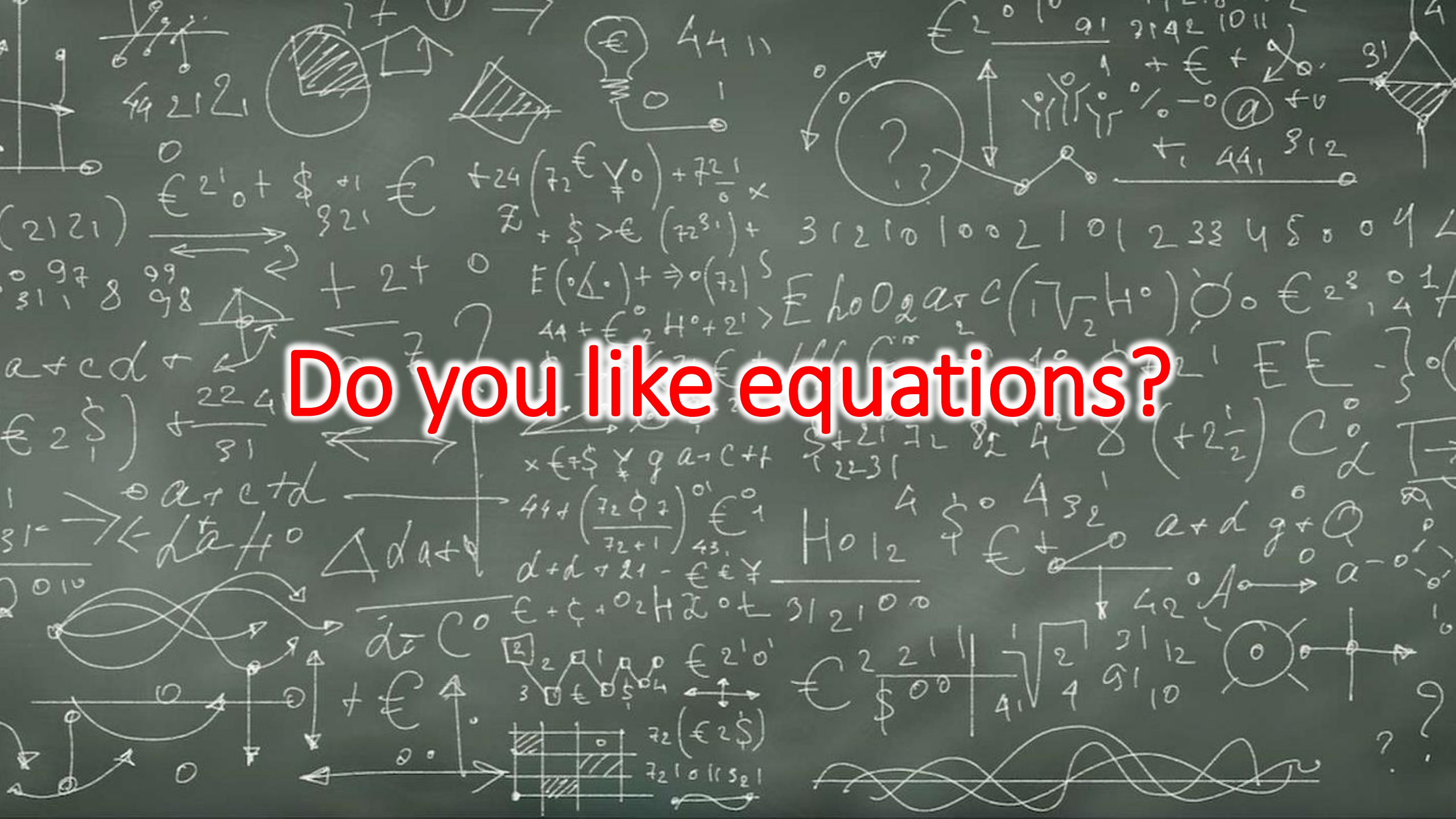
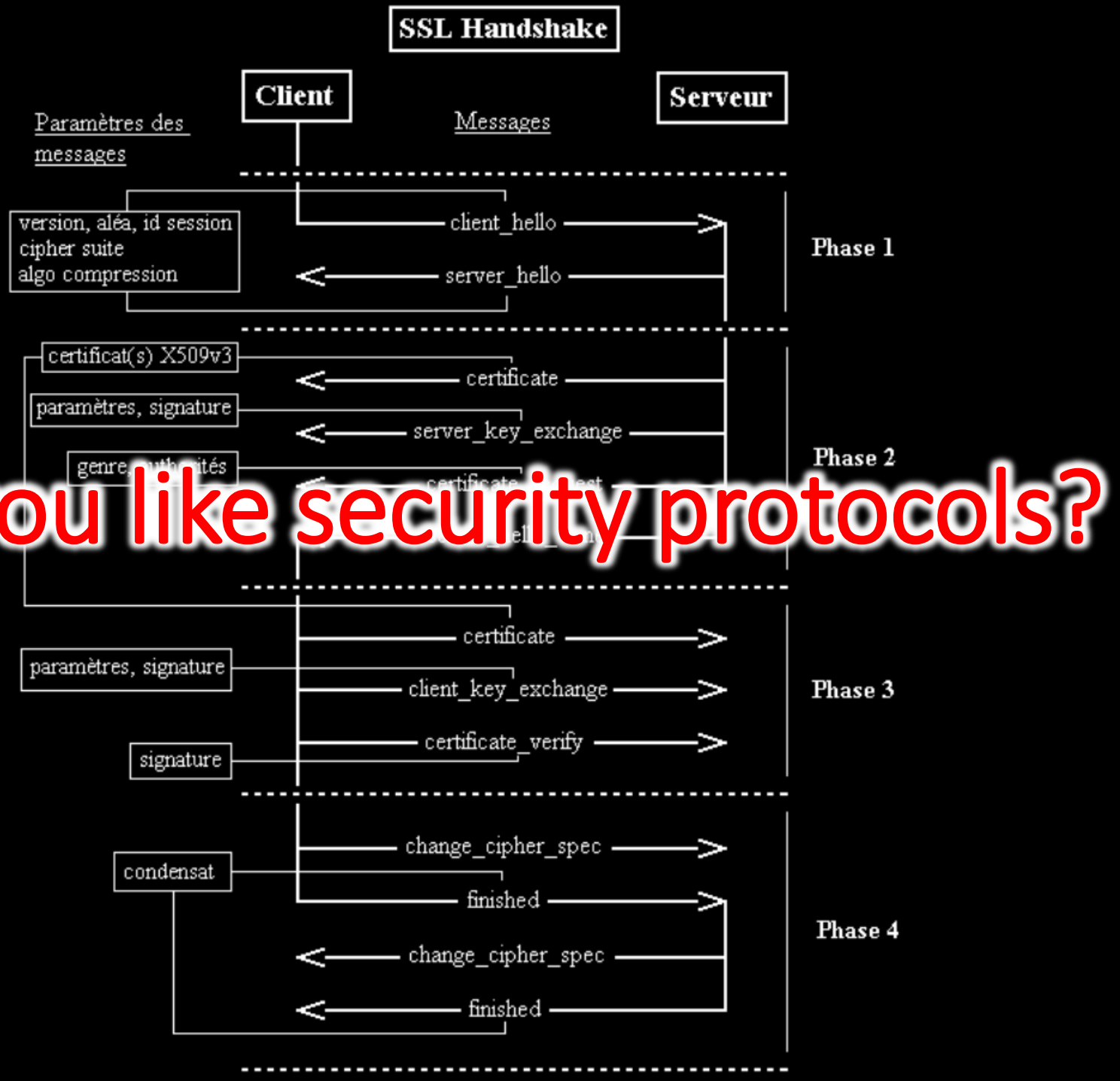
# Asymmetric Cryptography

Paul Dubois

for

info@lèze


Do you like equations?

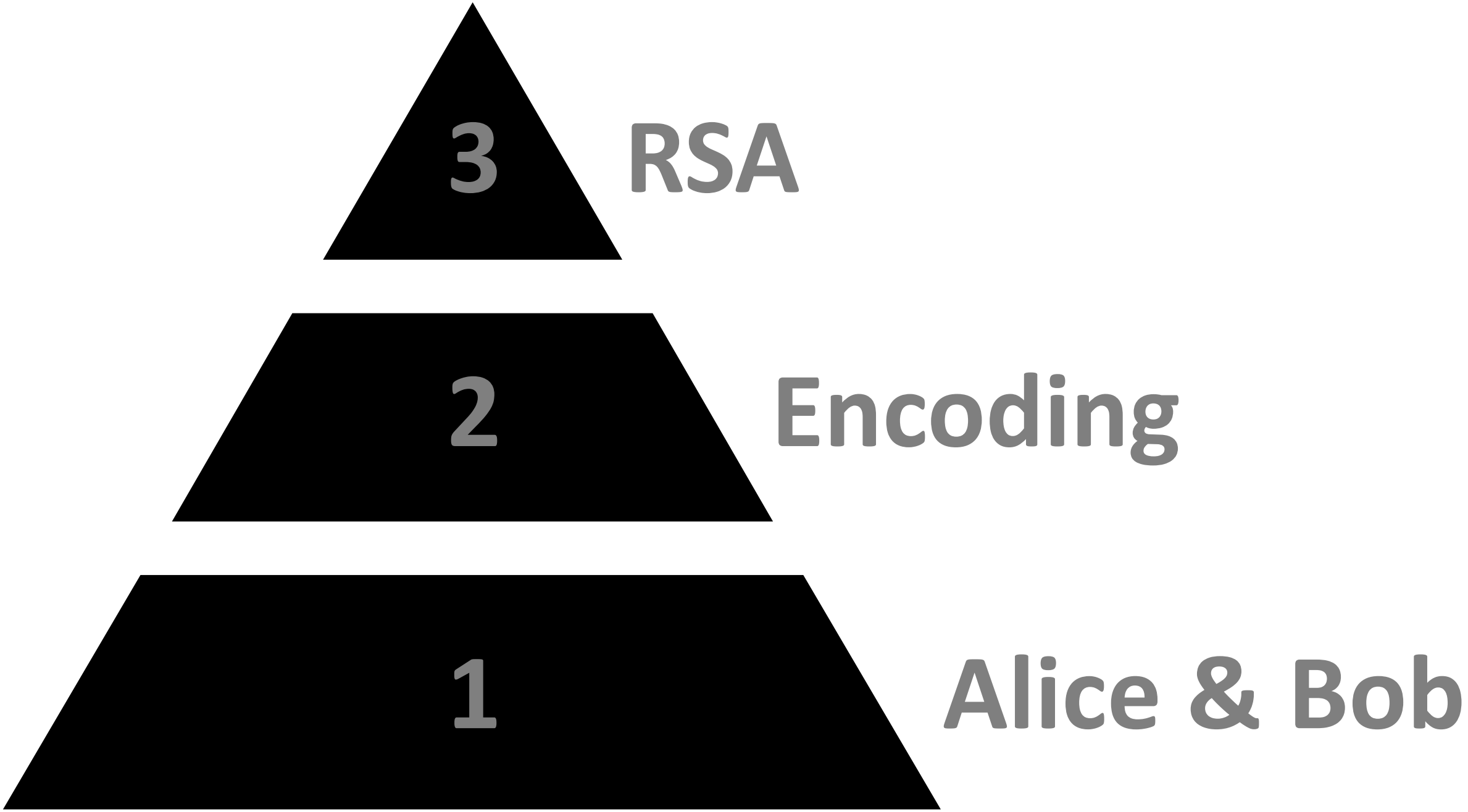



Do you like security protocols?


Do you like money?


Do you like Toulouse?

# How to send a secret…
# … with a speaker?


3
RSA
2
Encoding
1
Alice & Bob

5 Elliptic Curves

4 ElGamal

3 RSA

2 Encoding

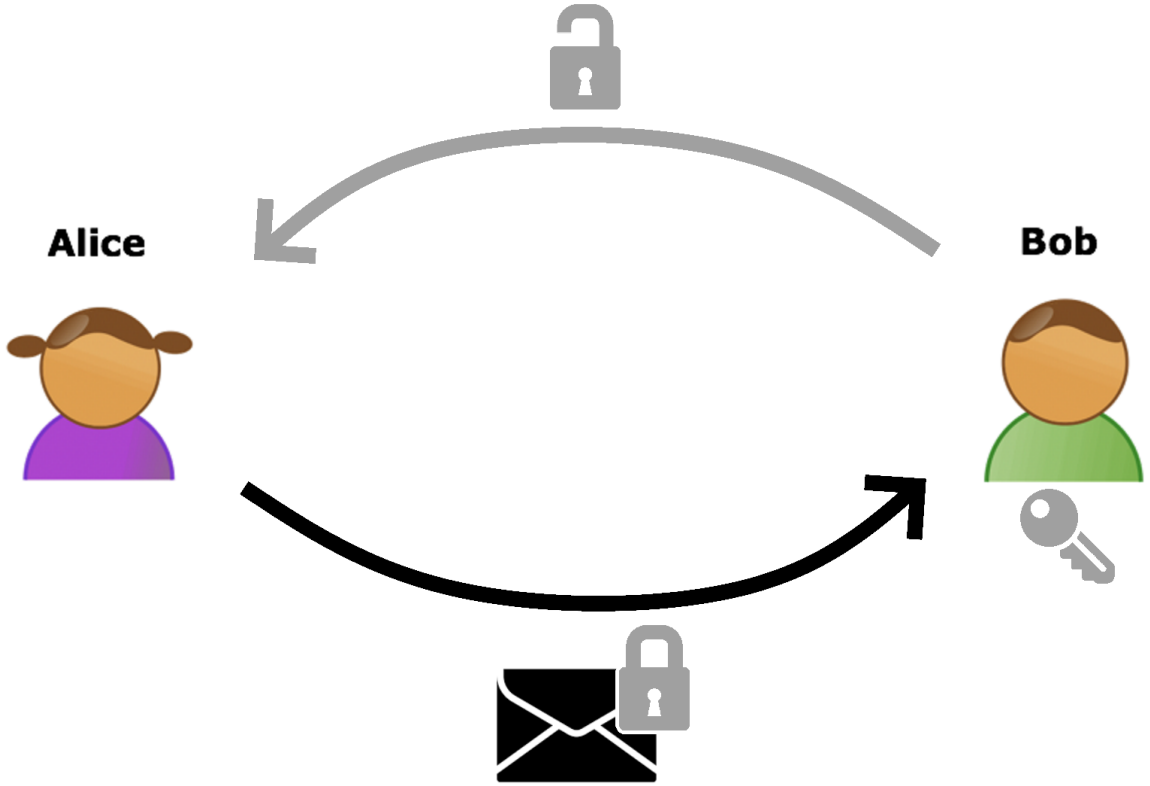
1 Alice & Bob

2nd Part

1st Part

# Level 1

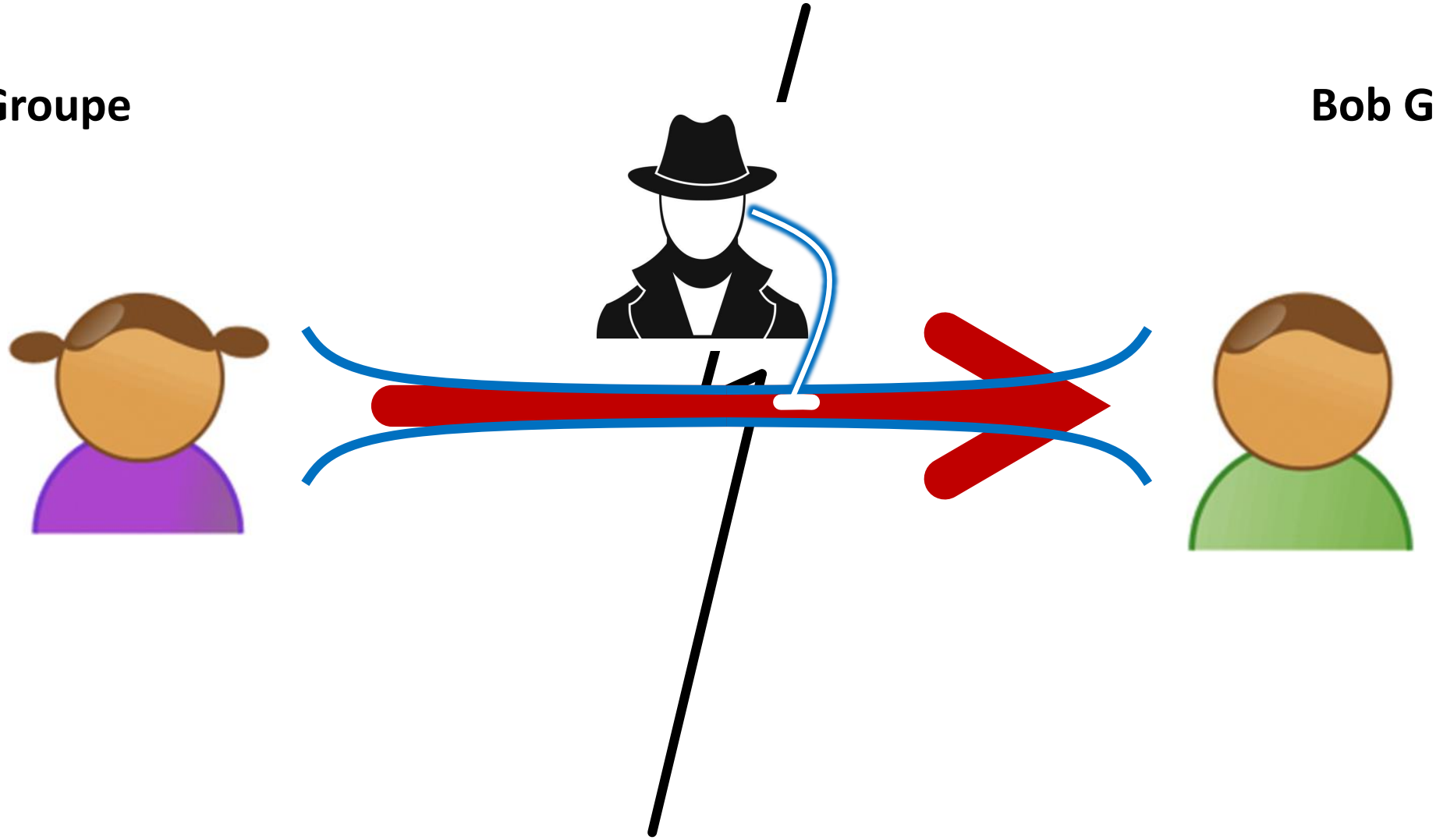
Lock Analogy



# Send a message without a secured connection

**Alice Groupe**

**Bob Groupe**

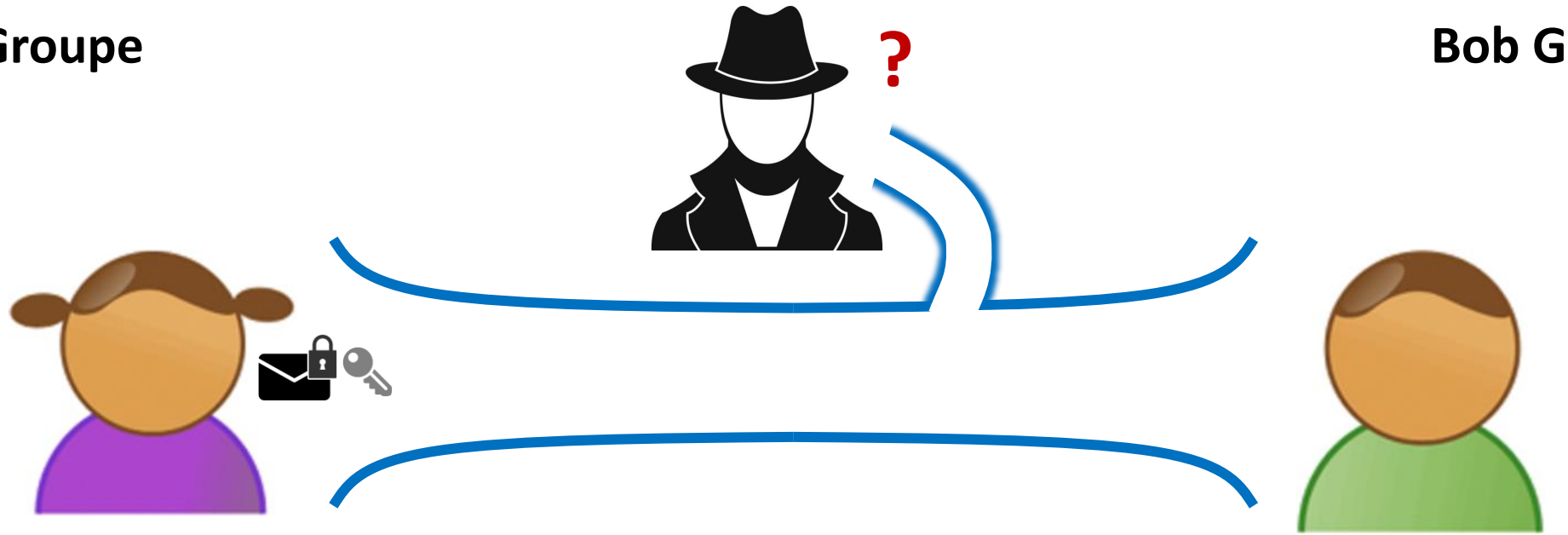


# Do it yourself

# *Send a message without a secured connection:*

## Intuition

**Alice Groupe**



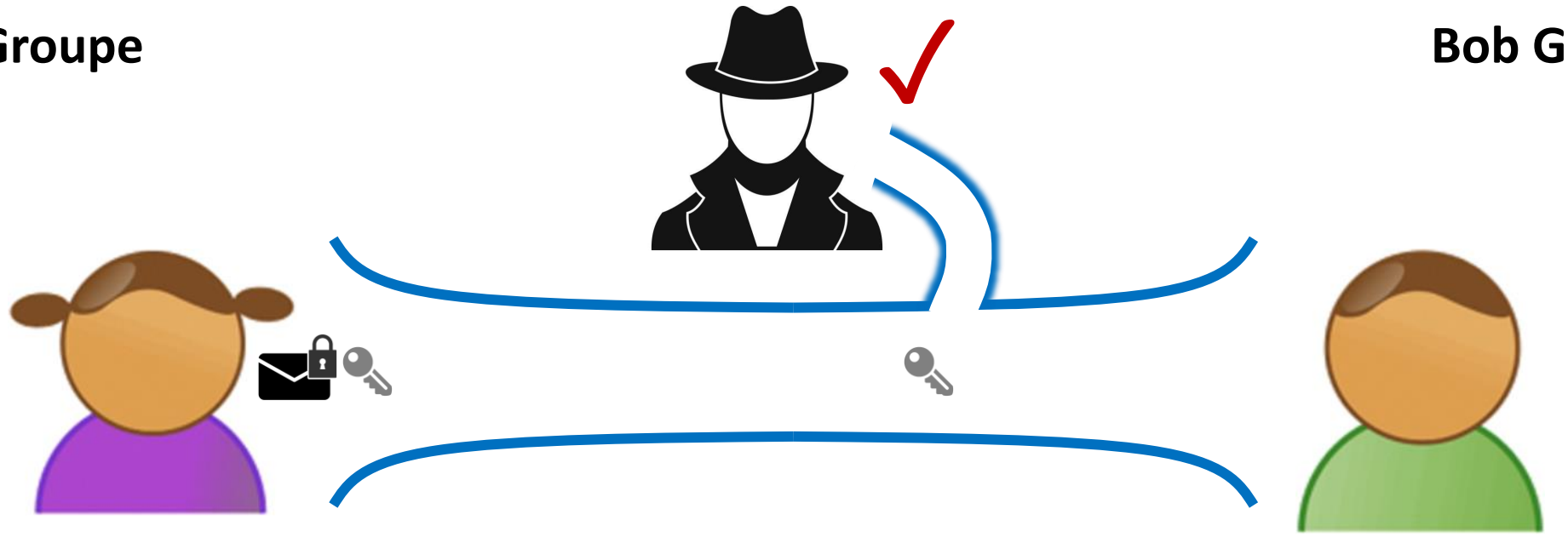

**Bob Groupe**

# *Send a message without a secured connection:*

## Reality

**Alice Groupe**

**Bob Groupe**

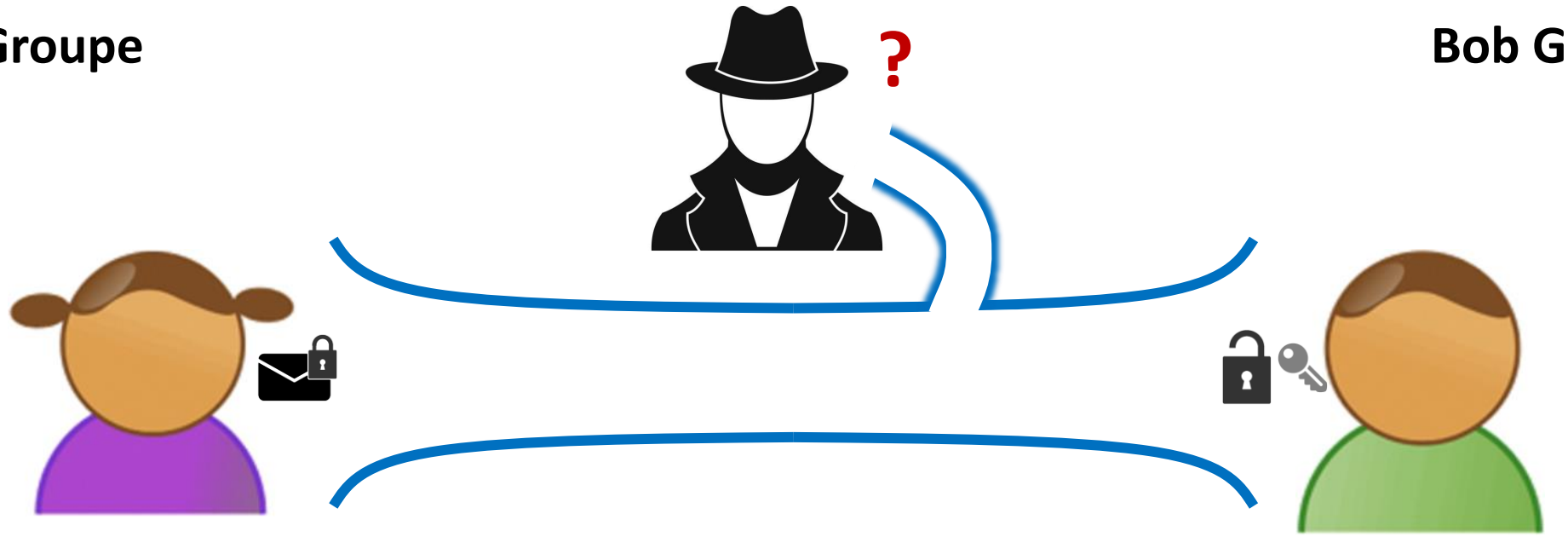


# *Send a message without a secured connection:*

## Solution

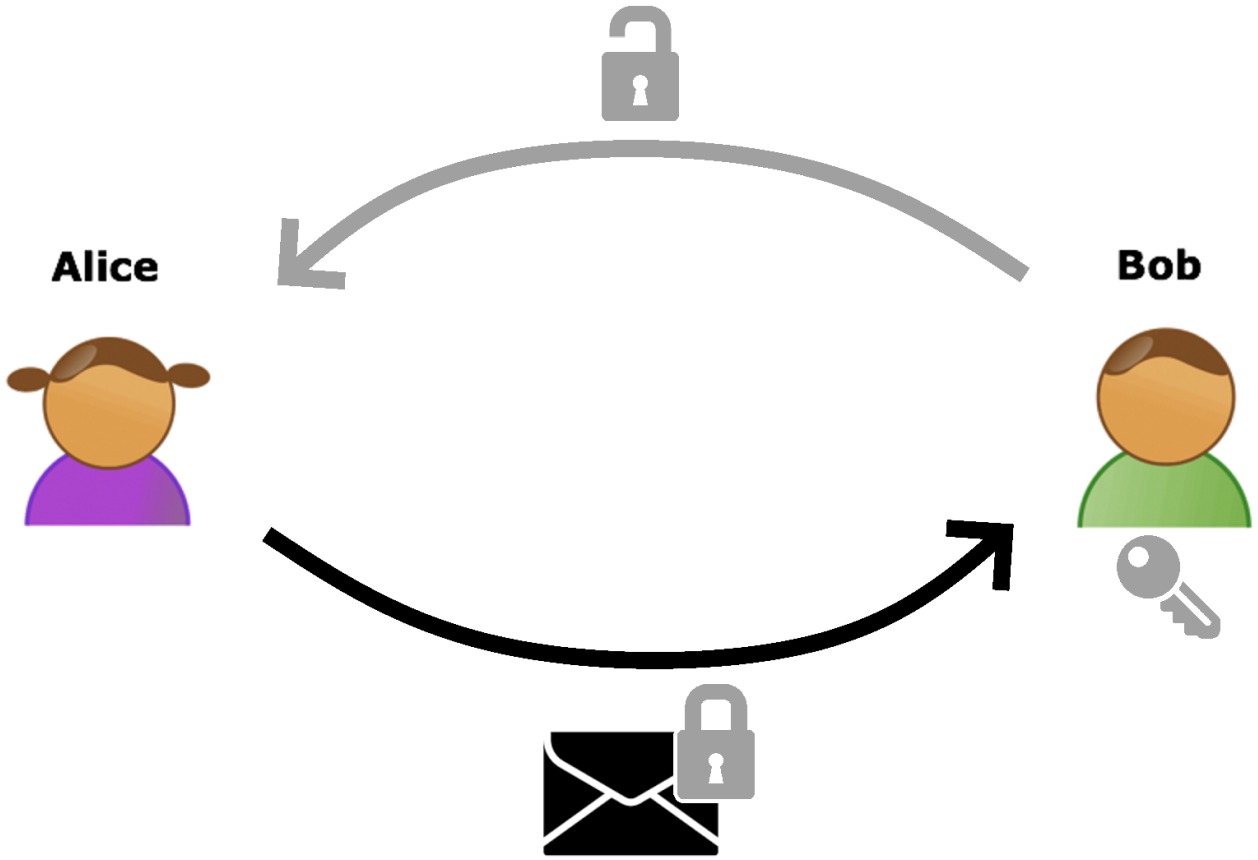
**Alice Groupe**

**Bob Groupe**



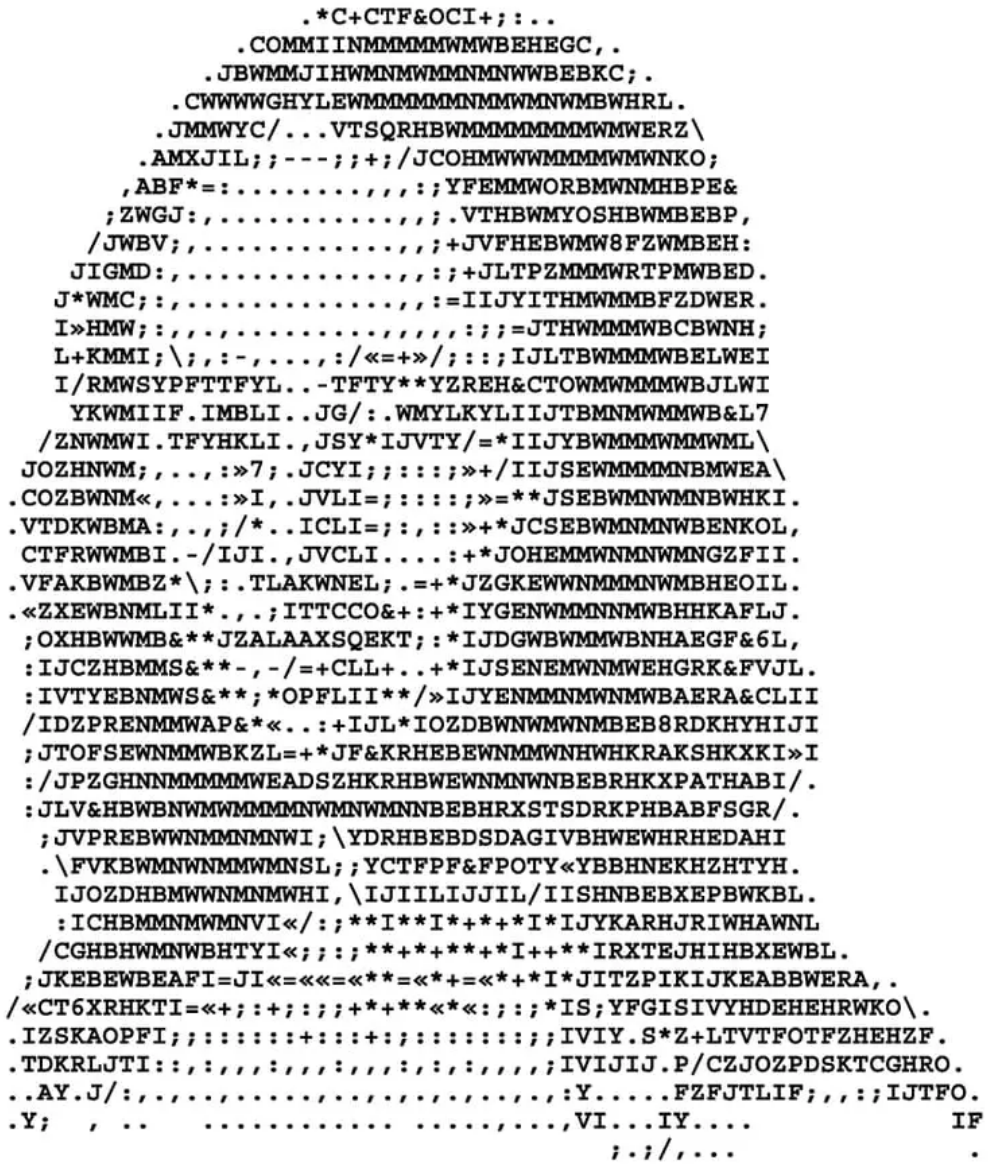
*Send a message without a secured connection:*

# Synthesis



# Level 2

Encoding of a message numerically



*Non-contractual image*

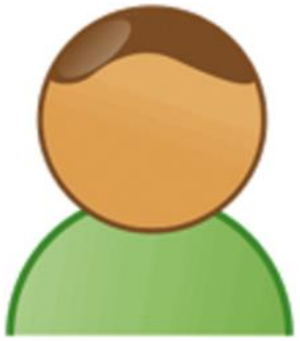
# Coding a message

**Alice Groupe**

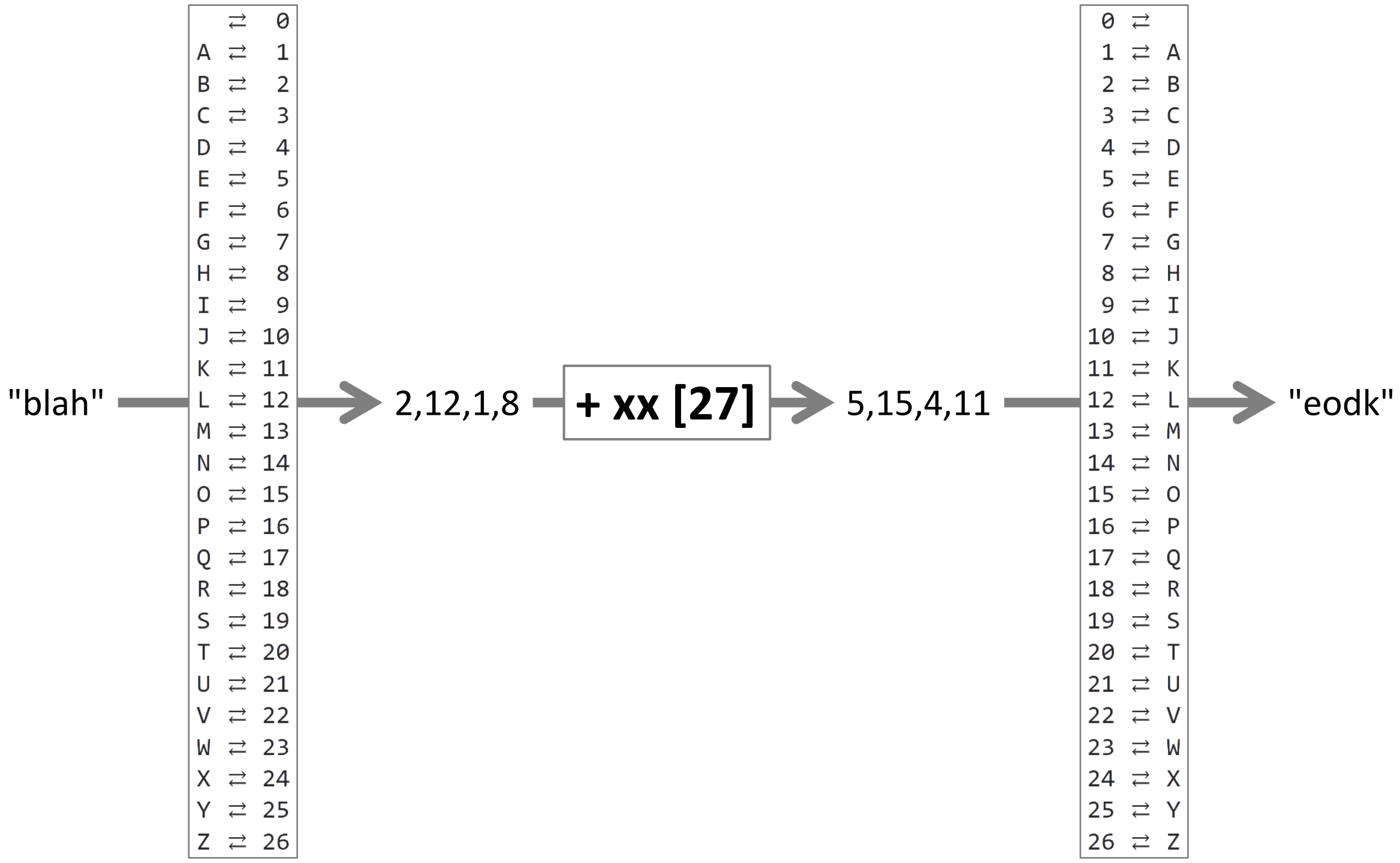
**Bob Groupe**

Code: ???

Code: xxx



Code: xxx

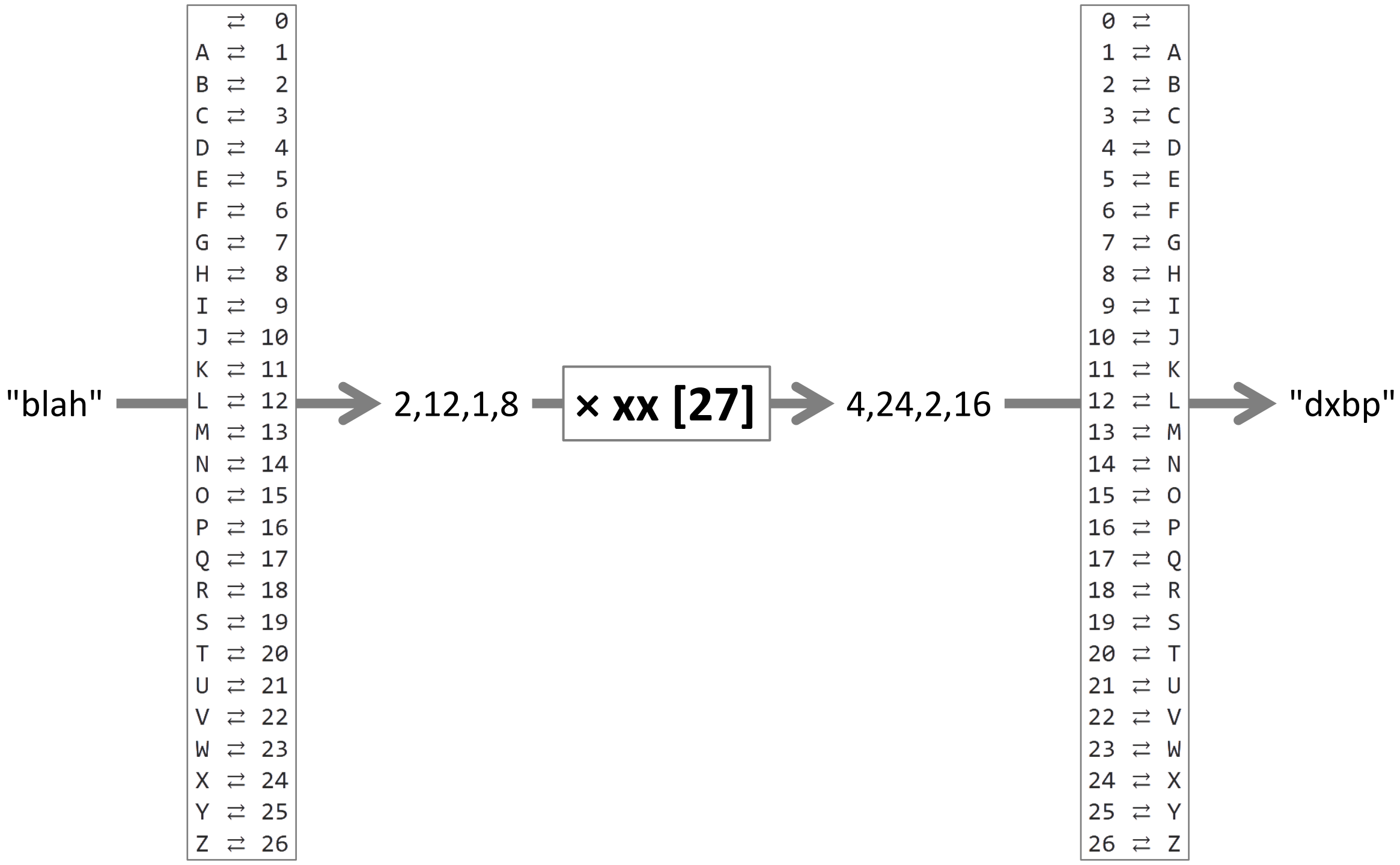

"blah"
2,12,1,8
+ xx [27]
5,15,4,11
"eodk"
A ⇄ 1
B ⇄ 2
C ⇄ 3
D ⇄ 4
E ⇄ 5
F ⇄ 6
G ⇄ 7
H ⇄ 8
I ⇄ 9
J ⇄ 10
K ⇄ 11
L ⇄ 12
M ⇄ 13
N ⇄ 14
O ⇄ 15
P ⇄ 16
Q ⇄ 17
R ⇄ 18
S ⇄ 19
T ⇄ 20
U ⇄ 21
V ⇄ 22
W ⇄ 23
X ⇄ 24
Y ⇄ 25
Z ⇄ 26
1 ⇄ A
2 ⇄ B
3 ⇄ C
4 ⇄ D
5 ⇄ E
6 ⇄ F
7 ⇄ G
8 ⇄ H
9 ⇄ I
10 ⇄ J
11 ⇄ K
12 ⇄ L
13 ⇄ M
14 ⇄ N
15 ⇄ O
16 ⇄ P
17 ⇄ Q
18 ⇄ R
19 ⇄ S
20 ⇄ T
21 ⇄ U
22 ⇄ V
23 ⇄ W
24 ⇄ X
25 ⇄ Y
26 ⇄ Z

# Representation of the encoding system

(addition)

https://pauldubois98.github.io/AsymmetricCryptographyTalk/tools/code_add.html

# Do it yourself


"blah"
2,12,1,8
× xx [27]
4,24,2,16
"dxbp"
⇄ 0
A ⇄ 1
B ⇄ 2
C ⇄ 3
D ⇄ 4
E ⇄ 5
F ⇄ 6
G ⇄ 7
H ⇄ 8
I ⇄ 9
J ⇄ 10
K ⇄ 11
L ⇄ 12
M ⇄ 13
N ⇄ 14
O ⇄ 15
P ⇄ 16
Q ⇄ 17
R ⇄ 18
S ⇄ 19
T ⇄ 20
U ⇄ 21
V ⇄ 22
W ⇄ 23
X ⇄ 24
Y ⇄ 25
Z ⇄ 26
0 ⇄
1 ⇄ A
2 ⇄ B
3 ⇄ C
4 ⇄ D
5 ⇄ E
6 ⇄ F
7 ⇄ G
8 ⇄ H
9 ⇄ I
10 ⇄ J
11 ⇄ K
12 ⇄ L
13 ⇄ M
14 ⇄ N
15 ⇄ O
16 ⇄ P
17 ⇄ Q
18 ⇄ R
19 ⇄ S
20 ⇄ T
21 ⇄ U
22 ⇄ V
23 ⇄ W
24 ⇄ X
25 ⇄ Y
26 ⇄ Z

# Representation of the encoding system

(multiplication)

https://pauldubois98.github.io/AsymmetricCryptographyTalk/tools/code_mult.html

# Principals conclusions

- **Representation** of the alphabet by **numbers**
- **Modulo** arithmetic so that the range of number used do not explode
- Coding with **multiplication** more complex than with addition
- Encryption is **symmetric**

**« + » :**

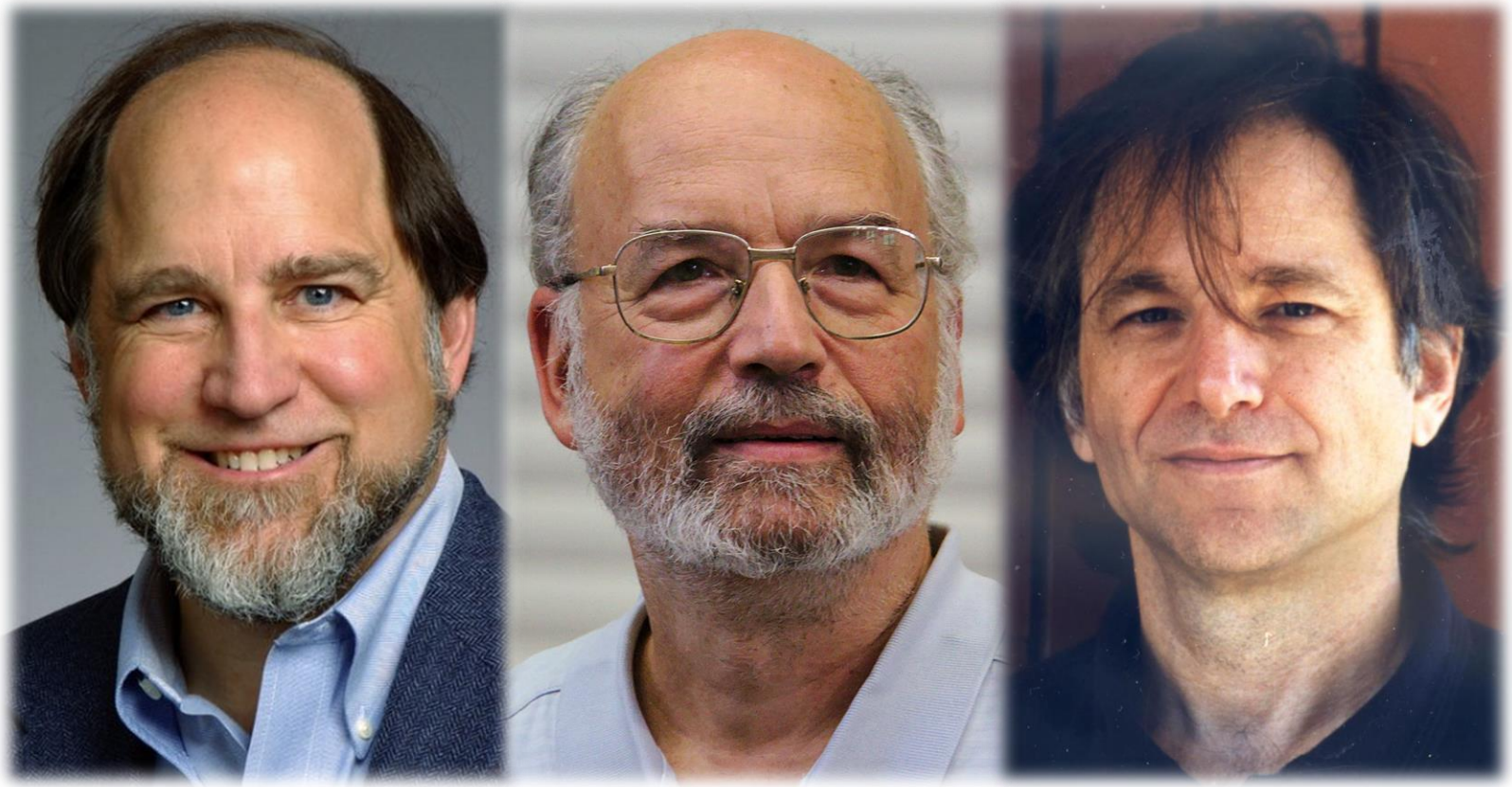
- **Low costs** in terms of computations

**« - » :**

- Need a **code key** that the **spy doesn’t know**

# Level 3

RSA Encryption



*Ronald **R**ivest* *Adi **S**hamir* *Leonard **A**dleman*

# What's a prime number?

« a number that we cannot factorize »

6 =2*3 ✗

7 =1*7, 7*1, 7 ✓ Trivial !

11 ✓ 12 ✗ 35 ✗

# Communication via a spy

*(without private key)*

$m$, the message

$c = m^e [n]$

$(n, e)$

$c$

$p$,$q$ primes

$n = pq$

$\varphi(n) = (p - 1)(q - 1)$

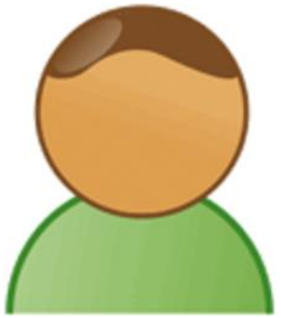
$e$ such that:

- $e < \varphi(n)$
- $pgcd(e, \varphi(n)) = 1$

$d = e^{-1}[\varphi(n)]$

$m = c^d\ [n]$

« Fermat's theorem »



# RSA Helper

https://pauldubois98.github.io/AsymmetricCryptographyTalk/tools/rsa_helper.html

# Do it yourself

$m$, the message

$c = m^e [n]$

$(n, e)$

$c$

$p$,$q$ primes

$n = pq$

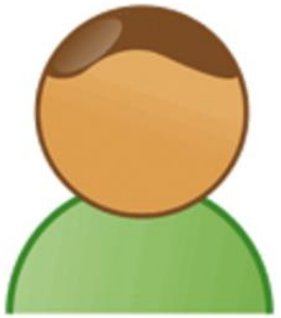
$\varphi(n) = (p - 1)(q - 1)$

$e$ such that:

- $e < \varphi(n)$
- $pgcd(e, \varphi(n)) = 1$

$d = e^{-1}[\varphi(n)]$

$m = c^d\ [n]$



# Principals conclusions

- The spy is **blocked** by the **factorization** of $n$ into $p$ et $q$
- While the **multiplication** $p * q = n$ is **fast**
- Need to **generate large prime numbers** ($p$ et $q$)
- Encryption is **NOT symmetric**

**« + »:**

- **No** need of a **private key**

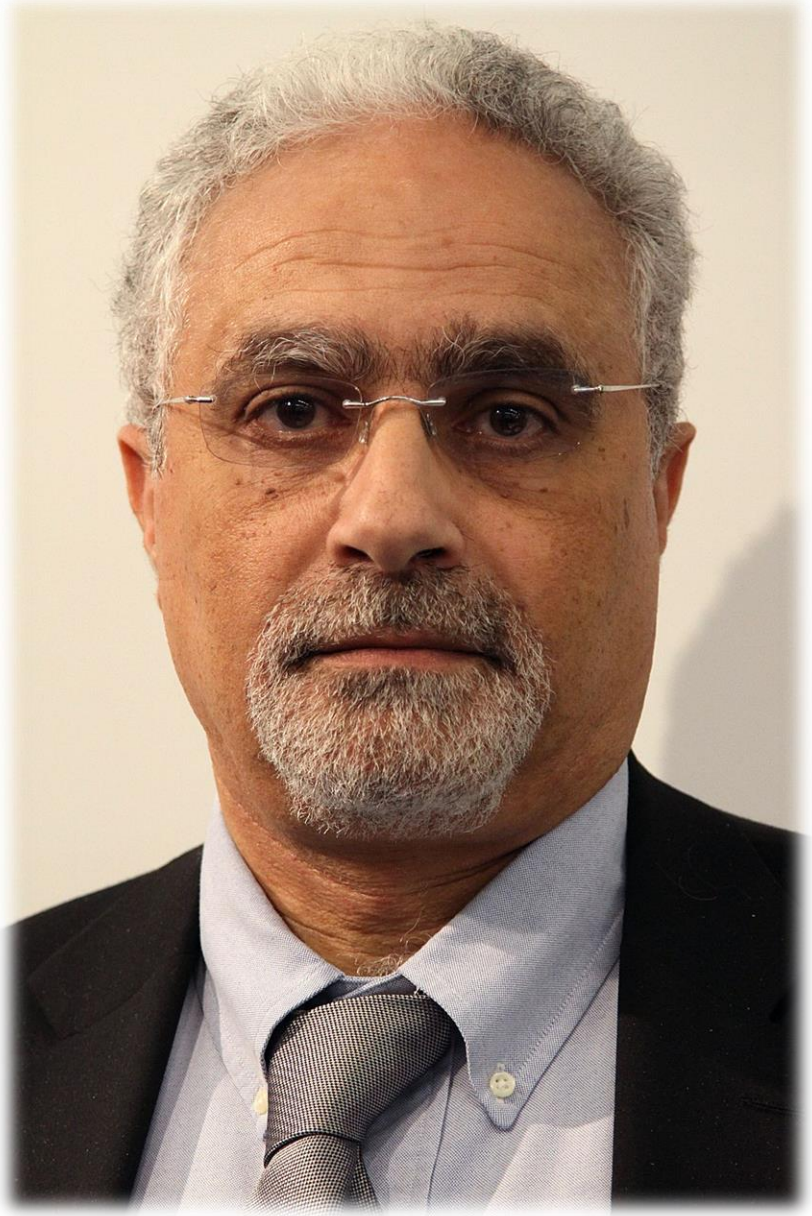
**« - »:**

- **Expensive** in terms of computations
- Subject to **quantum computers attacks**

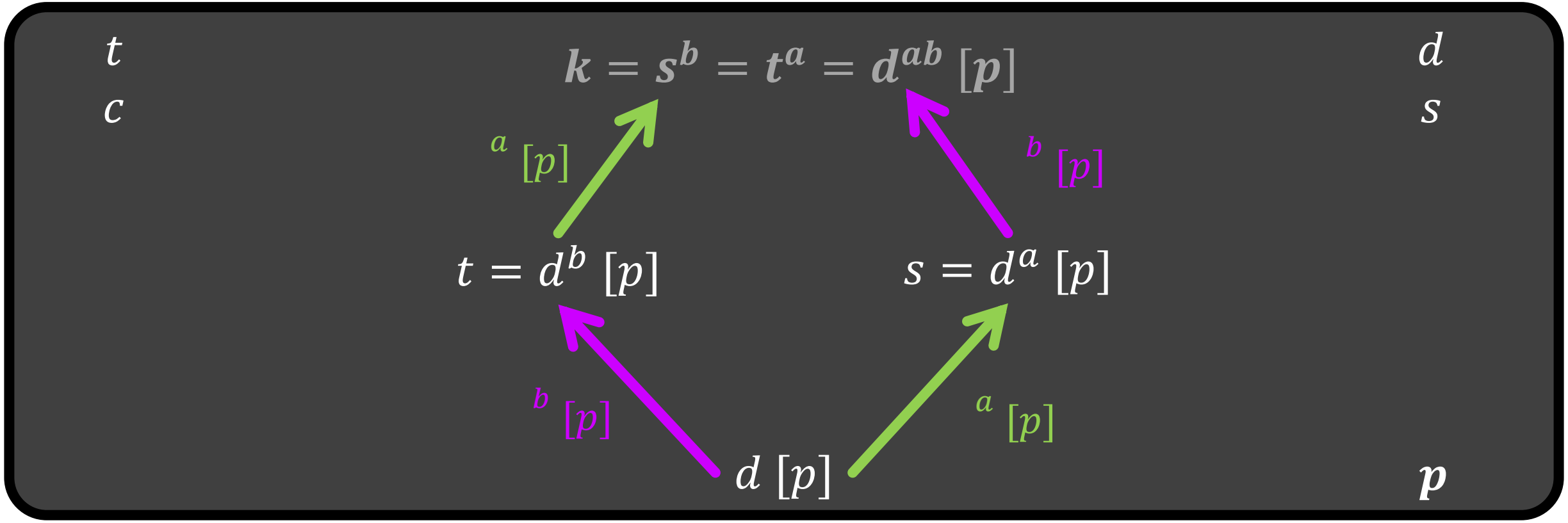
$\Rightarrow$ Used to **exchange a private key** (and then symmetric encryption is used)

# Level 4

ElGamal Encryption



*ElGamal*



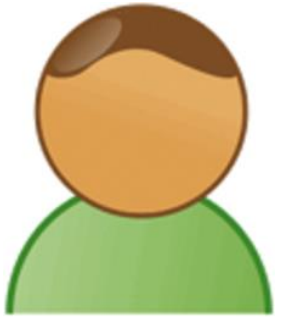

$b$

$m$

$t = d^b$ [p]

$k = s^b$ [p]

$c = mk$ [p]



$p, a, d$

$s = d^a$ [p]

$k = t^a$ [p]

$u = k^{-1}$[p]

$m = cu$ [p]

# Modulo Logarithm

$$s = d^{a}\ [p]$$

$$s = d^{a}$$

$$\log(s) = \log(d^{a})$$

$$= a \cdot \log(d)$$

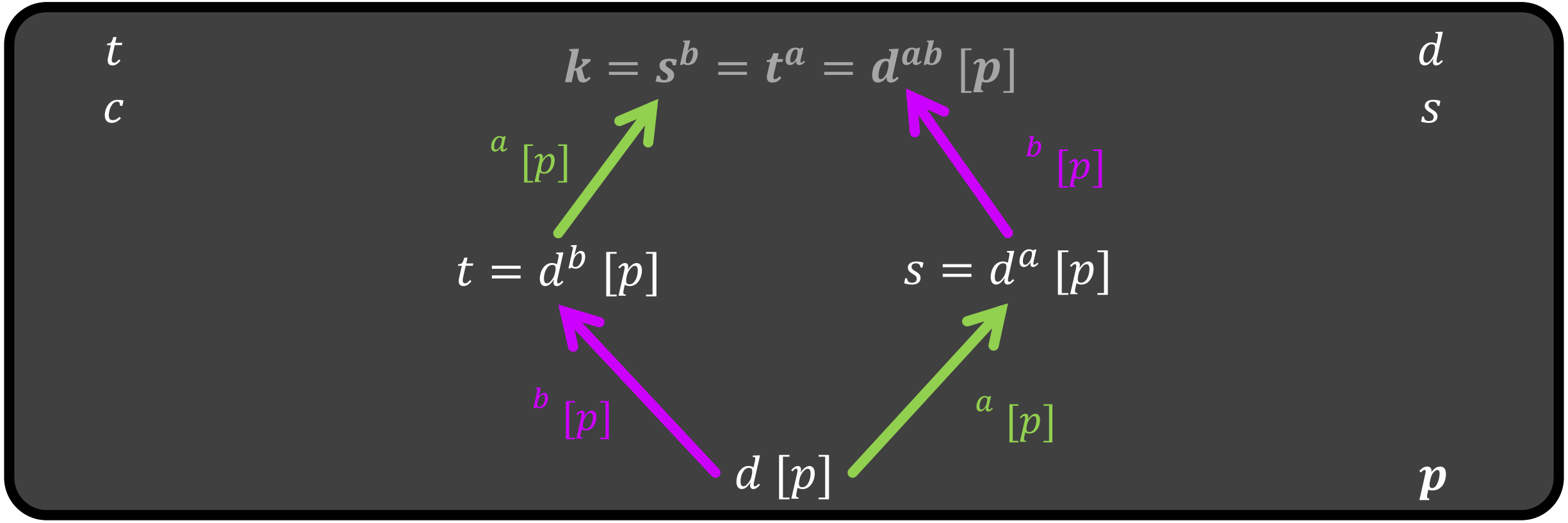
$$a = \frac{\log(s)}{\log(d)}$$

$$= \log(s - d)$$

# Modulo Calculator

https://pauldubois98.github.io/AsymmetricCryptographyTalk/tools/mod_calculator.html

# Do it yourself



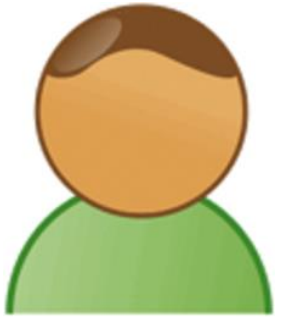

$b$

$m$

$t = d^b$ [p]

$k = s^b$ [p]

$c = mk$ [p]



$p, a, d$

$s = d^a$ [p]

$k = t^a$ [p]

$u = k^{-1}$[p]

$m = cu$ [p]

# Principals conclusions

- The spy is blocked by the fact that he can't compute $\boldsymbol{k}$ (using only public information) nor can he find $a$ or $b$ (modulo logarithm).
- **Pseudo-symmetric** encryption

**« + »:**

- **No** need of a **private key**

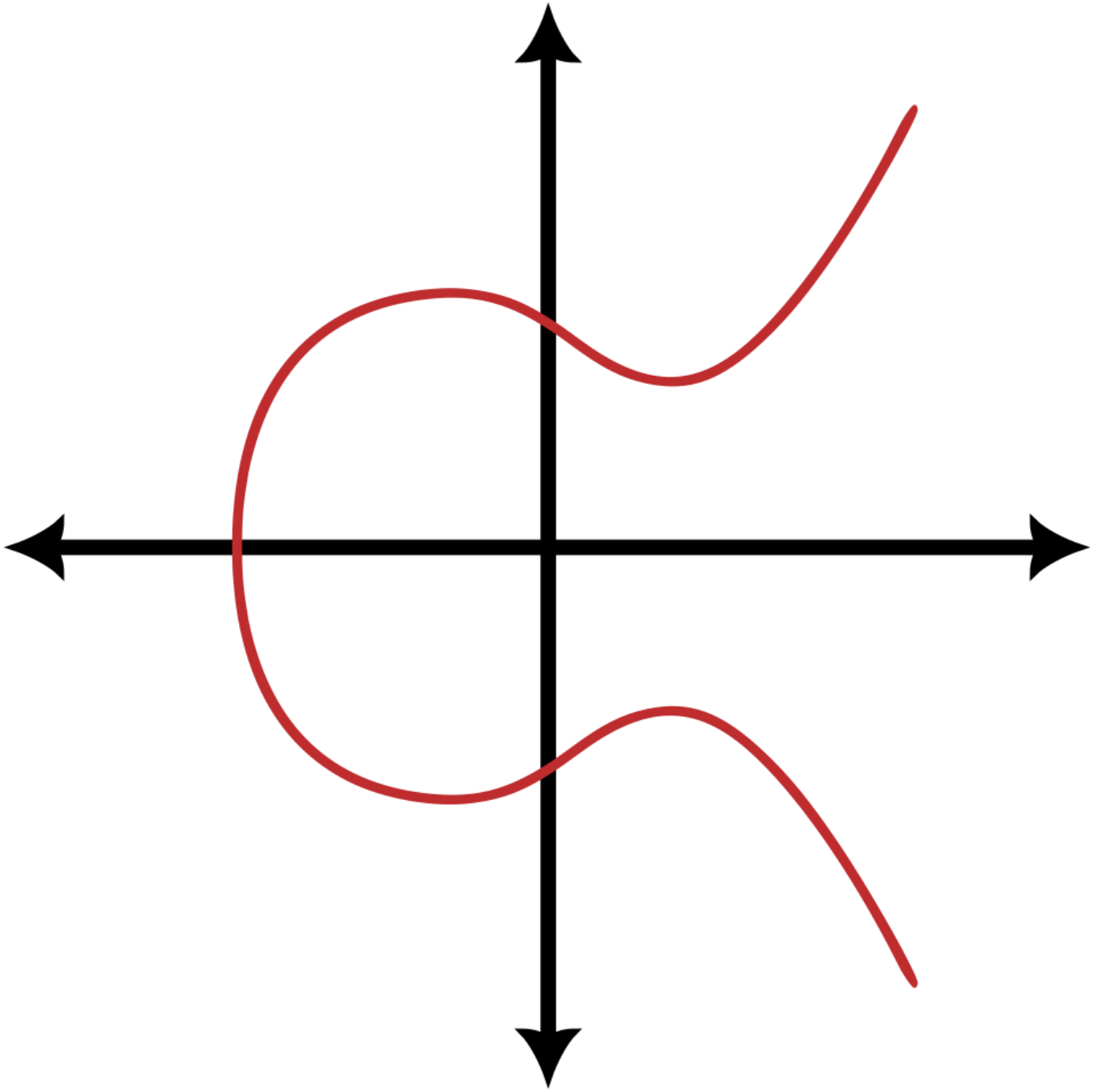
**« - »:**

- **Expensive** in terms of computations
- Subject to some **algebraic attacks**, that may **alter** the **message**

$\Rightarrow$ The system may be applied to otherr contexts than modulo arithmetic

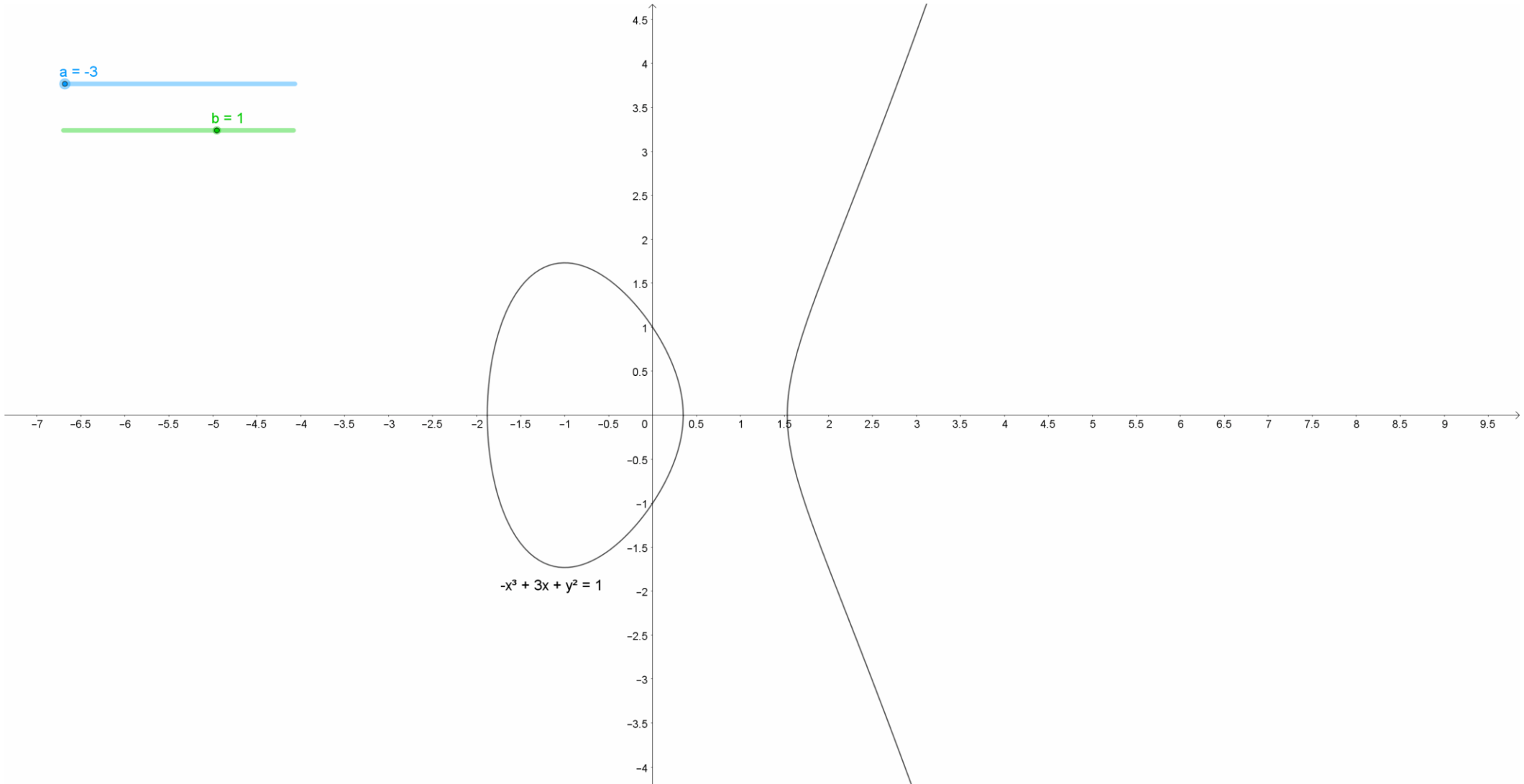
# Level 5

Elliptic Curves Cryptography



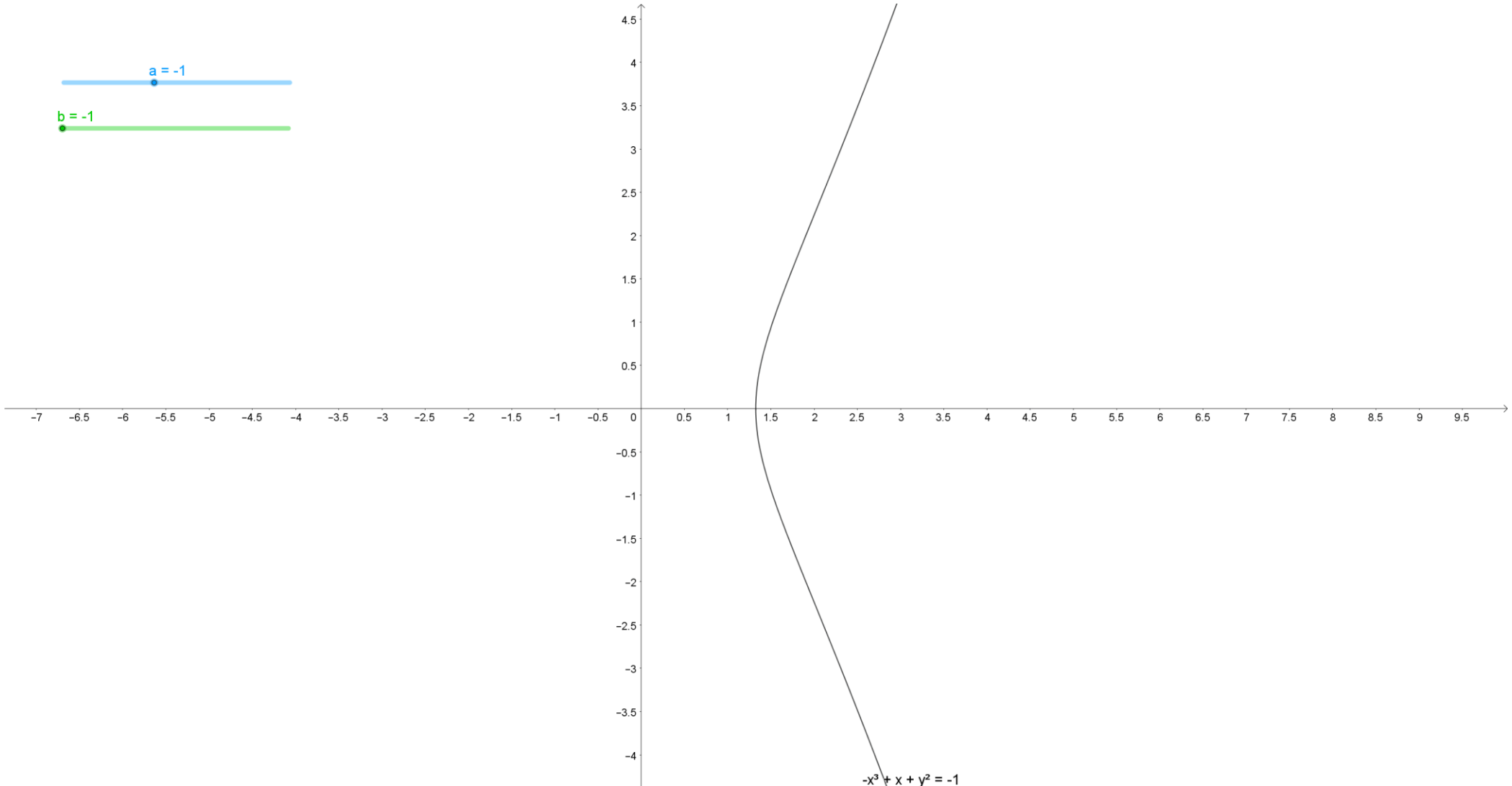
# What's an Elliptic Curve?

$$y^2 = x^3 + ax + b$$



# What's an Elliptic Curve?

$$y^2 = x^3 + ax + b$$



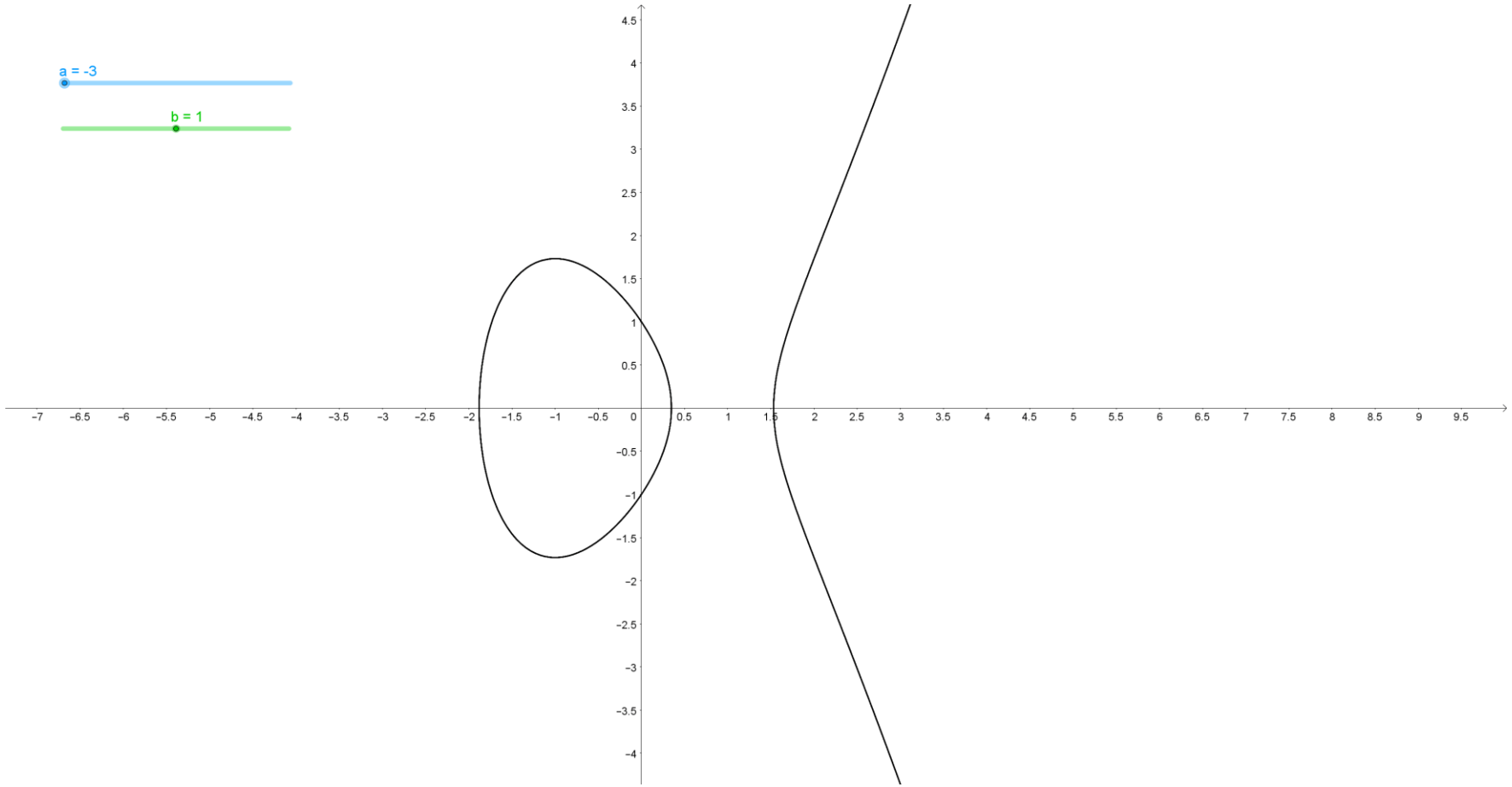
# What's an Elliptic Curve?

$$y^2 = x^3 + ax + b$$

https://pauldubois98.github.io/AsymmetricCryptographyTalk/tools/elliptic_curve.html

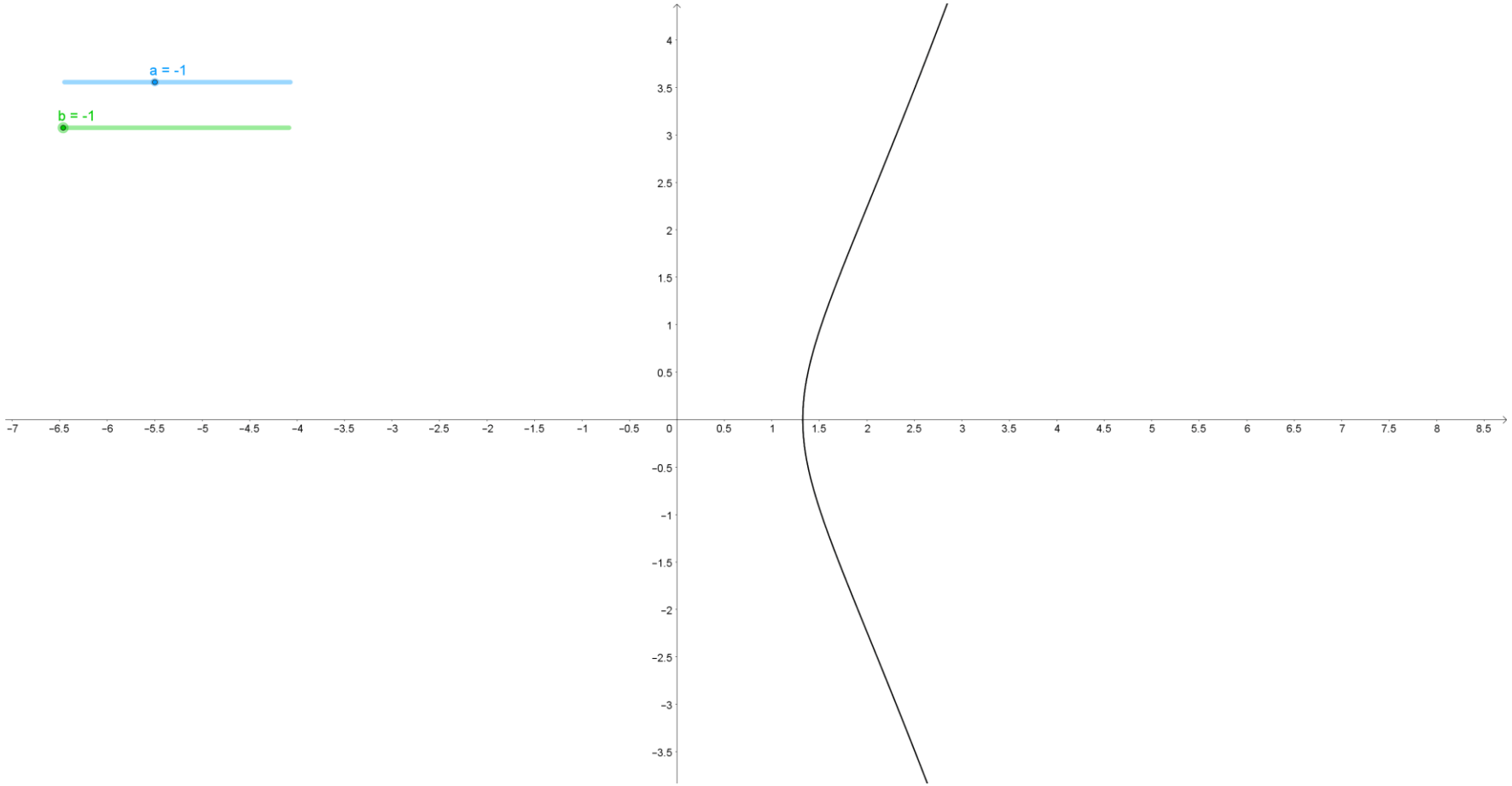
# What's an Elliptic Curve?

$$y^2 = x^3 + ax + b$$



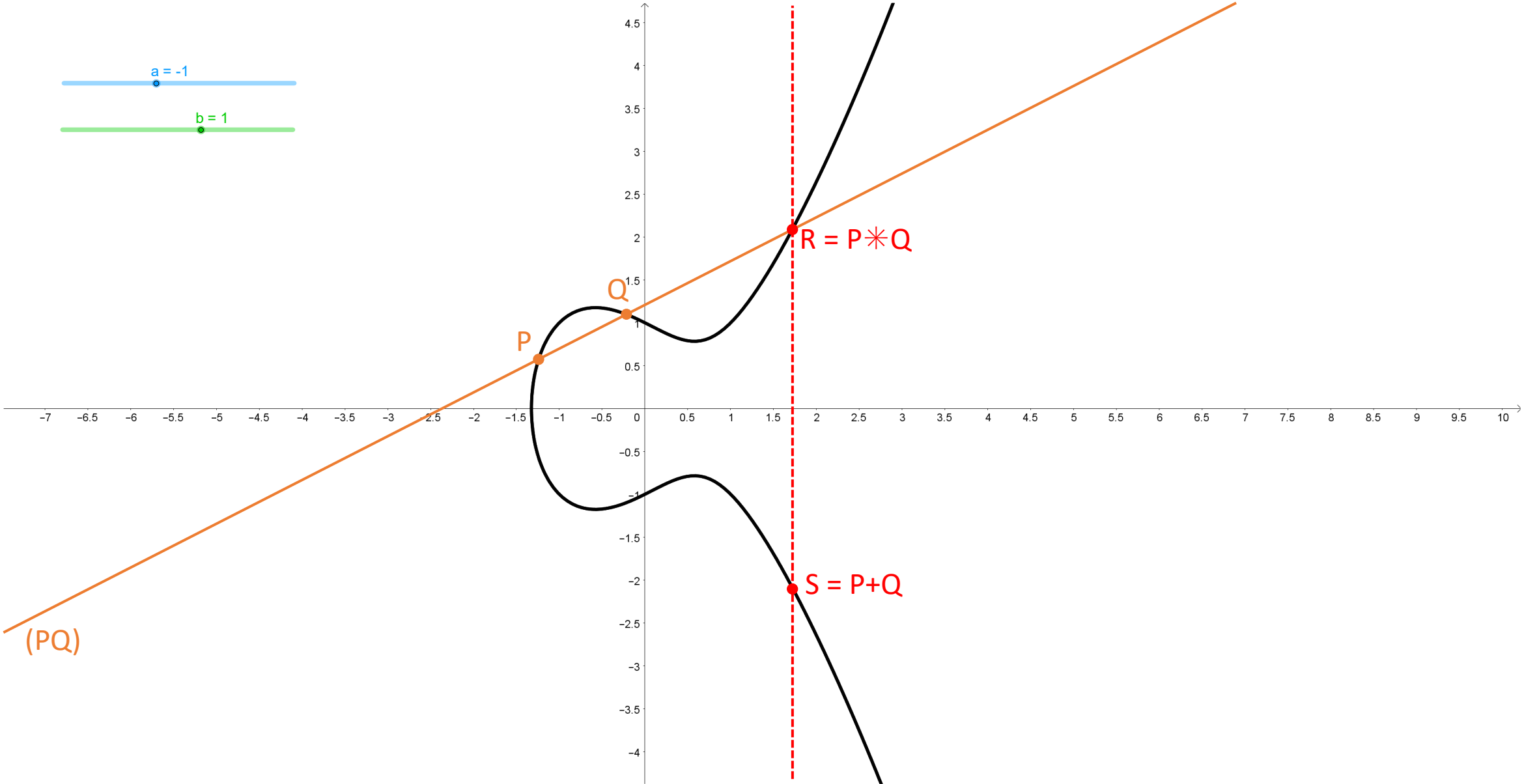
# What's an Elliptic Curve?

$$y^2 = x^3 + ax + b$$



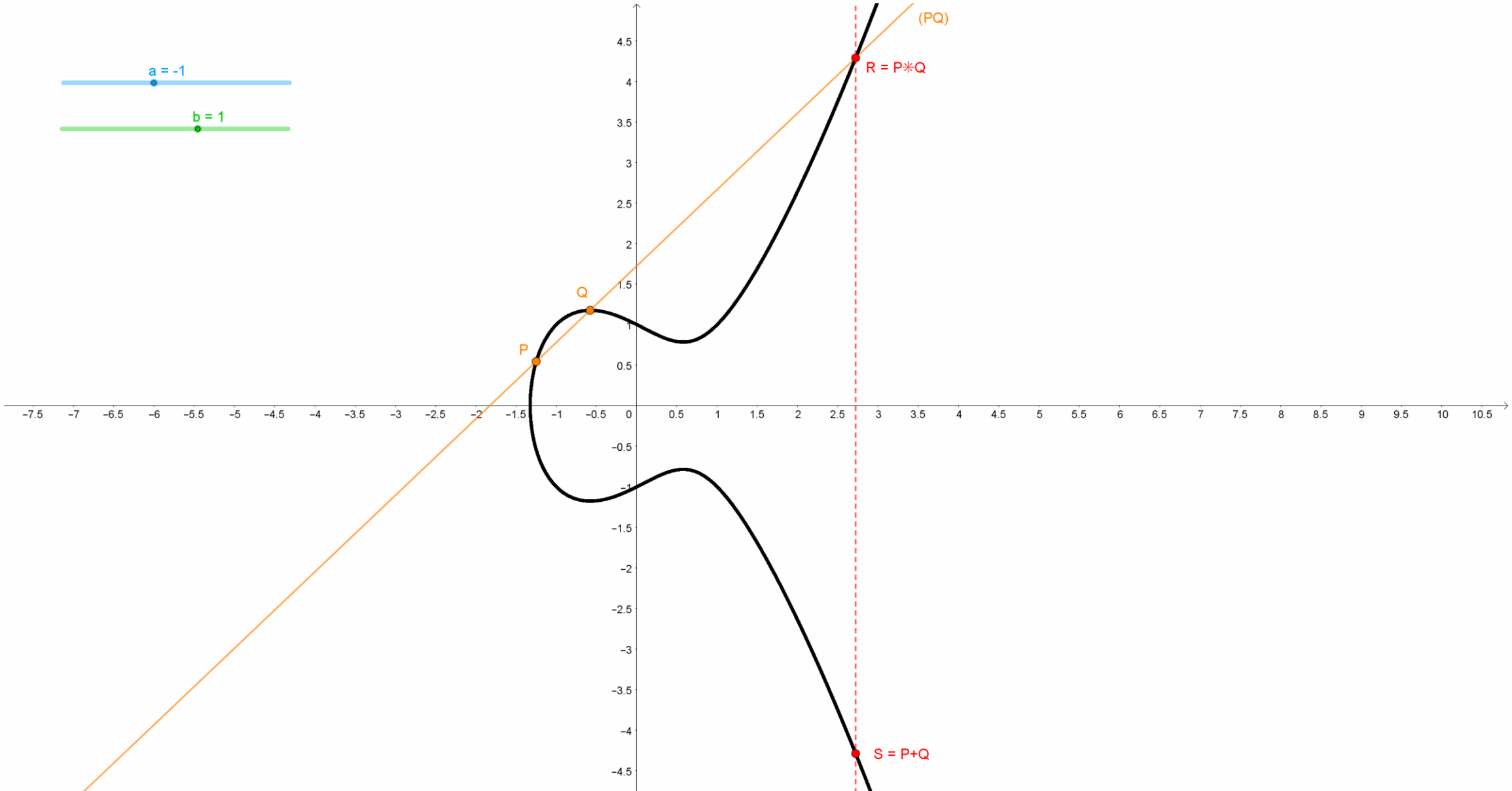
# Elliptic Curve: **Addition (P+Q)**

$$y^2 = x^3 + ax + b$$



# Elliptic Curve: **Addition P+P?**

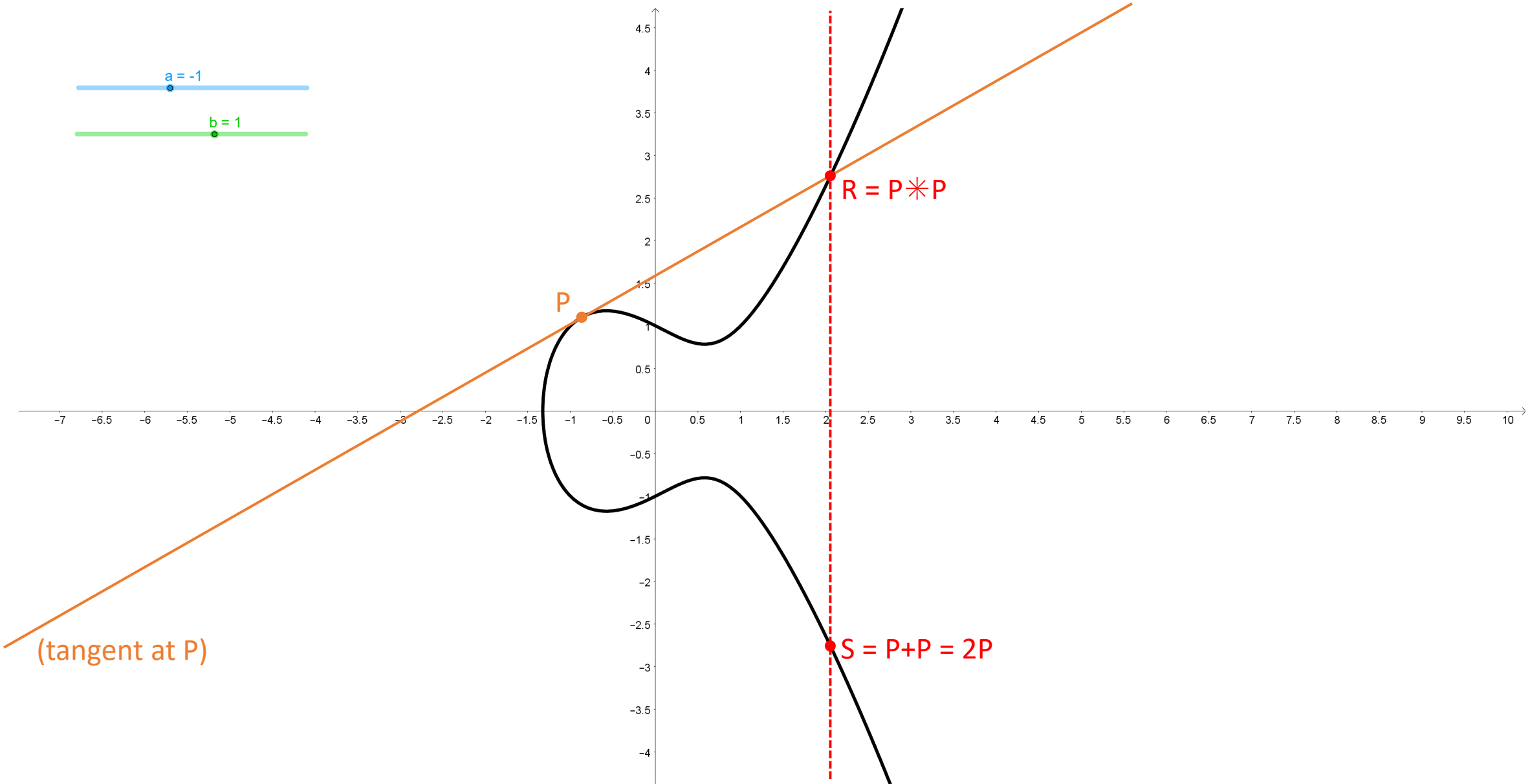
$$y^2 = x^3 + ax + b$$



tangent

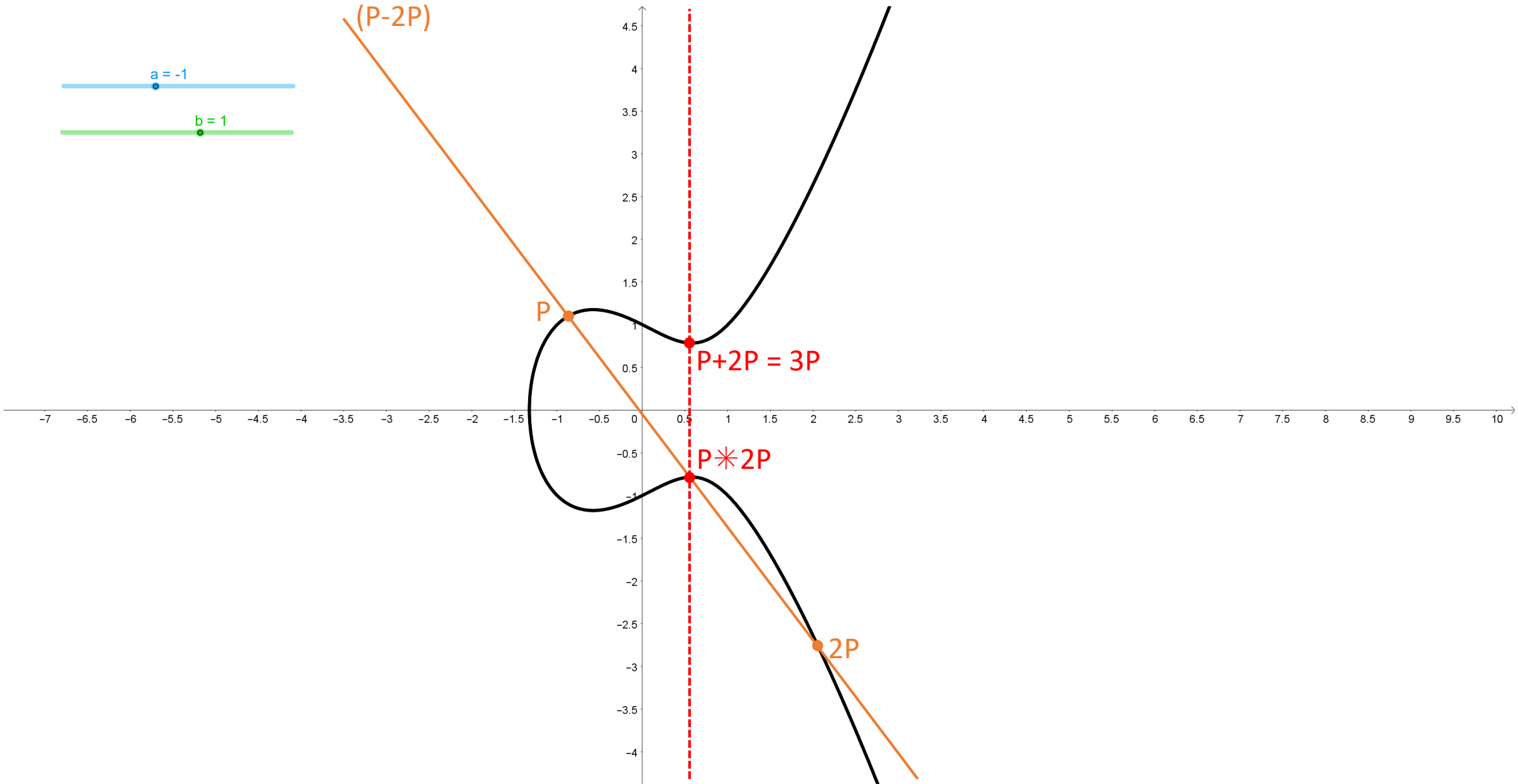
# Elliptic Curve: **Double (2P)**

$$y^2 = x^3 + ax + b$$



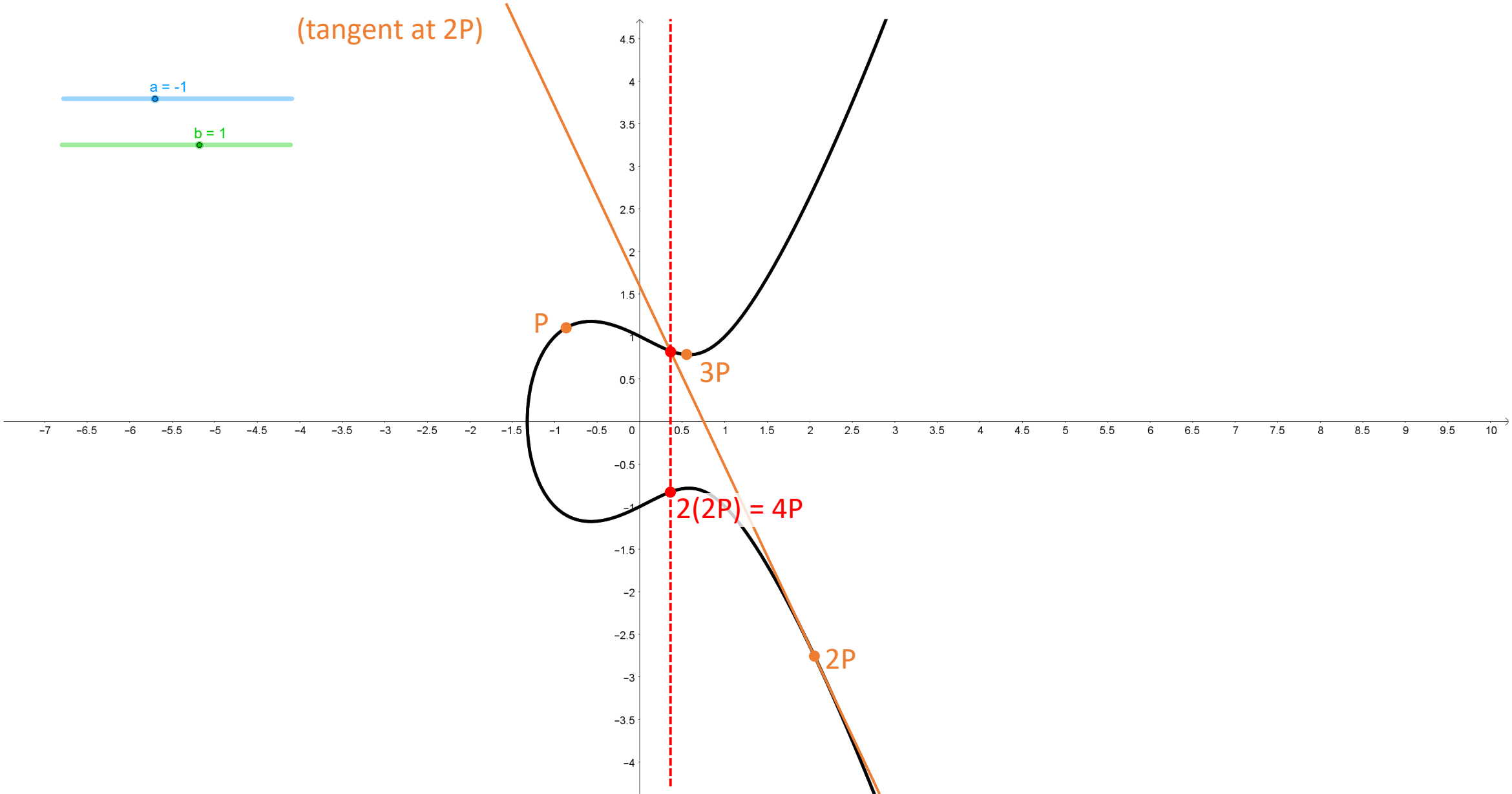
# Elliptic Curve: 3P = P+2P

$$y^2 = x^3 + ax + b$$



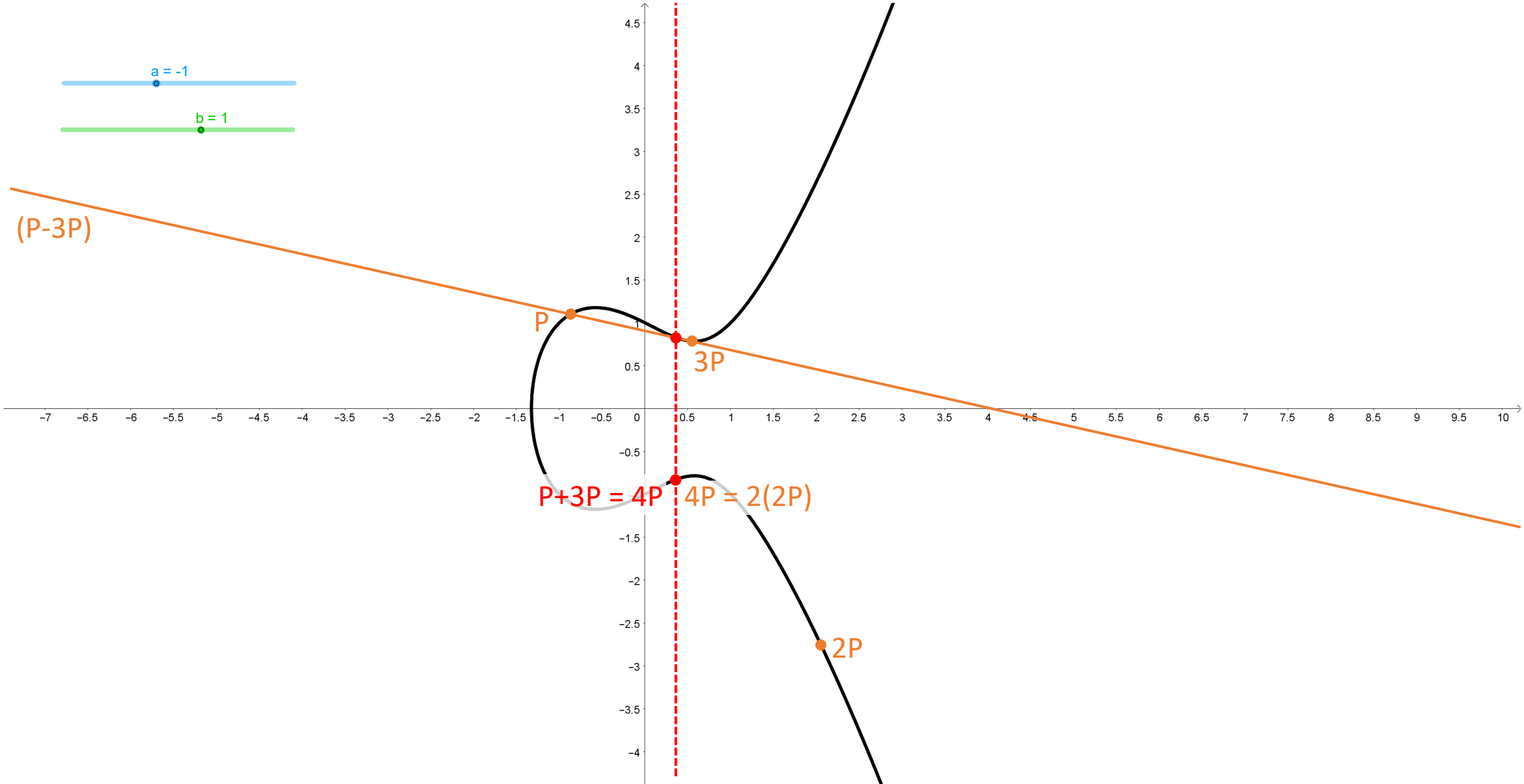
# Elliptic Curve: 4P = 2(2P) or P+3P?

$$y^2 = x^3 + ax + b$$



# Elliptic Curve: 4P = 2(2P) or P+3P?

$$y^2 = x^3 + ax + b$$



$R$
$T$

$$S = nR = kQ = nkP$$

$P$
$Q$

$* n$

$* k$

$R = kP$

$Q = nP$

$* k$

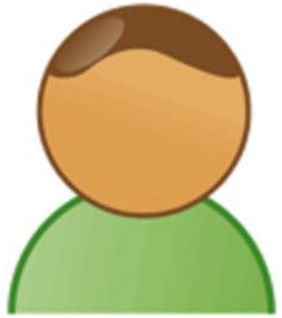
$* n$

$P$

$C: y^2 = x^3 + ax + b$

$M \in C$ (message)
$k \in \mathbb{N}$
$R = kP$
$S = kQ$

$T = S + M$



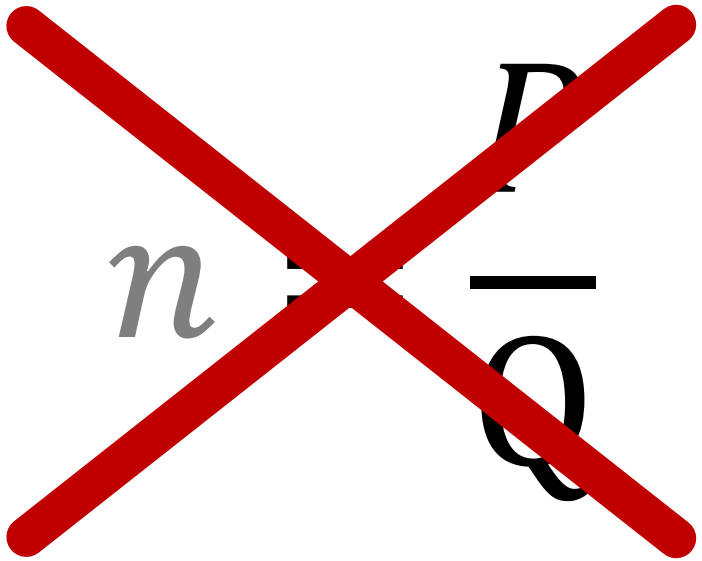
$n \in \mathbb{N}, P \in C$
$Q = nP$

$S = nR$

$M = T - S$

# Division on Elliptic Curve

$$nP = Q$$



# Principals conclusions

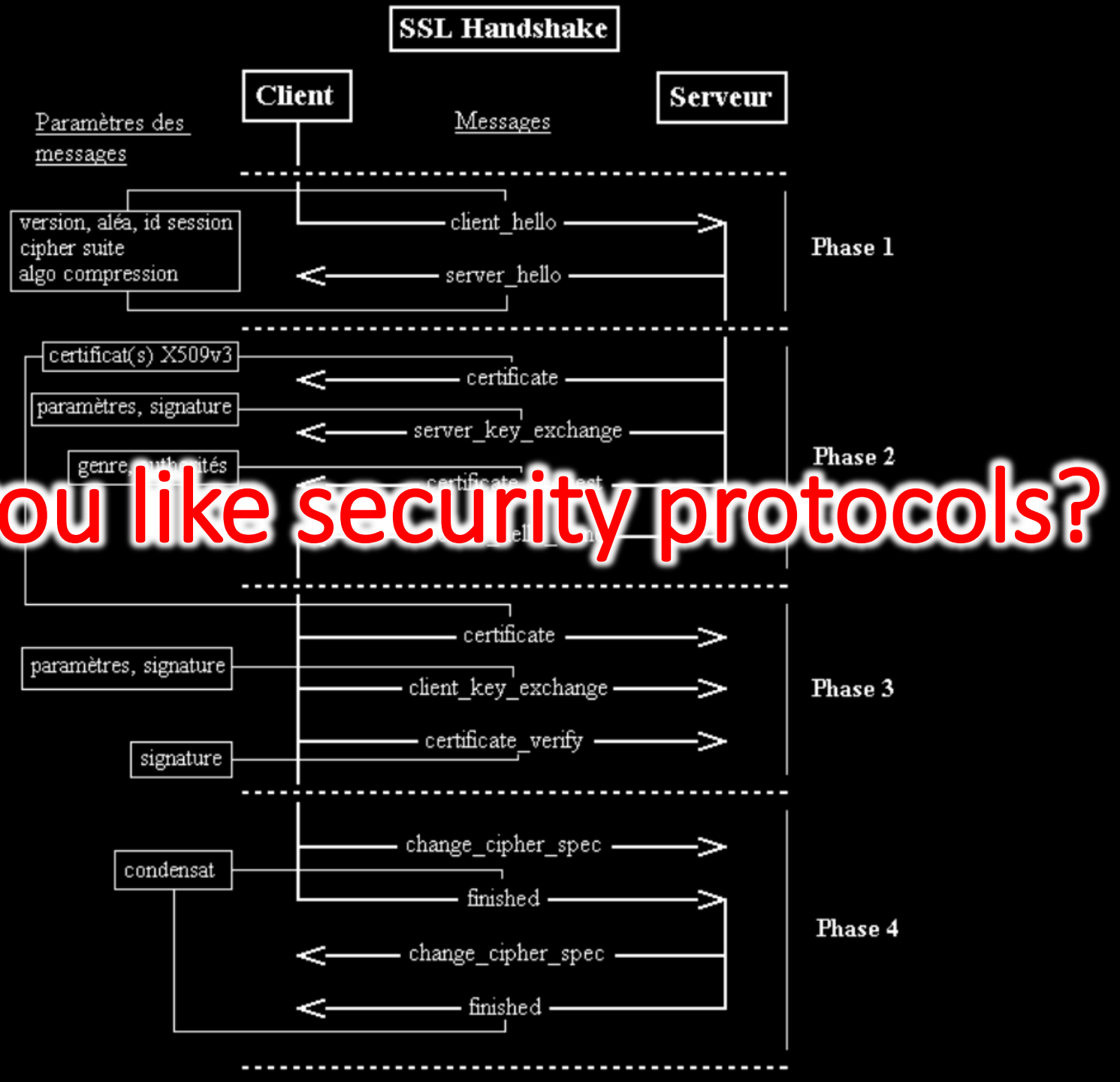
- The spy is blocked by the fact that he can't compute $\boldsymbol{S}$ (using only public information) nor can he find $n$ or $k$ (division on elliptic curves).
- **Pseudo-symmetric** encryption

**« + »:**

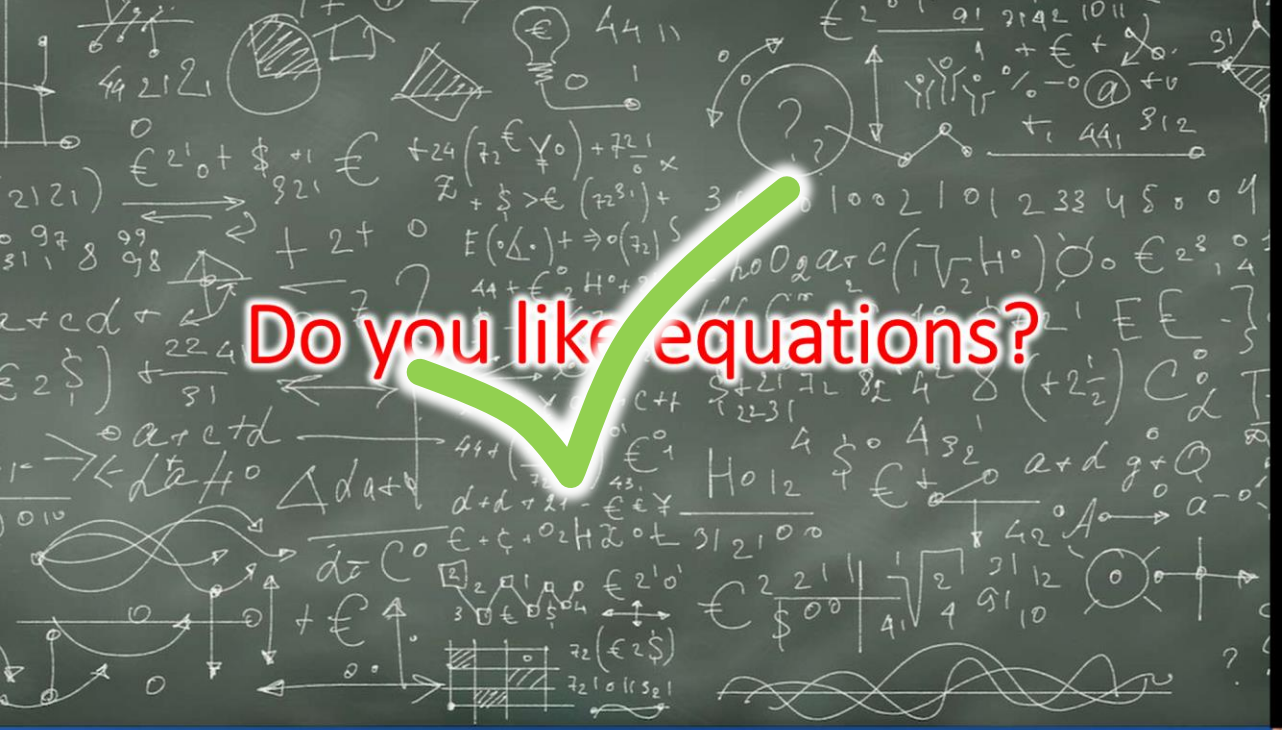
- **No** need of a **private key**

**« - »:**

- **Expensive** in terms of computations
- (Very) **Difficult to hack** via **brute force** method
- Subject to **quantum computers attacks**




# Do you like security protocols?


Do you like equations?


SSL Handshake
Client
Serveur
Paramètres des messages
Messages
version, aléa, id session
cipher suite
algo compression
client_hello
server_hello
Phase 1
certificat(s) X509v3
paramètres, signature
certificate
server_key_exchange
Phase 2
paramètres, signature
certificate_verify
signature
Phase 3
condensat
change_cipher_spec
finished
change_cipher_spec
finished
Phase 4
Do you like security protocols?


Do you like Toulouse?


Do you like money?

# Fermat… was from Toulouse!

« Fermat's theorem »:

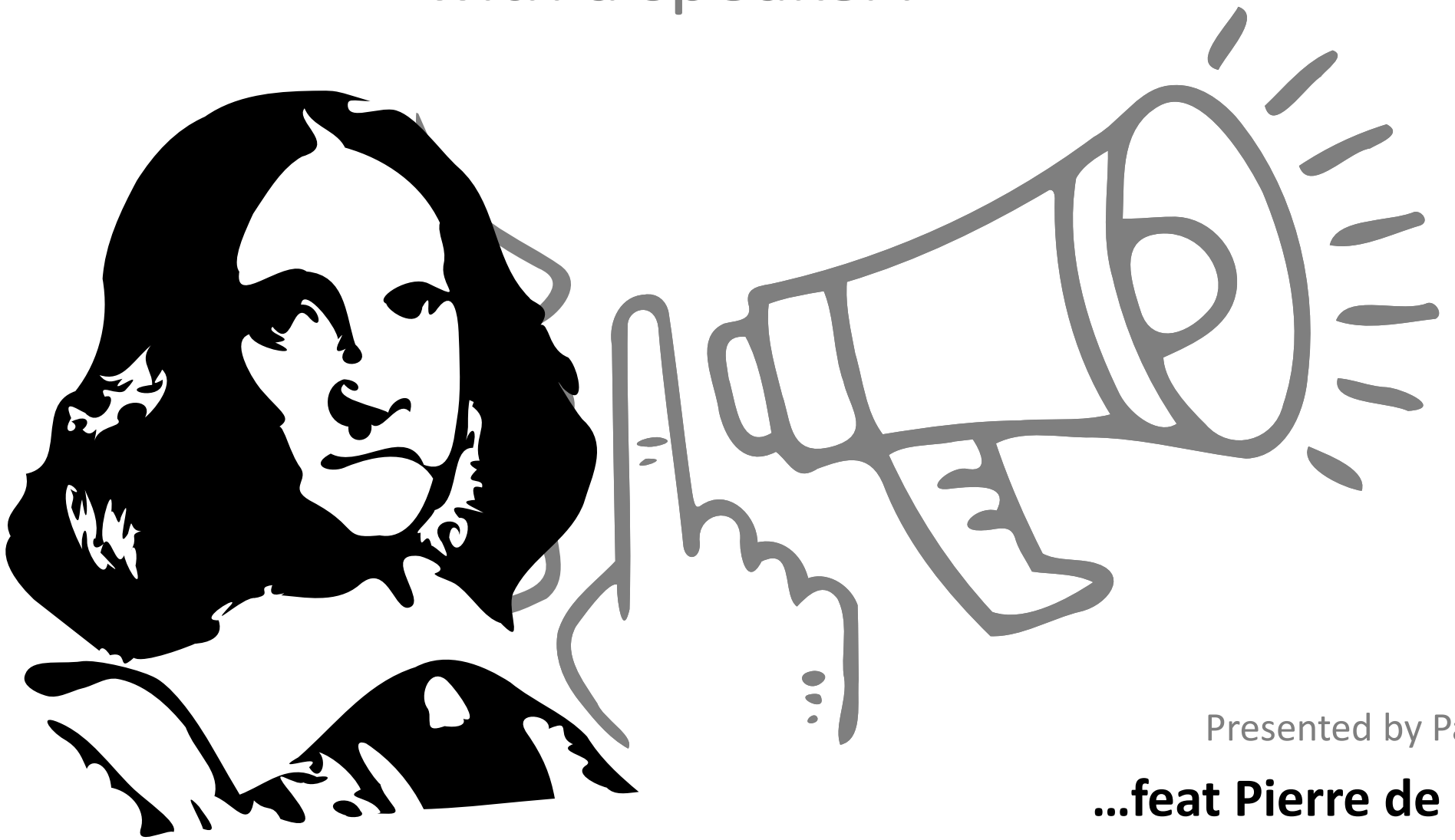
$$a^{p-1} \equiv 1\ [p]$$

« Fermat's last thorem »:

$$x^n + y^n = z^n$$

*Pierre de Fermat*

# ASYMETRIC CRYPTOGRAPHY:
# Or how to send a secret message with a speaker?



Presented by Paul DUBOIS

**…feat Pierre de FERMAT**