Cryptography and Elliptic Curves Lecture 23 11.22.23

From book by Hoffstein, Pipher, Silverman (UTM)
(originally I had learned from a book by Silverman-Tate)

<u>Goal</u>: Brief idea of cryptography + public key cryptosystems
Elliptic curves

~

I. Public Key Cryptography / Cryptosystems (PKC)

<u>Scenario</u>:

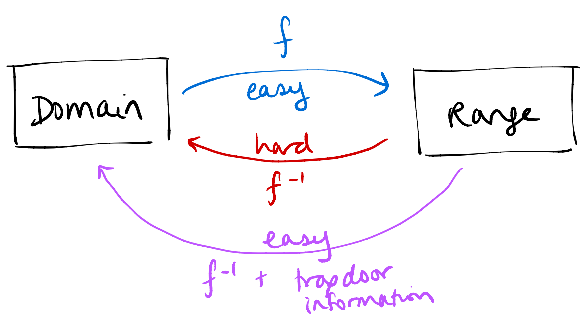
Alice (A) wants to send a secret message (M) to Bob (B).
But there is an "adversary" Eve.L (E), who wants to intercept M.

eg. A = you, B = your bank, M = secret financial info, E = interceptor

These names have been used by the CS literature for ages so I didn't change them.

Main idea behind PKC:

Find a function f : Domain(f) $\longrightarrow$ Codomain(f) $\supset$ Range(f) = Im(f)
such that $f(M)$ is easy to compute but f^{-1} is way harder:



"Hard": Impractical for the purposes of Eve.L
~ no poly time algorithm, eg.

Classic example (for an idea of PKC)

Let $\mathbb{P}$ be the set of all prime natural numbers.

Then $f: \mathbb{P} \times \mathbb{P} \longrightarrow \mathbb{P}^2$ is easy to compute.

$(p, q) \longmapsto pq$

↑ products of two primes

"semiprimes"

In grade school you learned an algo for multiplying numbers that basically $O(n^2)$

In grade school you learned an algorithm for multiplying numbers that was basically $O(n^2)$ (for multiplying two n-bit integers). Computers can do this in a little less (asymptotically). eg. $O(n \log n \log\log(n))$.

In any case, an adversary can handle multiplying two d digit numbers together effectively immediately.

On the other hand, if I gave you large number $N \in \mathbb{N}$ and told you it is a product of two primes p, q. it is exceedingly more difficult to figure out the pair

eg. $p =$ [redacted] $q =$ [redacted] $N = pq = 40186903$

[Redacted]

(What are p, q?)

Factoring is MUCH harder than multiplying.

$f^{-1}: \mathbb{P}^2 \longrightarrow \mathbb{P} \times \mathbb{P}$ is well-defined!

but hard to compute on each input

But, if I told you p, you could easily find q.

This is the trapdoor info. Just use division algorithm.

Back to Alice and Bob

$p = 180181$ $\quad q = 115249$

Alice wants to send Bob a secret message M later.

① Alice + Bob agree on a private key $p = 180181$ ahead of time. (Eve does not have this info)

for actual PKC algs, we don't need this step; see later.

② When Alice is ready to send M, they pick another prime $q = 115249$ as the password / cipher key for the document. (We don't have time to talk about this.)

So M' is a scrambled message that requires q to decipher (effectively).

③ Alice then computes $n = pq$ and posts the info (M', n) on a public Discord.

Both Bob and Eve can see this.

④ Bob computes $q = n/p$ and uses q to decipher $M' \rightsquigarrow M$.

It takes Eve a week to compute q; it's too late.

Actual PKC might go like this:

Diffie-Hellman key exchange relies on discrete logarithm problem being hard: $g^x \equiv h \pmod p$

ie $x = \log_g h$ in $\mathbb{F}_p^\times$

More precisely, $\log_g : \mathbb{F}_p^\times \longrightarrow \mathbb{Z}/(p-1)\mathbb{Z}$ turns out to be well-defined.

Diffie Hellman key exchange

① Public parameter creation $p=941$, $g=627$

Alice + Bob's organization agrees on a large prime p and an integer g with large prime order in $\mathbb{F}_p^\times$

(i.e. $\{g^k \mid k \in \mathbb{N}\}$ is a set with a large prime # of elements.

② Private computations

Alice:	Bob
chooses a secret integer a.	chooses secret int. b
Computes $A \equiv g^a \pmod p$	Computes $B \equiv g^b \pmod p$.
$a = 347$	$b = 781$
$A = 390 \equiv 627^{347} \pmod{941}$	$B = 691 \equiv 627^{781} \pmod{941}$

secret!

③ Public exchange of values

Alice + Bob send A, B to each other.

(Eve may be able to intercept).

④ Further Private Communication

Shared secret value is $470 \equiv 627^{347 \cdot 781} \equiv A^b \equiv B^a$

Alice can compute $B^a \equiv (g^b)^a \equiv (g^a)^b \equiv A^b \pmod p$ Bob can compute

Eve needs to solve the discrete log problem:

$$627^a \equiv 390 \pmod{941} \quad \text{or} \quad 627^b \equiv 691 \pmod{941}$$

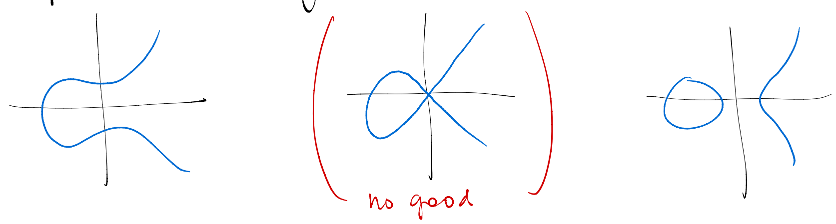
* As far as we know, there is not an efficient way for Eve to find the secret value $A^b \equiv B^a$.

II. Elliptic Curves

defn. An elliptic curve is the set of solutions to an equation $Y^2 = X^3 + AX + B$.

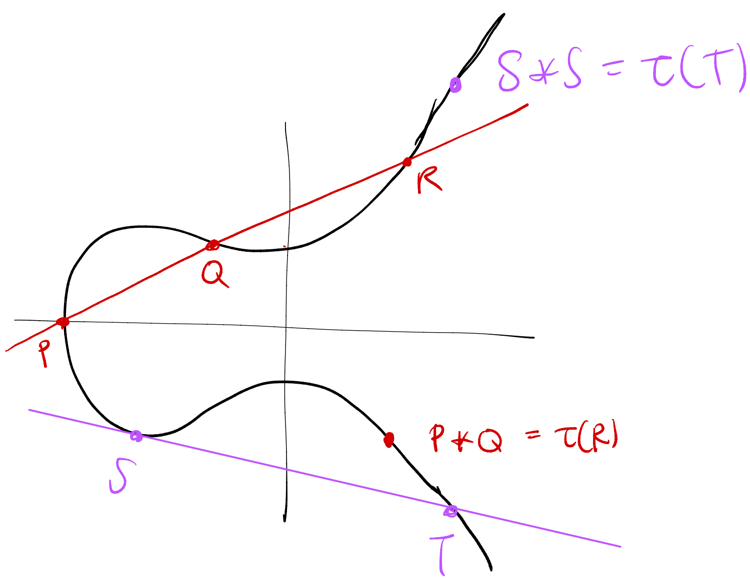
$$E(\mathbb{R}) = \{(x,y) \in \mathbb{R}\times\mathbb{R} \mid y^2 = x^3 + Ax + B\} \cup \{\infty\}$$

In the real plane, they look like this:



Claim The points on this curve form an abelian group!

Group operation: given P, Q, take the line $\overline{PQ}$, find the third intersection point R, and take $\tau(R)$.



This is the geometric description. You must of course use actual equations, intersections, tangent lines, etc to compute the points.

(Algebraic geometry)

Inverse & Identity? ① $P * \mathcal{O} = \mathcal{O} * P = P$ for $\mathcal{O} = \infty$.

② therefore $\tau(P) =$ "P^{-1}"

Elliptic Curves over finite fields:

Now everything is mod p.

$$E(\mathbb{F}_p) = \{ (x,y) \in \mathbb{F}_p \times \mathbb{F}_p \mid y^2 = x^3 + Ax + B \} \cup \{\mathcal{O}\}$$

note $p=2$ is a weird prime, so we usually require $p \geq 3$.

thm (Hasse) Let E be an elliptic curve over $\mathbb{F}_p$.
Then $\#E(\mathbb{F}_p) = p+1-t_p$ where $|t_p| \leq 2\sqrt{p}$.

The Elliptic Curve discrete log problem

Recall: For Diffie-Hellman: $h \equiv g^x \pmod p$

"Discrete log problem"

Eve has to figure out $h \equiv \underbrace{g \cdot g \cdot g \cdots g}_{x = \text{how many times?}} \pmod p$

Now Alice (or the organization) chooses an elliptic curve.
Alice chooses and publishes two points P, Q in $E(\mathbb{F}_p)$
such that $Q = \underbrace{P * P * \cdots * P}_{n} = P^{*n}$ in $E(\mathbb{F}_p)$

and keeps the secret n to themselves.

We can do more to define "$n = \log_P(Q)$" (if it exists, and if we agree on using the "smallest" one.)

In reality: Your browser might choose $E(\mathbb{F}_p)$ and point P.
Alice chooses n (secret) and publishes $Q = P^{*n}$.