Elliptic Curve Cryptography

Matej Penciak

Northeastern University

March 18, 2022
Slides available at:
https://mpenciak.github.io/assets/pdfs/reu-2022-slides.pdf

Outline

1. Basics of cryptography
   - Encryption and decryption
   - Uses of encryption
   - The RSA algorithm
2. Basics of elliptic curves
   - What?
   - Why?
   - How?
3. Putting it all together

The setup

Alice wants to send Bob a message, but wants to keep it a secret from Carol.

The setup

Alice wants to send Bob a message, but wants to keep it a secret from Carol.

Some failed attempts

Alice first tries sending Bob a letter. Opening someone elses' correspondence is against the law after all!

But Carol is a hardened felon, and has no moral qualms with opening other people's letters

Some failed attempts

Alice first tries sending Bob a letter. Opening someone elses' correspondence is against the law after all!

But Carol is a hardened felon, and has no moral qualms with opening other people's letters

So instead Alice sends a lockbox with a key that only Bob has!

But Carol has a hammer and breaks open the lockbox with ease!

Some failed attempts

Alice first tries sending Bob a letter. Opening someone elses' correspondence is against the law after all!

But Carol is a hardened felon, and has no moral qualms with opening other people's letters

So instead Alice sends a lockbox with a key that only Bob has!

But Carol has a hammer and breaks open the lockbox with ease!

Some more failed attempts

Alice is undeterred and tries a Caesar cipher:

```
a->c, b->d, c->e
...
x->z, y->a, z->b

cheese -> ejggug
```

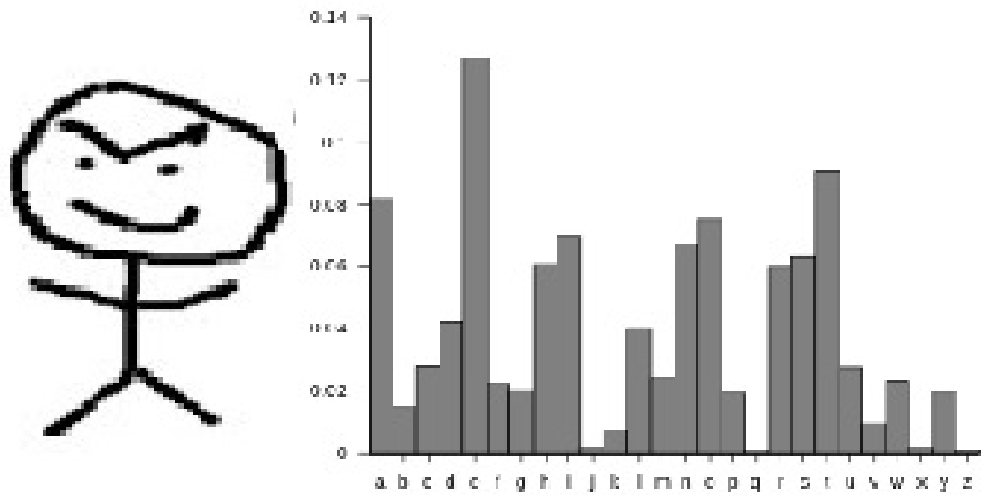
Some more failed attempts

Alice is undeterred and tries a Caesar cipher:

```
a->c, b->d, c->e
...
x->z, y->a, z->b

cheese -> ejggug
```

But Carol has access to basic letter frequency statistics, and can very easily un-do the cipher by determining the shift used!



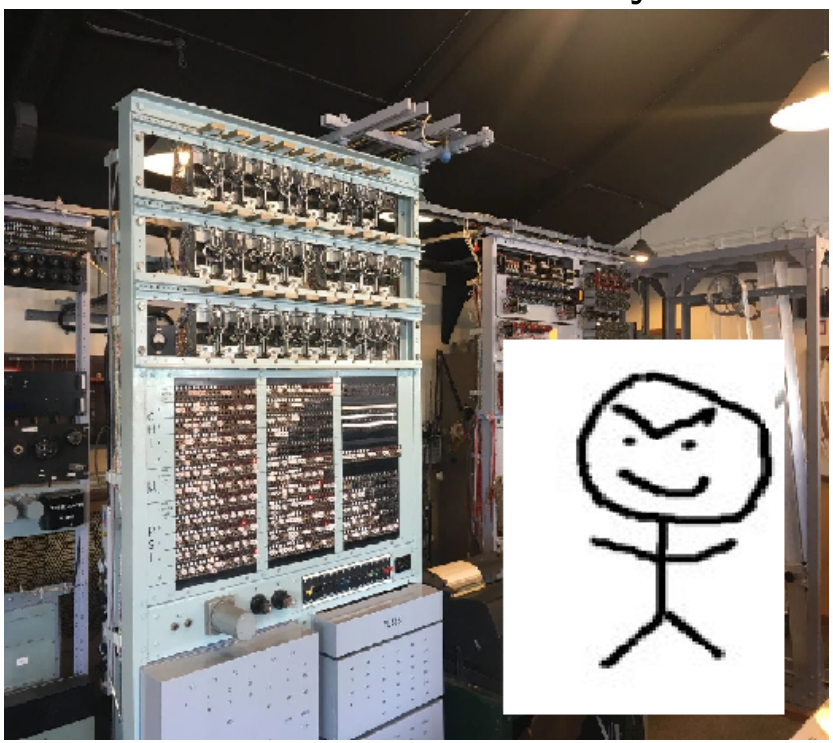
Some more failed attempts

Alice and Bob broke into the National Museum of Computing at Bletchly park, and have access to a couple replicas of German Enigma machines from WWII U-boats.

Some more failed attempts

Alice and Bob broke into the National Museum of Computing at Bletchly park, and have access to a couple replicas of German Enigma machines from WWII U-boats.

But Carol also broke into the museum, and stole the original Mark 2 Colossus, and can easily break the ciphers sent by Alice. Shoot!



Some more clever forms of encryption

The way forward now is clear. Alice needs to use math!

More specifically, we want to fine find some sort of "algorithm" that:

Allows Alice and Bob to enter a mutual understanding of the *encryption and decryption* scheme, and even with hearing the whole exchange, Carol would be unable to decrypt the message without *extreme computation effort*.

And hopefully the encryption and decryption process is *computationally easy*.

Buzzwords, and use-cases

- One-way functions (Hashing)
- Public-key cryptography (Diffie-Hellman key exchange)
- Discrete logarithm problem (RSA algorithm)

Sending messages back and forth is not the only use-case for encryption.

Cryptography is about the basic idea of trying to keep *something* secret.

- Alice wants to create a stream of pseudo-random numbers that are cryptographically secure

Buzzwords, and use-cases

- One-way functions (Hashing)
- Public-key cryptography (Diffie-Hellman key exchange)
- Discrete logarithm problem (RSA algorithm)

Sending messages back and forth is not the only use-case for encryption.

Cryptography is about the basic idea of trying to keep *something* secret.

- Alice wants to send Bob a message, and provide a signature ensuring she sent it.

Buzzwords, and use-cases

- One-way functions (Hashing)
- Public-key cryptography (Diffie-Hellman key exchange)
- Discrete logarithm problem (RSA algorithm)

Sending messages back and forth is not the only use-case for encryption.

Cryptography is about the basic idea of trying to keep *something* secret.

- Alice wants to access some information that Bob has stored

Buzzwords, and use-cases

- One-way functions (Hashing)
- Public-key cryptography (Diffie-Hellman key exchange)
- Discrete logarithm problem (RSA algorithm)

Sending messages back and forth is not the only use-case for encryption.

Cryptography is about the basic idea of trying to keep *something* secret.

- Bob sends Alice a choice between 3 options to commit to, and Alice wants to have a way to signal to Bob that she has made her commitment but does not want to reveal which selection was made.

Fermat's little theorem

The RSA algorithm is based on some relatively simple number theory: Fermat's *little* theorem (or an easy corollary of it)

Theorem (Fermat's Little Theorem)

Let p be a prime number, and a a natural number not divisible by p. Then

$$a^{p-1} \equiv 1 \mod p.$$

More generally, if we have p and q distinct primes, then for any a relatively prime to pq,

$$a^{(p-1)(q-1)} \equiv 1 \mod pq$$

The RSA Algorithm

Alice and Bob first have an agreed-up way of transforming plain text into a natural number, which in this case we refer to as c. For example

`'cheese' <-> [3, 8, 4, 4, 19, 4] <-> 030804041904 = c`

Alice and Bob both choose two (very large) prime numbers p_a, p_b and q_a, q_b. And calculate $n_a = p_a q_a$ and $n_b = p_b q_b$.

At the same time Bob chooses two natural numbers k, s so that $k \times s \equiv 1 \mod (p_b - 1)(q_b - 1)$.

The RSA Algorithm contd...

If Alice wants to send Bob the transformed plain text message 30804041904, she asks Bob for his *Public Key*, which in this case consists of the natural numbers k and n_b.

Alice then calculates

$$e \equiv c^k \mod n_b$$

and transmits the encrypted answer e to Bob.

Bob can now decrypt e by calculating $e^s \mod n_b$.

The RSA Algorithm contd...

Why does this work? Because $ks \equiv 1 \mod (p_b - 1)(q_b - 1)$ we know that $ks = 1 + t(p_b - 1)(q_b - 1)$ for some natural number t. So:

$$\begin{aligned} e^s \equiv (c^k)^s \equiv c^{ks} &= c^{1+t(p_b-1)(q_b-1)} \\ &= c \cdot c^{t(p_b-1)(q_b-1)} \\ &\equiv c \cdot 1^t \mod n_b \\ &= c \mod n_b \end{aligned}$$

Why this works

This public and private key system was originally introduced by Diffie and Hellman, and so the RSA (Rivest, Shamir, Adleman) cryptosystem described above is sometimes referred to as the RSA algorithm with Diffie Hellman key exchange.

All that Carol knows are Bob's public keys k and n_b, and the encrypted message e. In order to obtain c Carol would need to know s, or derive it by knowing the factorization $n_b = p_b \cdot q_b$.

Why this works

This public and private key system was originally introduced by Diffie and Hellman, and so the RSA (Rivest, Shamir, Adleman) cryptosystem described above is sometimes referred to as the RSA algorithm with Diffie Hellman key exchange.

All that Carol knows are Bob's public keys k and n_b, and the encrypted message e. In order to obtain c Carol would need to know s, or derive it by knowing the factorization $n_b = p_b \cdot q_b$.

But oh no! Carol has access to D-Wave's 5000 qubit quantum computer with an implementation of Shor's algorithm for the discrete logarithm, so Alice and Bob still cannot expect complete secrecy in their communications!

Oh no! Carol's got a quantum computer!

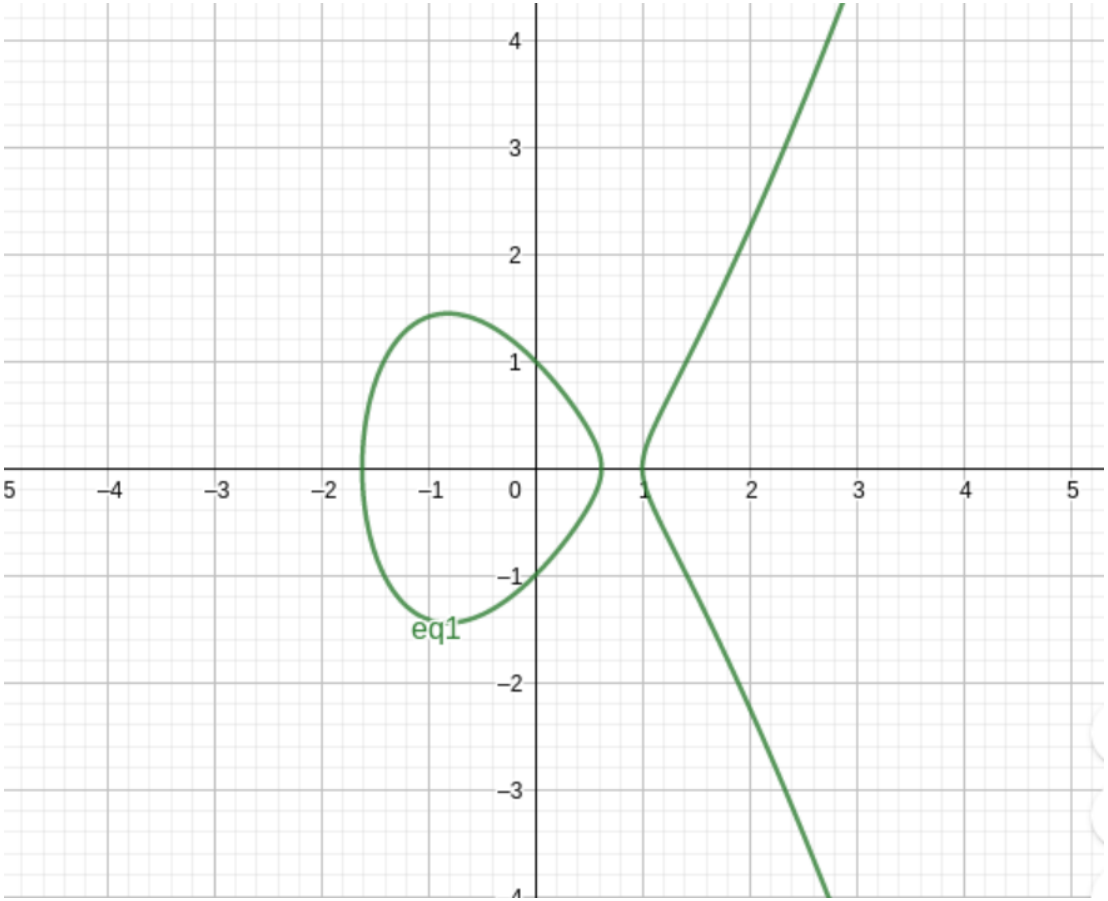
Elliptic Curves

Carol has one weakness: She didn't pay attention in algebraic geometry in grad school.

An elliptic curve is a cubic curve in the plane consisting of points (x, y) satisfying an equation of the form $y^2 = x^3 + ax + b$ for some constants a and b.

An elliptic curve

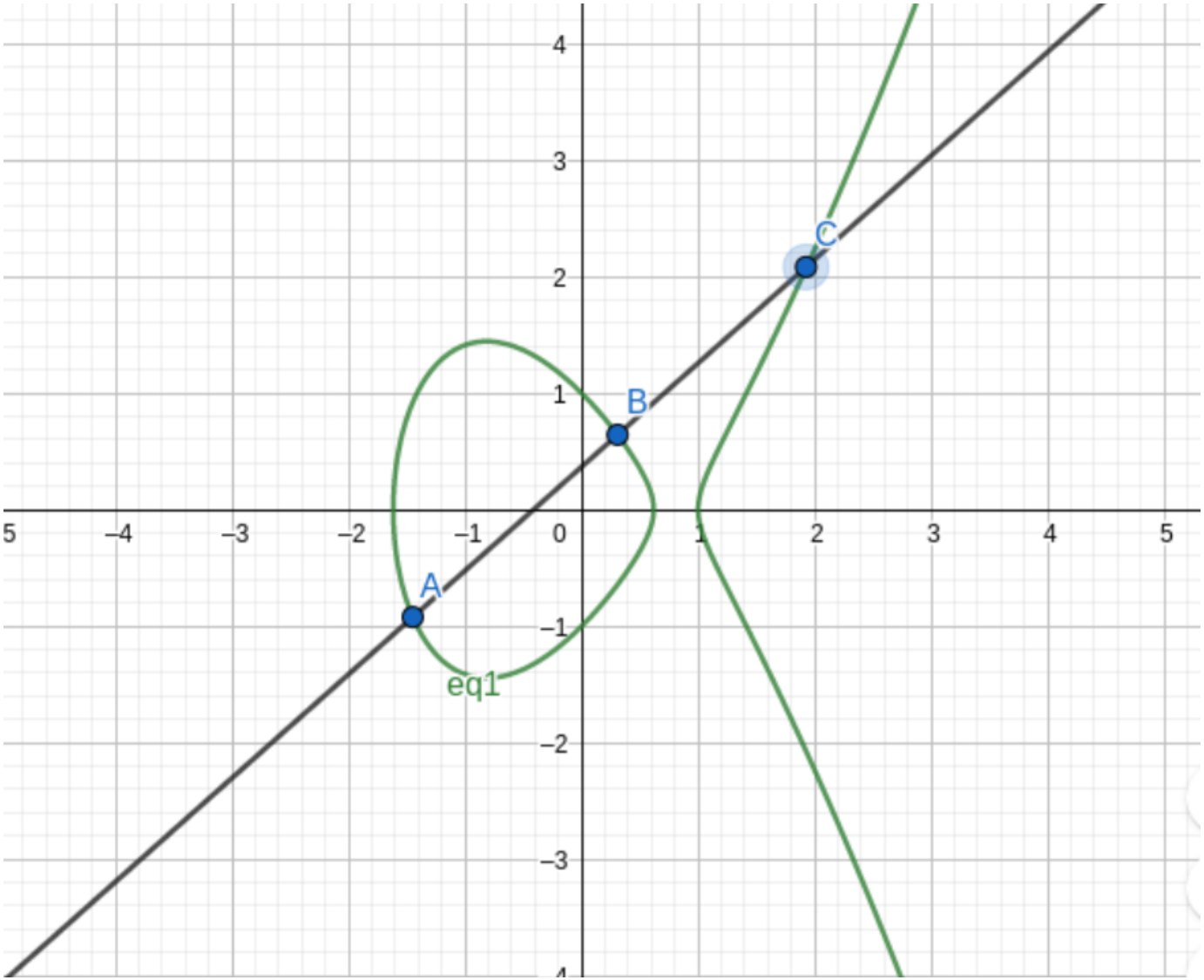
They look like



$$y^2 = x^3 - 2x + 1$$

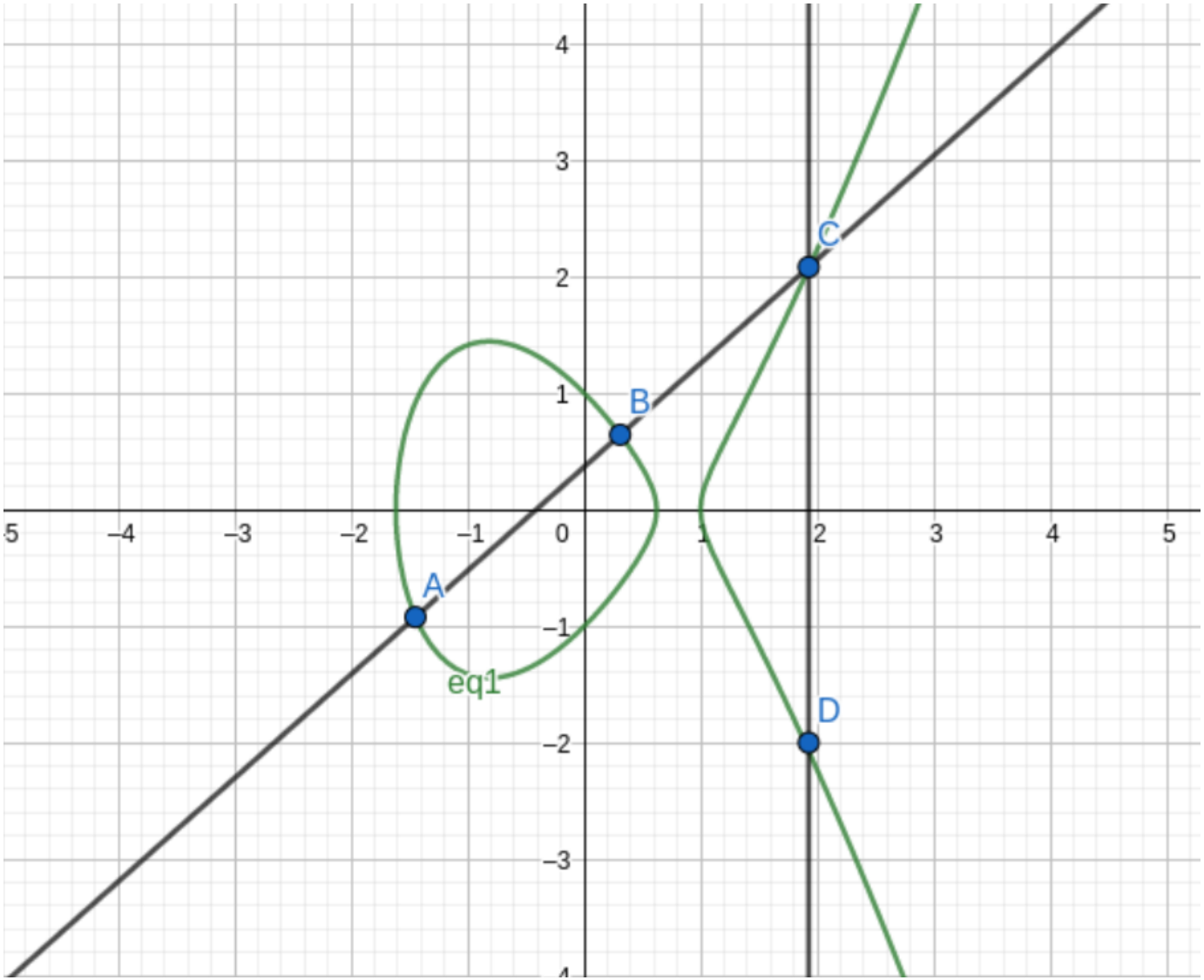
What's so special about elliptic curves?

The points of an elliptic curve form an additive group:



What's so special about elliptic curves?

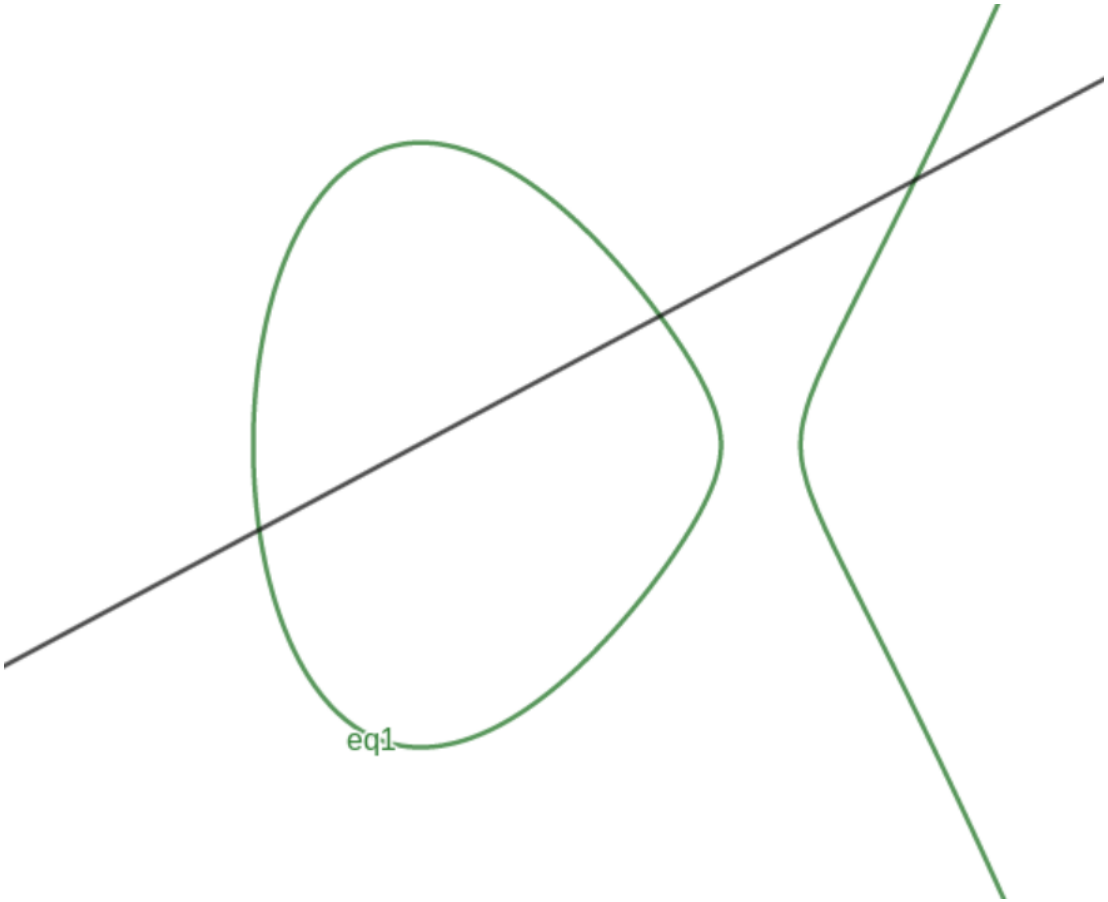
The points of an elliptic curve form an additive group:



What about the group laws?

I'm sweeping a lot under the rug!

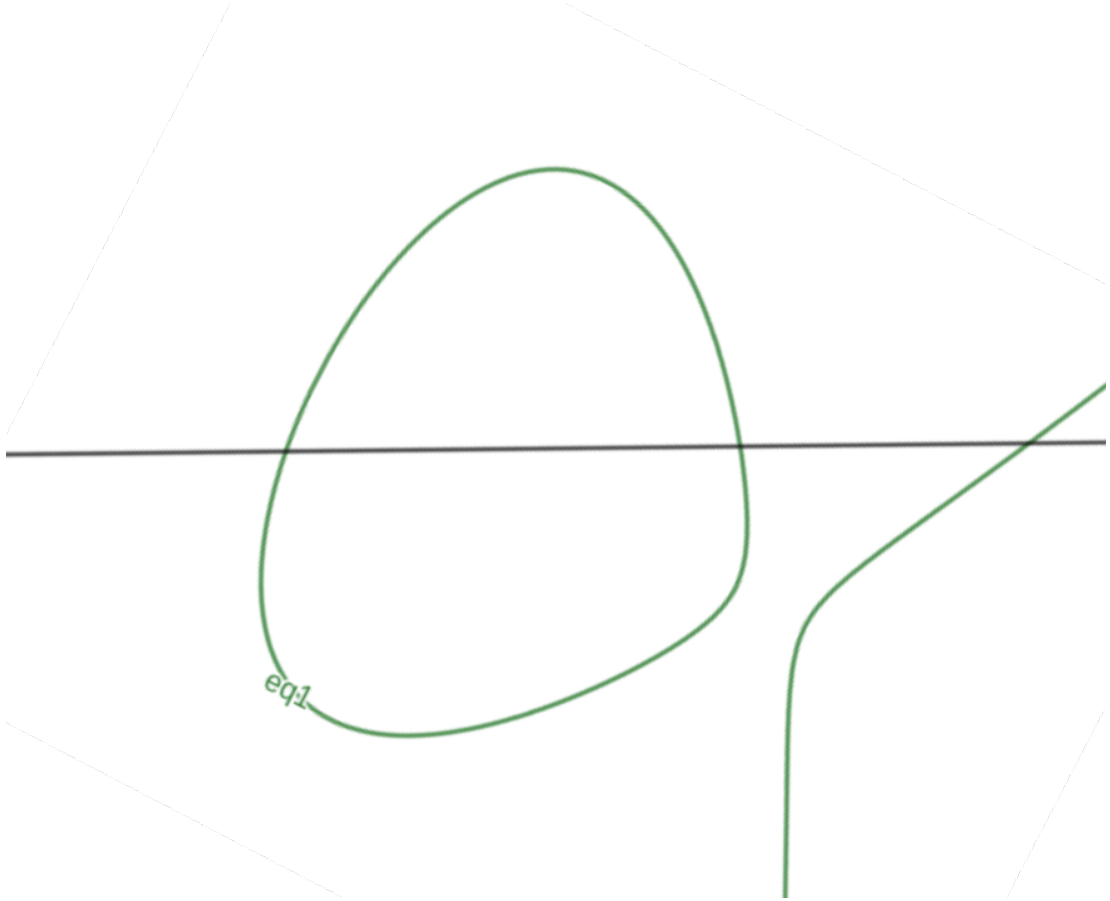
- How do I even know that a straight line will intersect in 3 points?



What about the group laws?

I'm sweeping a lot under the rug!

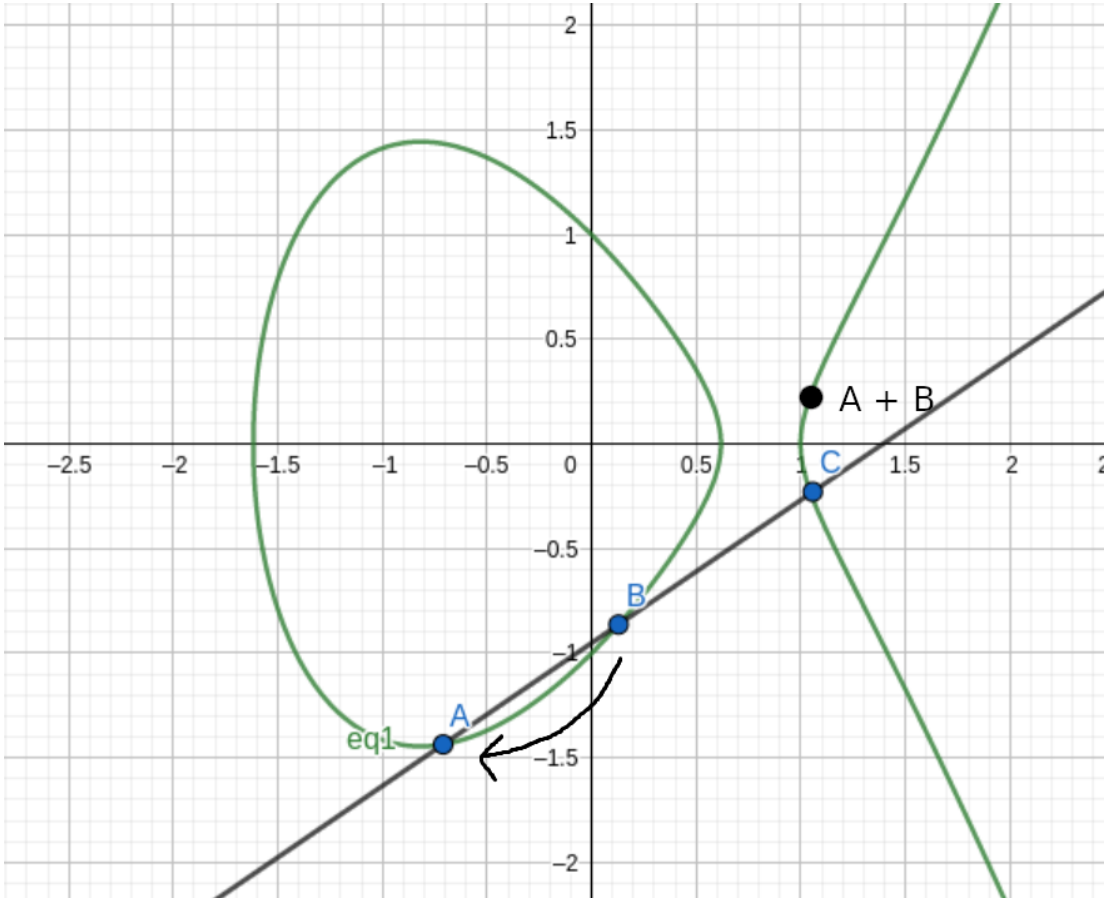
- How do I even know that a straight line will intersect in 3 points?



What about the group laws?

I'm sweeping a lot under the rug!

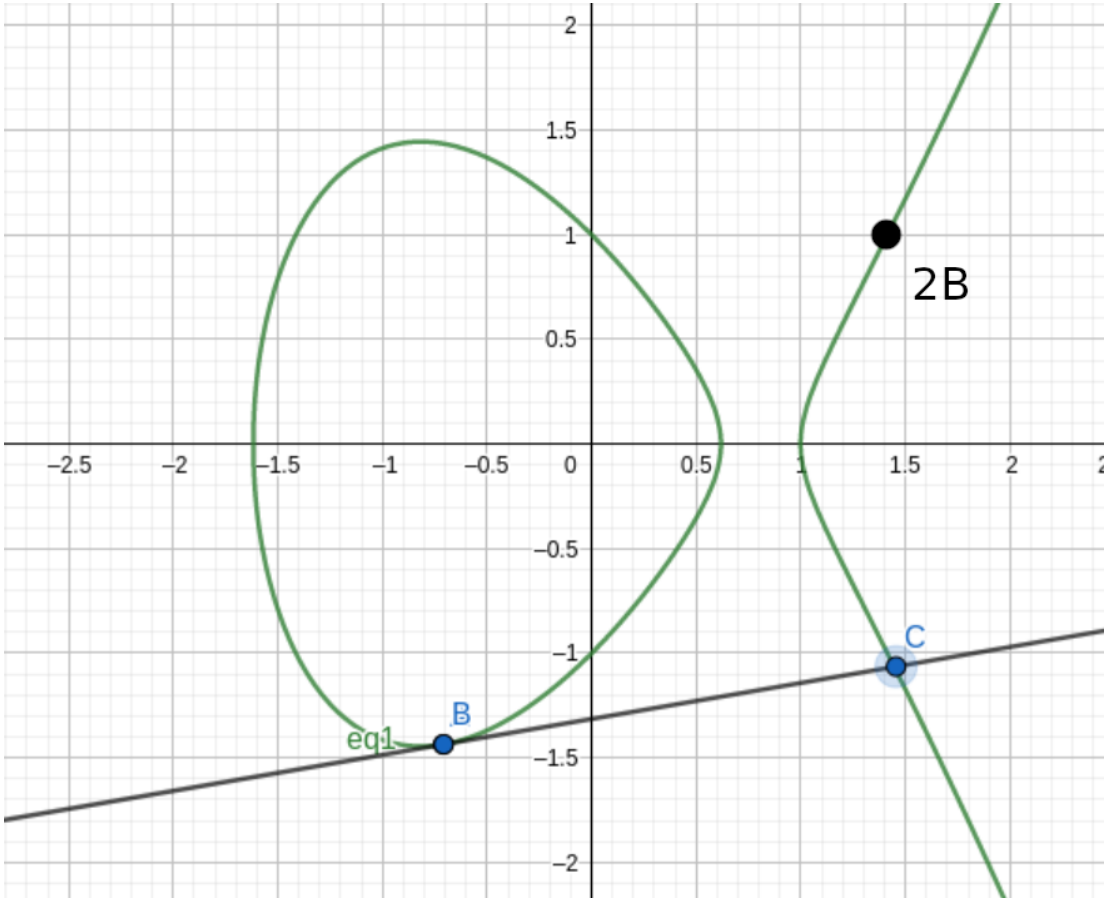
- How do I even know that a straight line will intersect in 3 points?
- How do I add add a point to itself?



What about the group laws?

I'm sweeping a lot under the rug!

- How do I even know that a straight line will intersect in 3 points?
- How do I add add a point to itself?



What about the group laws?

I'm sweeping a lot under the rug!

- How do I even know that a straight line will intersect in 3 points?
- How do I add add a point to itself?
- The identity is actually some "point at infinity" at the end of every straight line.

Proof of associativity

$$\mathcal{O},\ P,\ Q,\ R,\ P*Q,\ P+Q,\ Q*R,\ Q+R \qquad (\dagger)$$

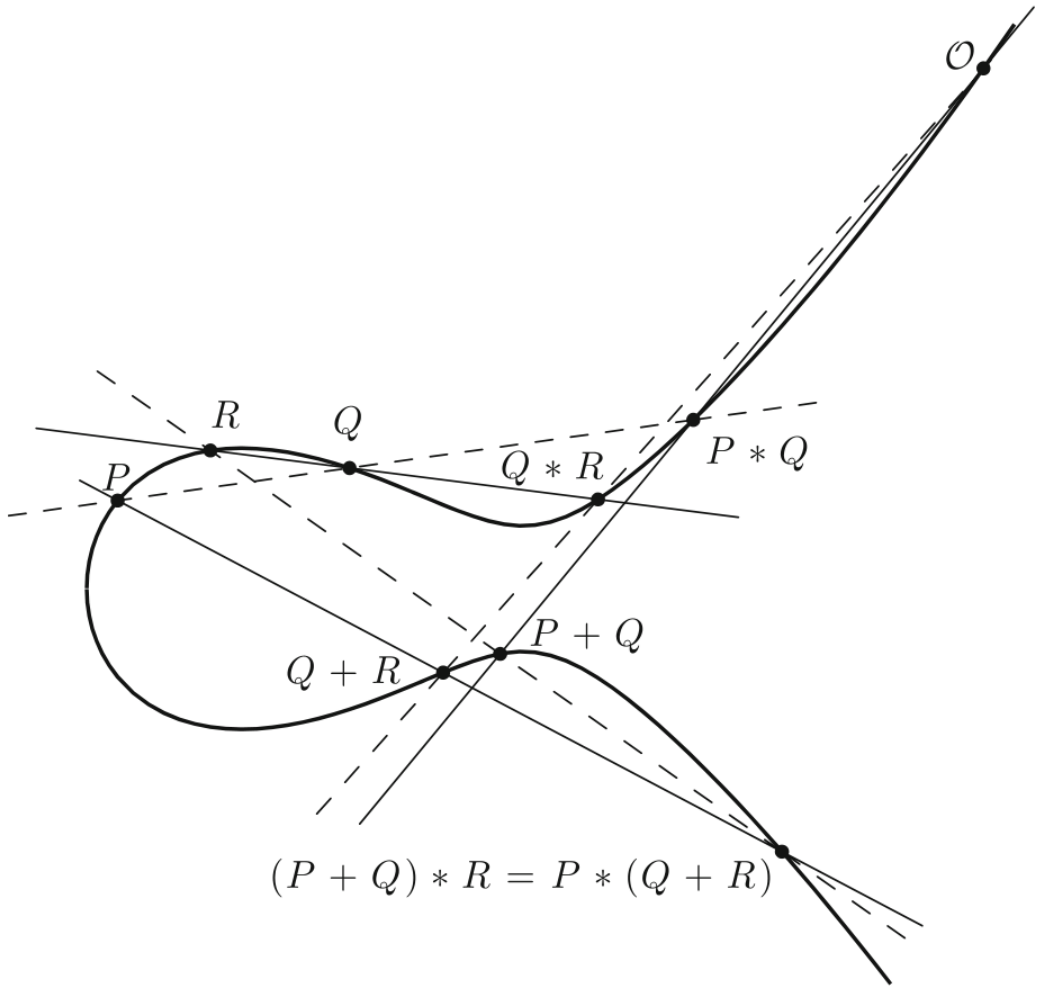


Figure 1.9: Verifying the associative law

Why does this help us?

Where do the points (x, y) live?. Even if the constants a and b are rational, there are a lot of irrational solutions to $y^2 = x^3 + ax + b$. We need to consider the points (x, y) in some sort of *algebraic closure*.

Why does this help us?

Where do the points (x, y) live?. Even if the constants a and b are rational, there are a lot of irrational solutions to $y^2 = x^3 + ax + b$. We need to consider the points (x, y) in some sort of *algebraic closure*.

For example, if a and b are in $\mathbb{Q}$, then the points (x, y) should live in some algebraic closure of $\mathbb{Q}$, say $\mathbb{C}$. Or if $a, b \in \mathbb{F}_p$, then $(x, y) \in \overline{\mathbb{F}_p}^2$

Why does this help us?

Where do the points (x, y) live?. Even if the constants a and b are rational, there are a lot of irrational solutions to $y^2 = x^3 + ax + b$. We need to consider the points (x, y) in some sort of *algebraic closure*.

For example, if a and b are in $\mathbb{Q}$, then the points (x, y) should live in some algebraic closure of $\mathbb{Q}$, say $\mathbb{C}$. Or if $a, b \in \mathbb{F}_p$, then $(x, y) \in \overline{\mathbb{F}_p}^2$

Key feature of the addition law: If the constants a and b and the coordinates of the points points $A = (x_1, y_1)$ and $B = (x_2, y_2)$ both lie in some restricted domain (for example $\mathbb{Q} \subseteq \mathbb{C}$, or $\mathbb{F}_p \subseteq \overline{\mathbb{F}_p}$) then their sum $A + B$ will lie in the same domain.

"Proof" of this fact:

Given the points A and B, form the straight line through them:

$$y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_2) + y_2$$

Substitute into the equation for the curve

$$\left(\frac{y_2 - y_1}{x_2 - x_1}(x - x_2) + y_2\right)^2 = x^3 + ax + b$$

which is a cubic polynomial equation with rational coefficients. It is "obvious" that x_1 and x_2 are two rational roots, so it must be the case that the third root is also rational!

(Note: This is actually a good way of implementing the addition law in a computer too!)

The essence of the RSA algorithm

Lets abstract the number theory out of RSA:

Given a finite abelian group G of order n, then any element $g \in G$ has order dividing n. In particular $g^n = g$.

If Bob chooses a very large group G, and an element $g \in G$ whose order is kept secret, he can have Alice encode her message in the exponent of g! ($<-$ this is me being excited, not a factorial)

Elliptic curve cryptosystems

The way forward is clear: Take the abelian group G to be the additive group of points on an elliptic curve.

In particular, pick some prime p, and choose natural numbers (or elements of $\mathbb{F}_p$) a and b. This data yields an elliptic curve $y^2 = x^3 + ax + b$ over $\mathbb{F}_p$.

Now pick a point $Q = (x, y)$ living on the curve which generates a large subgroup of the group of points on an elliptic curve. Denote the order of this group by n, and it is traditionally chosen to be prime.

The NSA recommends using

$$p = 6277101735386680763835789423207666416083908700390324961279$$

$$a = -3$$

$$b = \texttt{0x64210519 e59c80e7 0fa7e9ab 72243049 feb8deec c146b9b1}$$

$$Q = ($$

$$\texttt{0x188da80e b03090f6 7cbf20eb 43a18800 f4ff0afd 82ff1012},$$

$$\texttt{0x07192b95 ffc8da78 631011ed 6b24cdd5 73f977a1 1e794811})$$

for no particular reason. 😉

Elliptic curve cryptosystems contd...

The rest of the algorithm proceeds in exact analogy with RSA:

Alice and Both choose a pair of numbers s_A and s_B. These will be kept secret. What will be shared though, are the points $Q_A = s_A \cdot Q$ and $Q_B = s_B \cdot Q$ as public keys.

If Alice wants to send a message to Bob she:

1. encodes her plain-text message in an agreed-upon way into a natural number c.
2. calculates $P = c \cdot Q_B$.
3. sends Bob the result P.

Bob recovers the message!

Bob knows the s_B that got Q_B, so he can easily calculate some t_B for which $s_B \cdot t_B \equiv 1 \mod n$. Therefore, Bob calculates

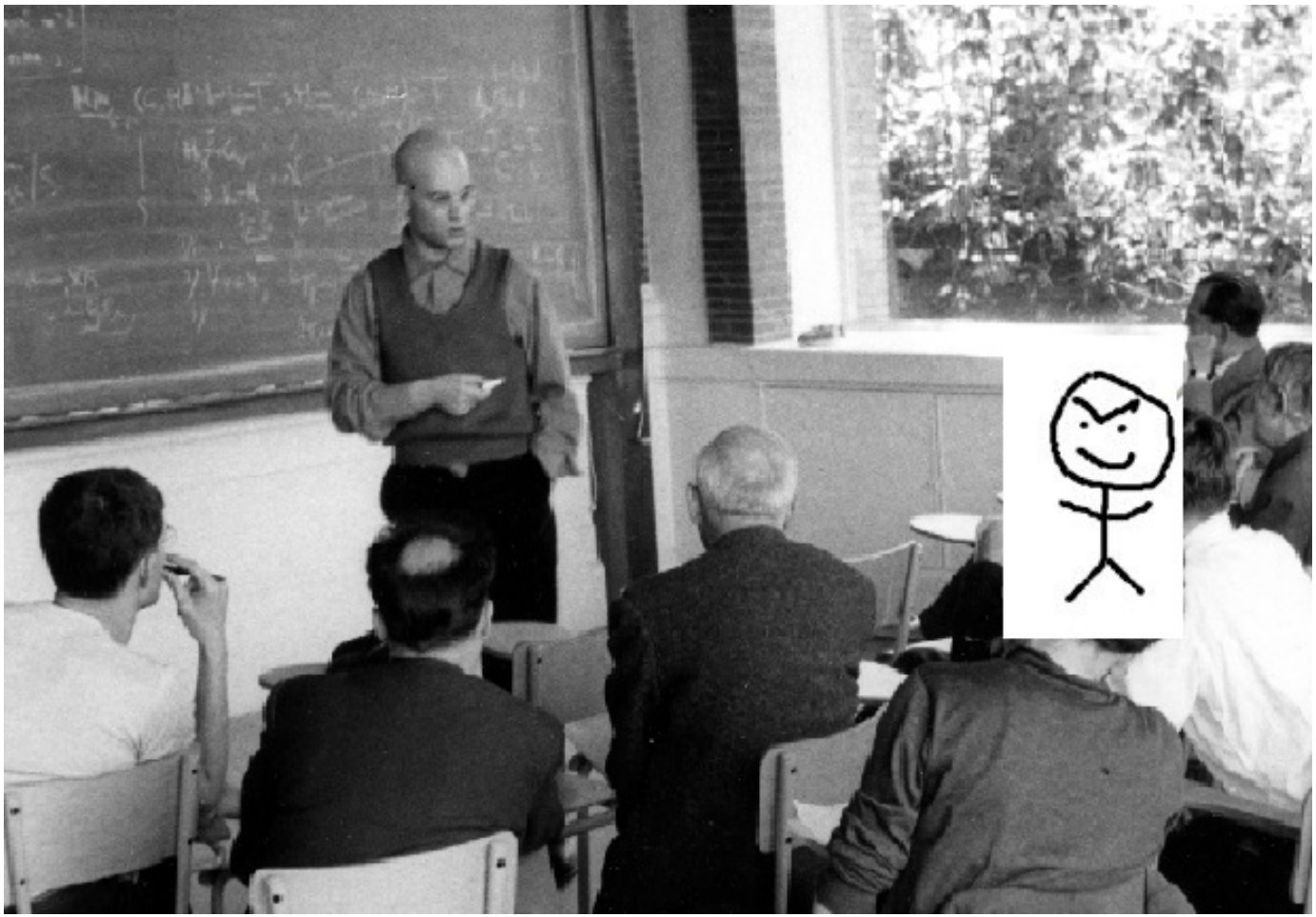
$$t_B \cdot P = t_B \cdot c \cdot Q_B = (t_B \cdot c \cdot s_B) \cdot Q = c \cdot Q$$

and recovers c.

Carol on the other hand, only knows Q_B, so her best bet in decrypting the message is just calculating $k \cdot Q_B$ for all k until luckily stumbling upon the right power s_B

If p and n are very large, this could potentially take a VERY long time.

Carol goes back to grad school



Benefits and limitations of this protocol

The good:
The size of the keys and hashes needed with elliptic curves are much smaller than
This is actually used! Security on bluetooth connections uses ECDH, as do many other secure forms of communication.

Benefits and limitations of this protocol

The good:
The size of the keys and hashes needed with elliptic curves are much smaller than
This is actually used! Security on bluetooth connections uses ECDH, as do many other secure forms of communication.

The bad:
The same algorithms (Schor's algorithm) that are used to crack RSA on quantum computers can be used on elliptic curve cryptography...

Benefits and limitations of this protocol

The good:
The size of the keys and hashes needed with elliptic curves are much smaller than
This is actually used! Security on bluetooth connections uses ECDH, as do many other secure forms of communication.

The bad:
The same algorithms (Schor's algorithm) that are used to crack RSA on quantum computers can be used on elliptic curve cryptography...

The ugly:
Turns out quantum computers are even better at cracking elliptic curve cryptography than they are solving problems in modular arithmetic...

An interesting observation...

Carol notices that the number of points N on an elliptic curve over a finite field $\mathbb{F}_p$ is always pretty close to p...
For example for the curve above:

$$p = 627710173538668076383578942320766641608390870039032496127$$
$$N = 627710173538668076383578942317605901376719477318284228408$$

An interesting observation...

Carol notices that the number of points N on an elliptic curve over a finite field $\mathbb{F}_p$ is always pretty close to p...
For example for the curve above:

$$p = 627710173538668076383578942320766641608390870039032496127$$
$$N = 627710173538668076383578942317605901376719477318284228408$$

In fact, it always seems like

$$|N - (p + 1)| \leq 2\sqrt{p}$$

An interesting observation...

Carol notices that the number of points N on an elliptic curve over a finite field $\mathbb{F}_p$ is always pretty close to p...
For example for the curve above:

$$p = 627710173538668076383578942320766641608390870039032496127$$
$$N = 627710173538668076383578942317605901376719477318284228408$$

In fact, it always seems like

$$|N - (p+1)| \leq 2\sqrt{p}$$