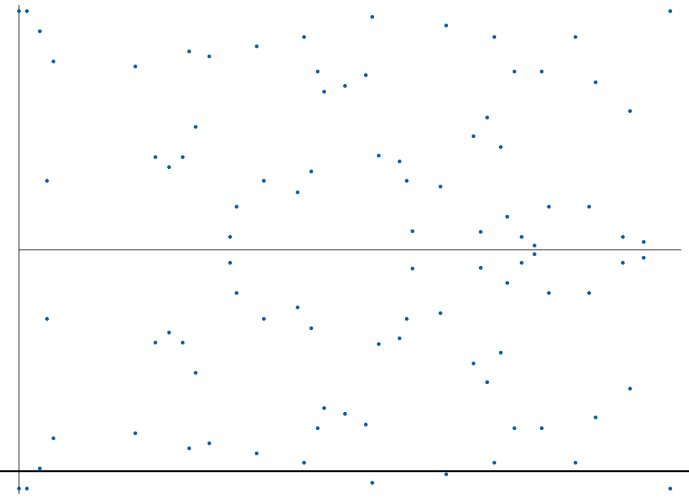
Elliptic Curve Cryptography

- **From real numbers**
  - $y^2 = x^3 - x + 1$
- **To finite fields (fixed range)**
  - $y^2 = x^3 - x + 1 \pmod{97}$
- **Same rules apply!**
- **ECC: nA = E (finding n is hard)**
- **Discrete logarithm problem**
- **$a=b^k \bmod n$**
  - $9 = 3^k \bmod 13$ (k=5, finding is hard)



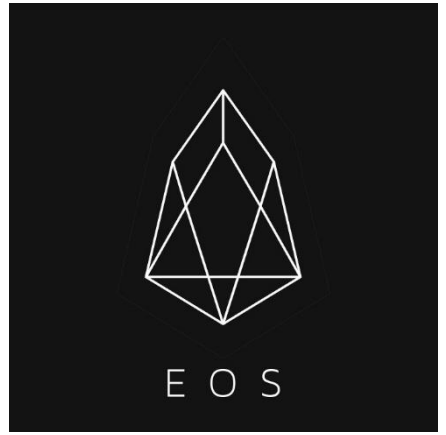
source





Elliptic Curve Cryptography: Cryptocurrencies

- **Bitcoin ECC**
  - $y^2=x^3+ax+b$
  - $a=0$
  - $b=7 \rightarrow \mathbf{y^2=x^3+7}$
  - Modulo p = $2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$
  - Base point G = 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8
  - Also used in: Bitcoin, Ethereum, EOS, Litecoin, Dash, Zcash

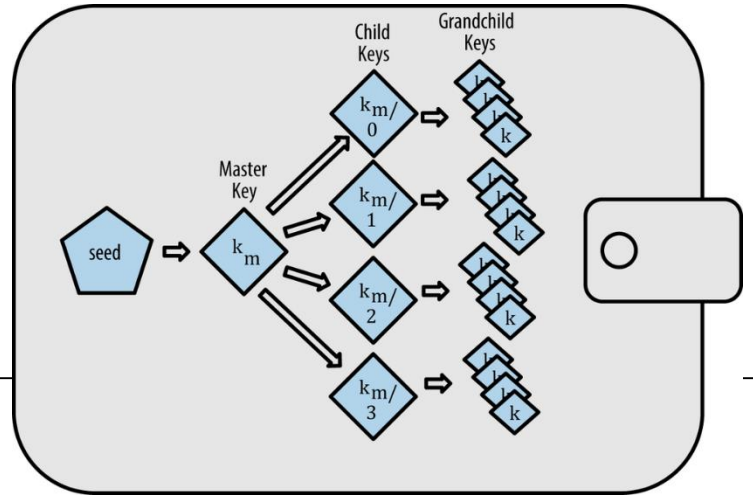


ECC Key Exchange + Signature

- **Choose base curve, set base point G**
- **Private key: random number d, public key H result of dG = H**
  - Recovering p knowing only the point H and G in dG = H is very hard
- **ECDH, Alice, Bob, same G, same curve**
  - Alice: $H_A=d_AG$
  - Bob: $H_B=d_BG$
  - Alice, Bob exchanges H_A ,H_B
  - Alice: $S = d_AH_B$, Bob: $S = d_BH_A$
  - Alice: $d_A(d_BG)$, Bob: $d_B(d_AG)$ = shared secret
- **Static vs. ephemeral ECDH**
  - Reuse or use key pair only once
- **Signatures**
  - Random number k
  - r = kG (x-coordinate)
  - $s = (hash+rd_A) / k$ (same, $k = (hash+rd_A) / s$)
- **Verification – hash, G, r, s, H_A**
  - $P=u_1G+u_2HA$
  - $P=(hash/s)G + (r/s)H_A \rightarrow kG$
- **Reusing random number (PS3 hack, Android, 55BTC lost/stolen)**
  - $s_1 = (hash_1+rd_A) / k$
  - $s_2 = (hash_2+rd_A) / k$
  - $k=(hash_1-hash_2)/(s_1-s_2)$
  - $d_A=(sk-z)/r$

HD Keys

- **Ledger / wallets**
  - Seed, 24-words, BIP39
- **Create word mnemonic seed**
  - Random number R, 256bits
  - The first 8 bits of the SHA-256 hash of R is appended to the end, giving us 264 bits.
  - All 264 bits are split into 24 groups of 11 bits.
  - Each group of 11 bits is interpreted as a number in the range 0 - 2047, which serves as an index to the BIP 39 wordlist, giving us 24 words
- **Hierarchical Deterministic (HD) keys**
  - BIP32
- **Alice: $H_A=d_AG$**
  - Deterministic: calculate same key over and over again
  - $xG+H_A=(x+d_A)G$
    - Non-hardened, include d_A to calculate x
- **Hierarchical Deterministic Wallets**
  - Hardened: hash(parent private key + index)
  - Non-hardened: hash(parent public key + index)



ECC Signature - Key Take-aways

- **Key Takeaways**
  - An elliptic curve is nothing more than: Point = Multiplier * Public Base Point
  - Given the point, it's impossible to get the multiplier
- **Signatures are not deterministic, always using new random number**
  - Proof-of-stake uses commitment
  - Commitment = public key
  - Commitment proof = sign message

$$\frac{\text{SHA-256}([Seed_{PrevBlocks}] \cdot Seed_{Local} \cdot BlockHeight \cdot TimeInSeconds)}{Coins} \leq Target. \quad (1)$$

- **Local seed cannot be ECC signature, otherwise we end up with PoW**
  - VRF: deterministic signatures, IETF standard
  - Or use RSA…
- **Roman Blum, worked on VRF/RSA, now ScP (self-contained proofs)**

URL: b.socrative.com/student/

Room: HSR

Questions?

Name