Smart Contracts and Blockchain Technology

Lecture 7. Digital signatures

Christian Ewerhart

University of Zurich

Fall 2022

Introduction and overview

Last lecture: Cryptographic underpinnings

- Hash functions
- Residue class fields
- Discrete logarithm problem

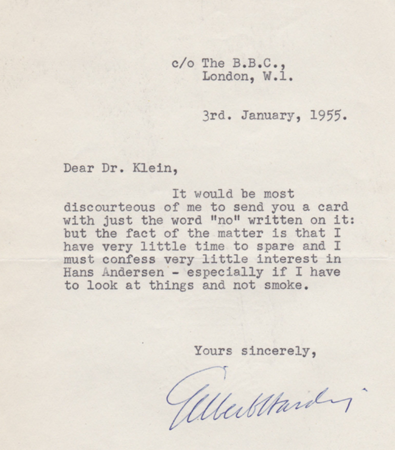
This lecture: Digital signatures

- Private and public keys
- Ethereum wallet addresses
- Elliptic curve cryptography

Digital signatures (1)

A problem

Signing **a letter** is easy...



c/o The B.B.C.,
London, W.1.

3rd. January, 1955.

Dear Dr. Klein,

It would be most discourteous of me to send you a card with just the word "no" written on it: but the fact of the matter is that I have very little time to spare and I must confess very little interest in Hans Andersen - especially if I have to look at things and not smoke.

Yours sincerely,

...but how do we sign an **electronic message**?

Digital signatures (2)

Public-key cryptography

Public-key cryptography, also known as asymmetric cryptography, is the basis of modern information security.

- Asymmetric means: There is **no shared secret**!

The protocol allows a sender to digitally sign a message. This allows the receiver of the message to verify that:

- the message comes from the sender (**authentication**), and
- the message has not been modified (**integrity**).

Non-repudiation is a closely related legal concept, which captures that there is evidence usable in court that proves that the sender has created a specific message.

Digital signatures (3)

Asymmetric cryptography

Asymmetric cryptography uses a combination of a **private key** and a **public key**. An example for a key pair in Ethereum is:

A 256-bit **private key** in hexadecimal representation:

k=0xf8f8a2f43c8376ccb0871305060d7b27b0554d2cc72bccf41b2705608452f315

A corresponding 520-bit **public key**, likewise in hexadecimal representation:

K=0x046e145ccef1033dea239875dd00dfb4fee6e3348b84985c92f103444683bae07b
83b5c38e5e2b0c8529d7fa3f64d46daa1ece2d9ac14cab9477d042c84c32ccd0

Public key -> prefix 0x04 (8 bits) and is followed by two elliptic coordinates
-> public key is uncondensed

Digital signatures (4)

Trapdoor functions

The public key is computed as the value of a **trapdoor function** when evaluated at the private key.

A trapdoor function is:

- **difficult to invert** (just as a hash function), but in contrast to a simple hash function, it is also
- **homomorphic**, meaning that arithmetic operations on the input translate into arithmetic operations on the output.

Digital signatures (5)

Examples for trapdoor functions

Exponentiation in finite fields

- Computing the residue class $r \equiv g^m \bmod p$ for some generator $g \in \mathbb{F}_p$, with p prime, is simple.[1]
- However, computing the **discrete logarithm** m from r (given the prime p and the generator g) is very difficult.

Scalar multiplication on elliptic curves

- Scalar multiplication on elliptic curves over a finite field $\mathbb{F}_p$ is comparably simple (as will be seen).
- However, inverting the operation is nearly impossible if p is sufficiently large.

[1] E.g., computing $g^{37} \equiv g^{32} \cdot g^4 \cdot g^1$, with $g^4 \equiv \left(g^2\right)^2$ and $g^{32} = \left(\left(g^4\right)^2\right)^2$ requires only six multiplications. This technique is known as **repeated squaring** or **square & multiply**.

Digital signatures (6)

Determination of the Ethereum address

Private key k (a number)
-> Trapdoor
Public key k·G = K -> K=(xk, xy)
-> generator of (E, +) elliptic curve
Ë=(xË, xy)

Start with the public key:

K=0x046e145ccef1033dea239875dd00dfb4fee6e3348b84985c92f103444683bae07b83b5c38e5e2b0c8529d7fa3f64d46daa1ece2d9ac14cab9477d042c84c32ccd0

The prefix `0x04` is dropped before the hash is computed:

K'=6e145ccef1033dea239875dd00dfb4fee6e3348b84985c92f103444683bae07b83b5c38e5e2b0c8529d7fa3f64d46daa1ece2d9ac14cab9477d042c84c32ccd0

Compute the Keccak256 of the byte code:

Keccak256(K') =
0x2a5bc342ed616b5ba5732269001d3f1ef827552ae1114027bd3ecf1f086ba0f9

To obtain the **Ethereum wallet address**, keep the last 20 bytes (and add the prefix `0x`, as usual):

`0x001d3f1ef827552ae1114027bd3ecf1f086ba0f9`

Digital signatures (7)

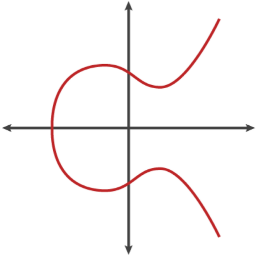
Elliptic curves

An **elliptic curve E** over the real numbers consists of:

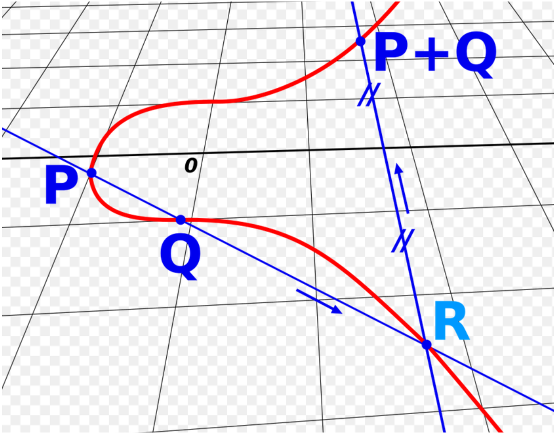
- the set of solutions $(x, y) \in \mathbb{R}^2$ of a cubic equation of the form

$$y^2 = x^3 + ax + b, \tag{1}$$

  where $a, b \in \mathbb{R}$ are constants such that $\Delta \equiv 4a^3 - 27b^2 \neq 0$, and
- a **point at infinity**, denoted by $\mathcal{O}$.



Surprising fact: Two points $P, Q \in \mathbf{E}$ can be geometrically **added** to obtain a new point $P + Q \in \mathbf{E}$.



Digital signatures (9)

Addition on elliptic curves (cont.)

This also works in "nongeneric cases":

$$P + \mathcal{O} = P, \tag{2}$$
$$P + (-P) = \mathcal{O}, \tag{3}$$

where $-P = (x, -y)$ is the **inverse** of P.

Lemma:

- $P + Q = Q + P$ (**commutativity**)
- $(P + Q) + S = Q + (P + S)$ (**associativity**) $\rightarrow$ brackets are not needed...

Digital signatures (10)

Algebraic counterpart of addition

The **slope** of the line connecting two points P and Q (in generic position) is given as

$$\sigma = \frac{y_P - y_Q}{x_P - x_Q}. \tag{4}$$

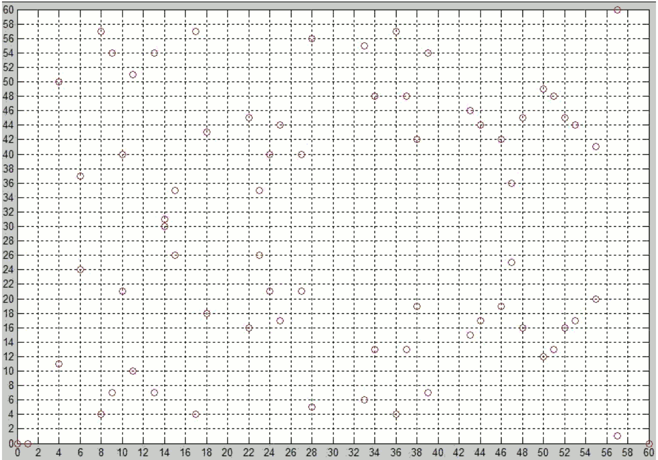
Then the **sum** $P + Q$ has the coordinates

$$x_{P+Q} = \sigma^2 - x_P - x_Q, \tag{5}$$

$$y_{P+Q} = -y_P + \sigma(x_P - x_{P+Q}). \tag{6}$$

Another surprising fact: These geometrically motivated formulas work in the same way for elliptic curves over finite fields (i.e., where $(x, y) \in \mathbb{F}_p$ for some prime p).

Shown is the set of solutions to $y^2 = x^3 - x$ over the finite field $\mathbb{F}_{61}$.



Digital signatures (12)

Scalar multiplication on elliptic curves

Scalar multiplication by n corresponds to an iterated addition:

$$n \cdot P = \underbrace{P + \ldots + P}_{n \text{ times}} \tag{7}$$

This is the **trapdoor function:**

- Scalar multiplication is easy.[2]
- However, determining n from $Q = n \cdot P$ and P can be very hard.

This is called the **discrete logarithm problem for elliptic curves**.

[2] Use a variant of the square & multiply technique: "double & add"

Digital signatures (13)

The elliptic curve used by Ethereum and Bitcoin

Ethereum and Bitcoin use the same elliptic curve, called **secp256k1**.

That elliptic curve is defined over the finite field $\mathbb{F}_p$ through

$$y^2 \equiv x^3 + 7 \mod p, \tag{8}$$

where $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$, which is a very large prime.[3]

The generator used is (prefix `04` followed by x and y coordinates):

G=0479BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798
483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8

[3] In fact, p=115 792 089 237 316 195 423 570 985 008 687 907 853 269 984 665 640 564 039 457 584 007 908 834 671 663.

Digital signatures (14)

Signing a message

Suppose you plan to send an **electronic message** m, and you wish to add a digital signature so that the receiver can verify that the received message is authentic and unchanged.

You create a cryptographically secure random number q, the **ephemeral private key.**

Then, you compute the **ephemeral public key** $Q = q \cdot G$ on the elliptic curve $\mathbf{E} = \mathbf{secp256k1}$.

Then, the **digital signature** $(r, s) \in \mathbb{F}_p \times \mathbb{F}_p$ is given as

$$r = x_Q, \tag{9}$$

$$s \equiv q^{-1}(\text{Keccak256}(m) + rk) \bmod p. \tag{10}$$

Suppose you receive a message m, augmented by a digital signature (s, r).

Calculate in $\mathbb{F}_p$:

$$\begin{aligned} w &\equiv s^{-1} \bmod p, && (11) \\ u_1 &\equiv \text{Keccak256}(m) w \bmod p, && (12) \\ u_2 &\equiv rw \bmod p, && (13) \end{aligned}$$

Next, calculate on the elliptic curve $\mathbf{E} = \mathbf{secp256k1}$.the point

$$\widehat{Q} = u_1 \cdot G + u_2 \cdot K. \quad (14)$$

If $x_{\widehat{Q}} = r$, then **the signature is valid**!

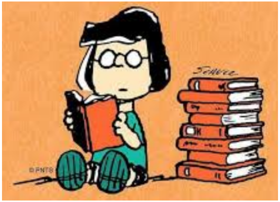
Digital signatures (16)

Crucial point

Each signature requires a new ephemeral private key.

This chapter is based on Antonopoulos and Wood (2018, Chapters 4 and 6).

The basic idea underlying asymmetric cryptography is due to Diffie and Hellman (1976) and Merkle (1978).



Digital signatures (18)

Antonopoulos, Andreas M., and Gavin Wood. Mastering ethereum: building smart contracts and dapps. O'Reilly Media, 2018.

Diffie, W., & Hellman, M. E. (1976). New directions in cryptography (1976) https://doi. org/10.1109.

R. C. Merkle, "Secure communication over an insecure channel." Common. Ass. Comput. Moch., vol. 21. pp. 294-299, Apr. 1978.