View online

Download PDF



Maths or magic?
End-to-end encryption explained with art

Paolo Insogna

Node.js TSC, Principal Engineer @ **Platformatic**

Encrypting is like painting!



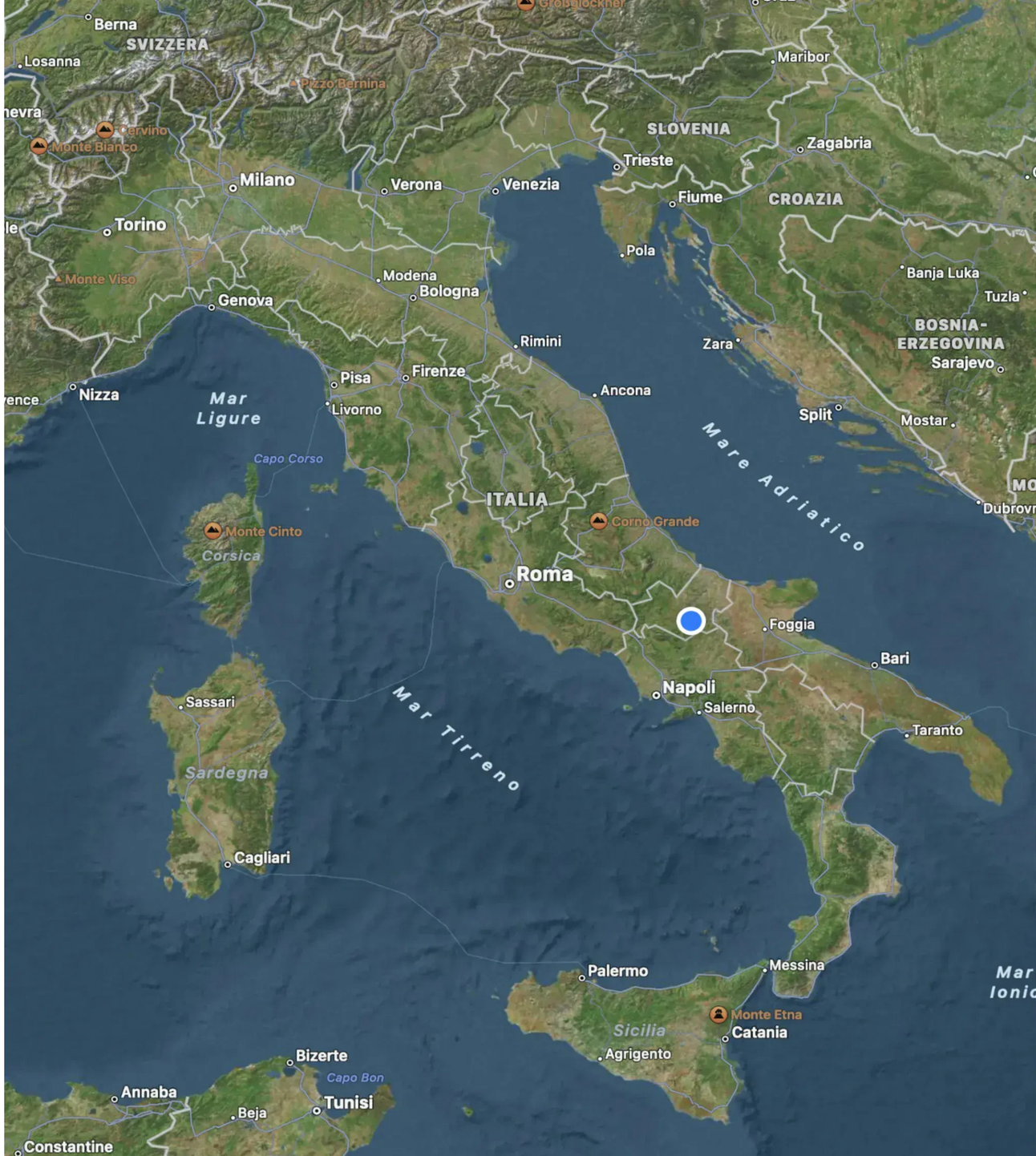
Hello, I'm **Paolo**!



Node.js Technical Steering Committee Member

Platformatic Principal Engineer

`paoloinsogna.dev` `ShogunPanda` `p_insogna` `pinsogna`



First of all, let's give credits!

This talk has originally been written by my colleague and friend **Michele Riva**.

Whatever goes wrong today, please complain directly to him on Twitter!

@MicheleRivaCode



Why do we need end-to-end encryption?



Network communications are complicated

To reach a distant peer, many compromisable network devices and links are used.



Wireless networks can be extremely insecure

Misconfigured networks allow an attacker to listen to all the traffic.



LAN connectors can be hacked

Do you believe in conspiracies?

Let's send a message ...



That was not an encrypted channel!



What is encryption anyway?

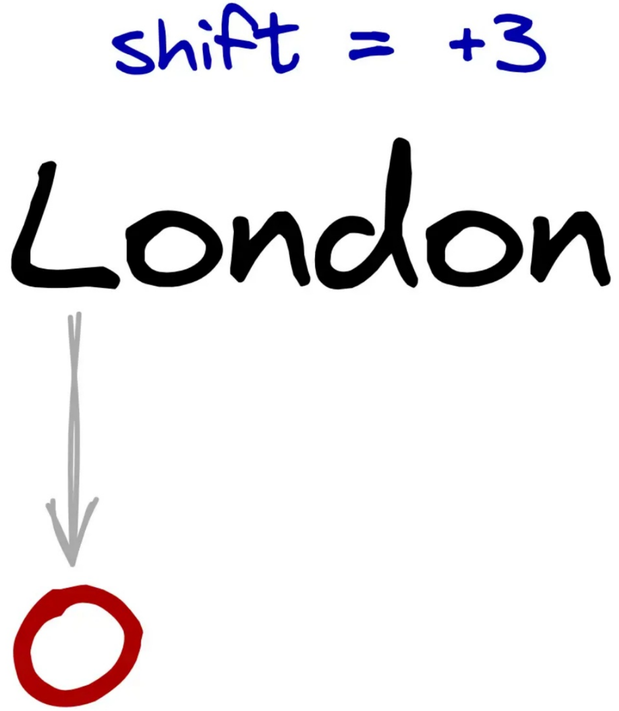
“The act of putting information into a special code, especially in order to prevent people from looking at it without authority”

The Oxford Dictionary

An example:
The Caesar Cipher



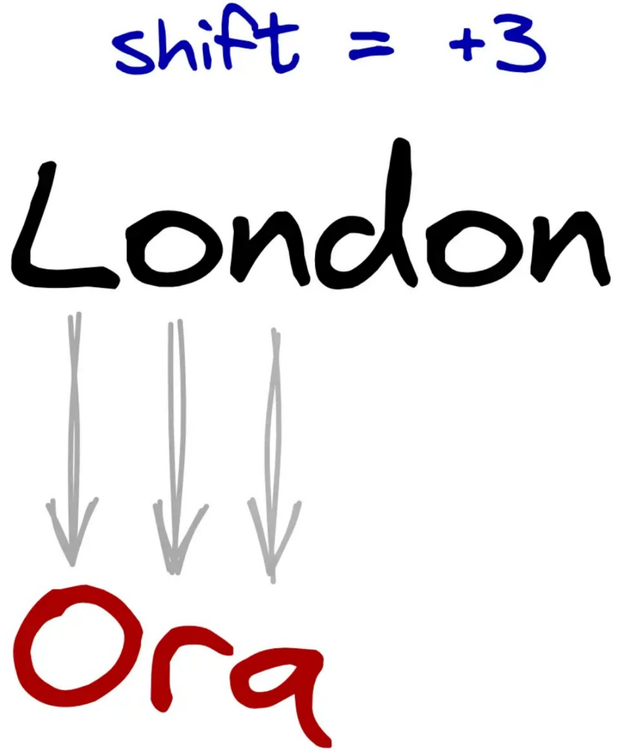
How does the Caesar Cipher work?



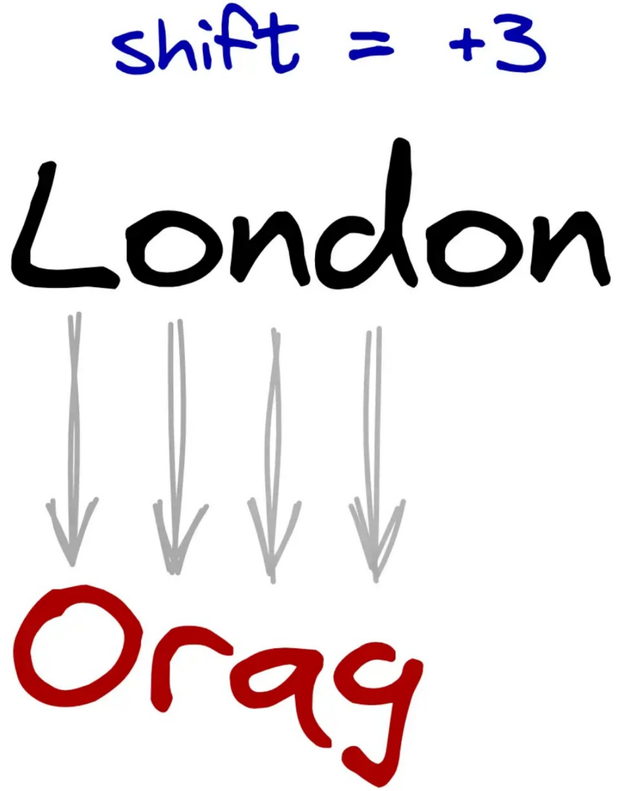
How does the Caesar Cipher work?



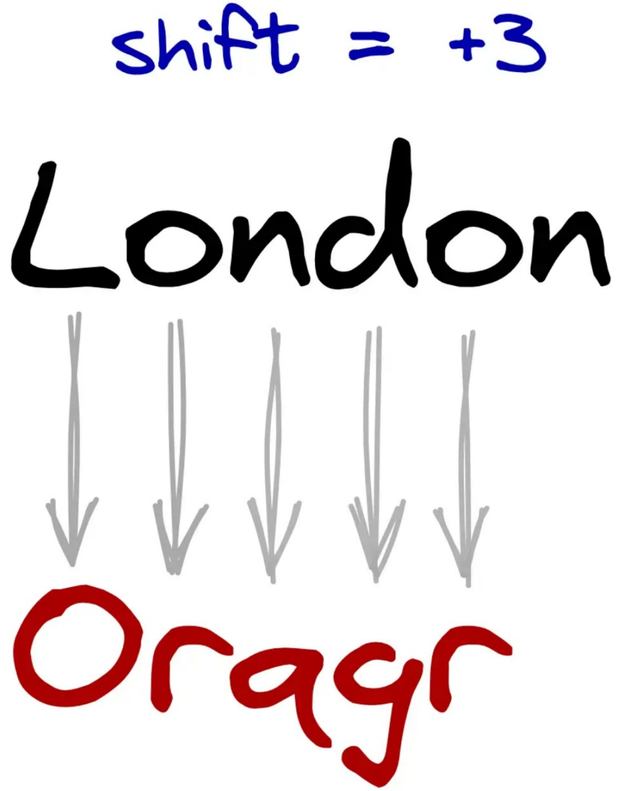
How does the Caesar Cipher work?



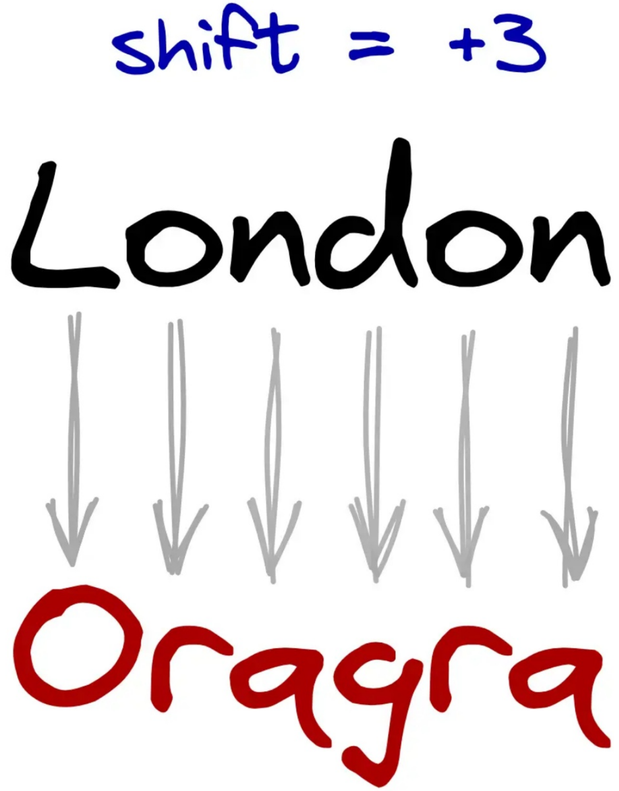
How does the Caesar Cipher work?



How does the Caesar Cipher work?



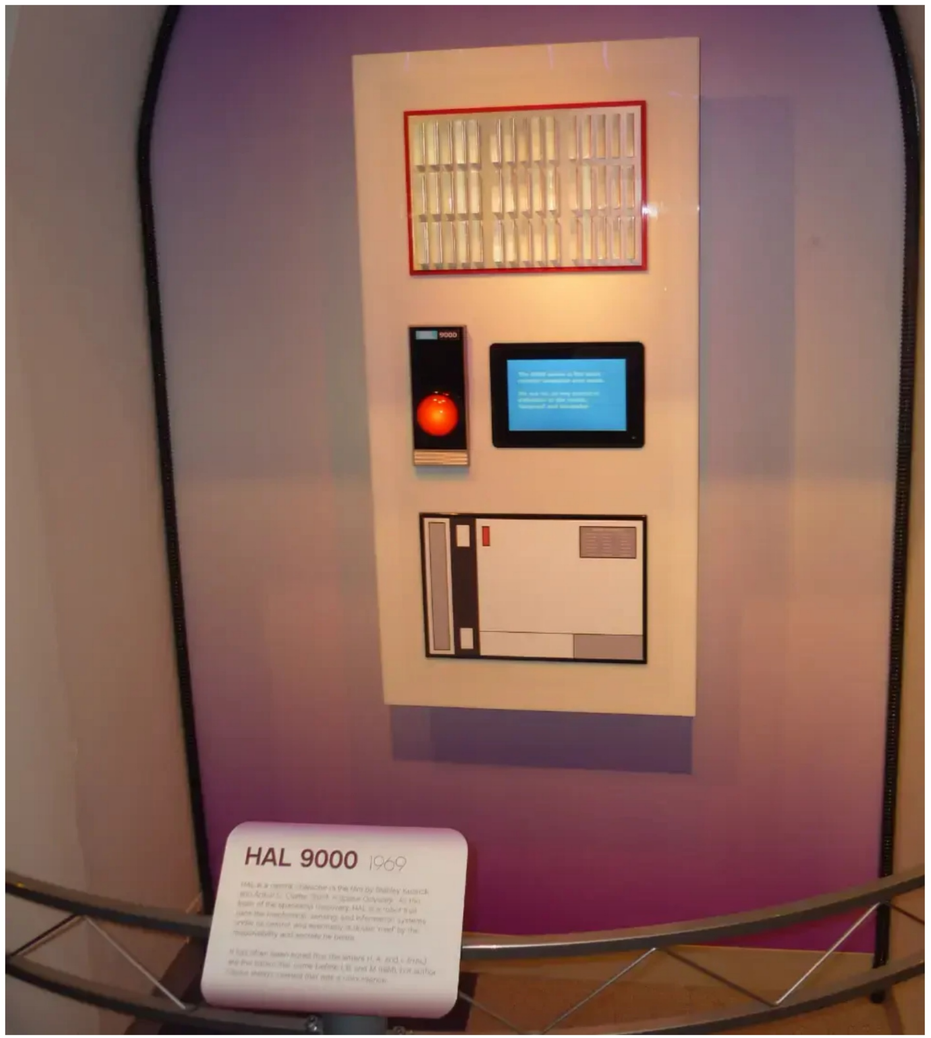
How does the Caesar Cipher work?



Trivia question

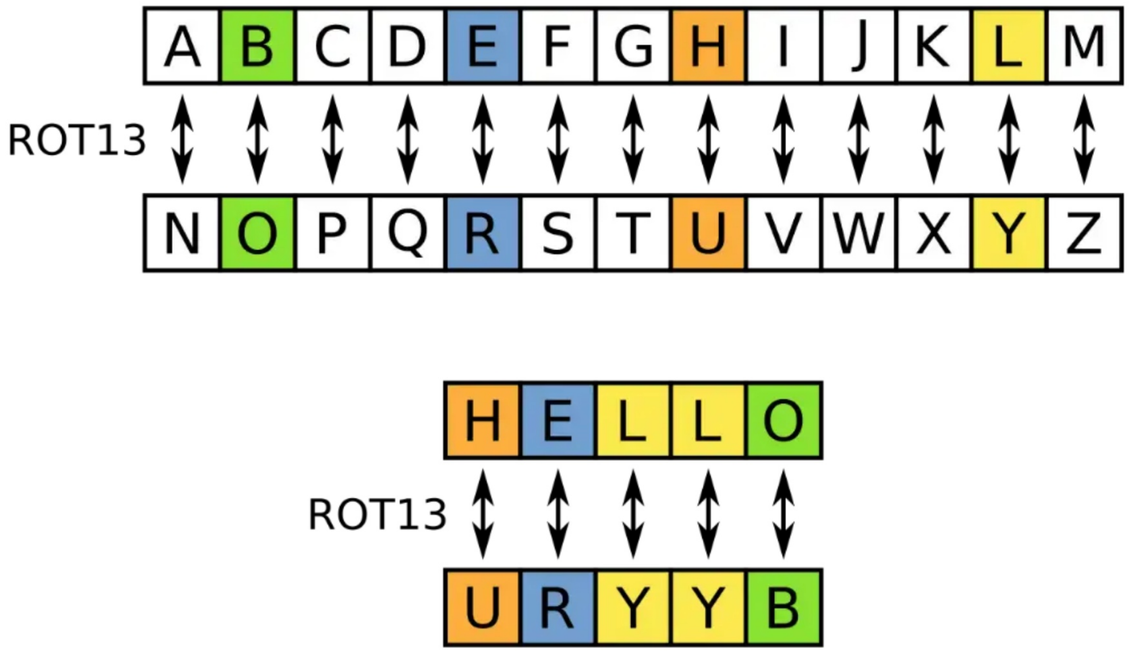
Where does its name come from?

It's a Caesar Cipher!



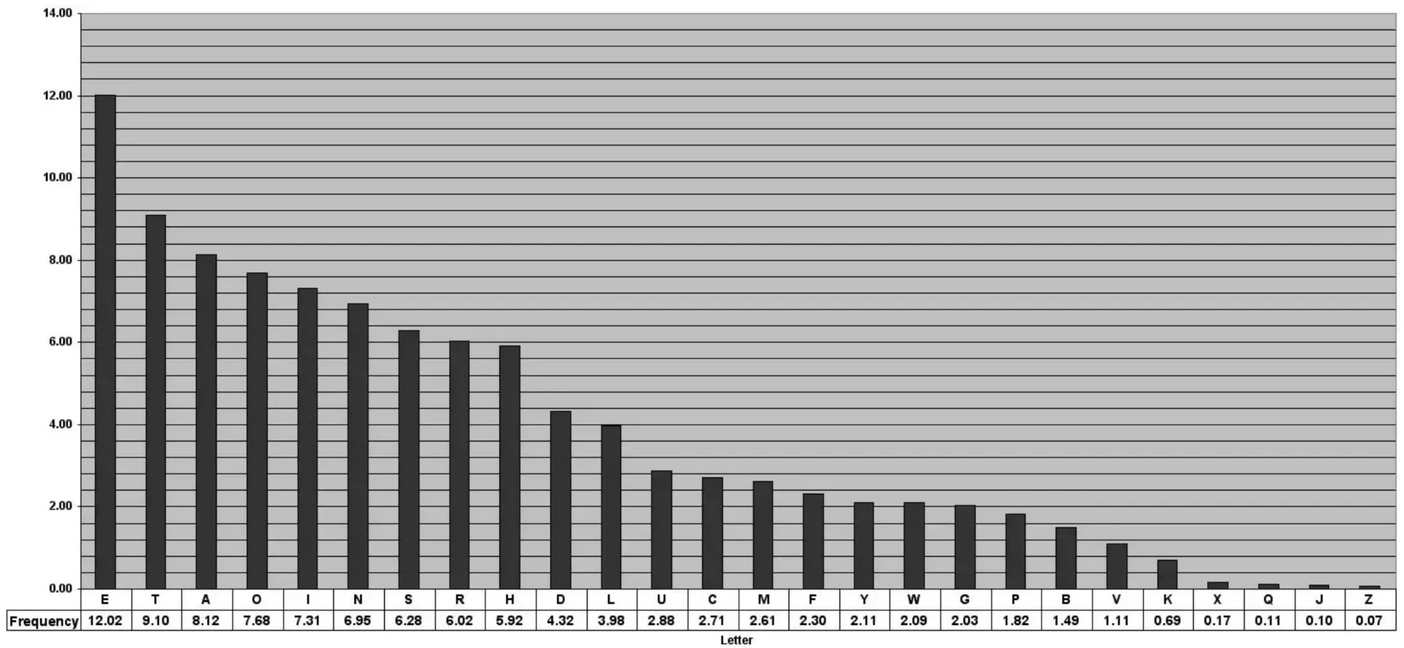
And now, a bit of pain for some of us

How old are you? Do you recognize this?



Why is Caesar Cipher weak?

Since only a few permutations are possible, brute forcing is quite easy (especially if using letter frequency analysis).

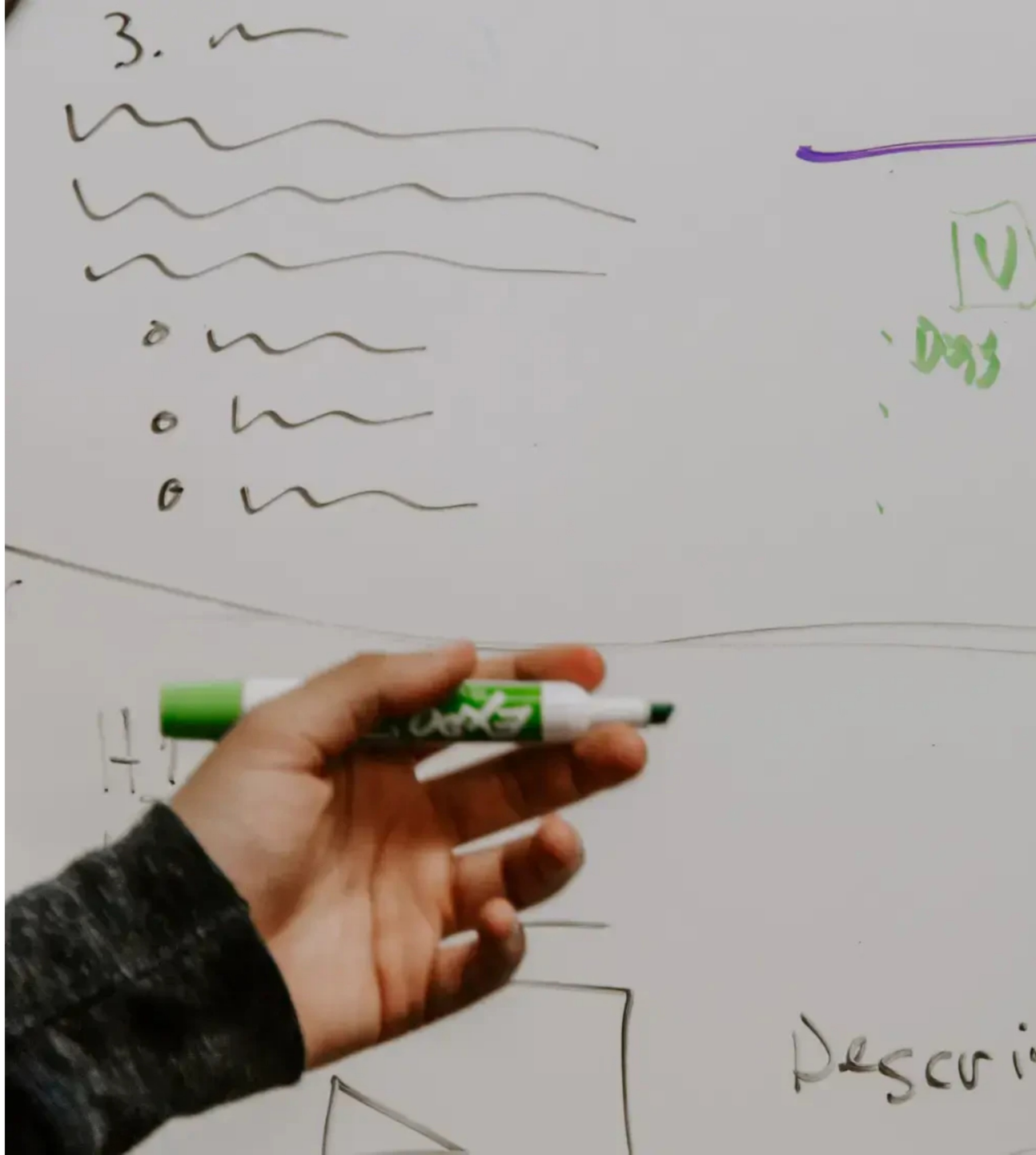


	E	T	A	O	I	N	S	R	H	D	L	U	C	M	F	Y	W	G	P	B	V	K	X	Q	J	Z
Frequency	12.02	9.10	8.12	7.68	7.31	6.95	6.28	6.02	5.92	4.32	3.98	2.88	2.71	2.61	2.30	2.11	2.09	2.03	1.82	1.49	1.11	0.69	0.17	0.11	0.10	0.07

A good algorithm

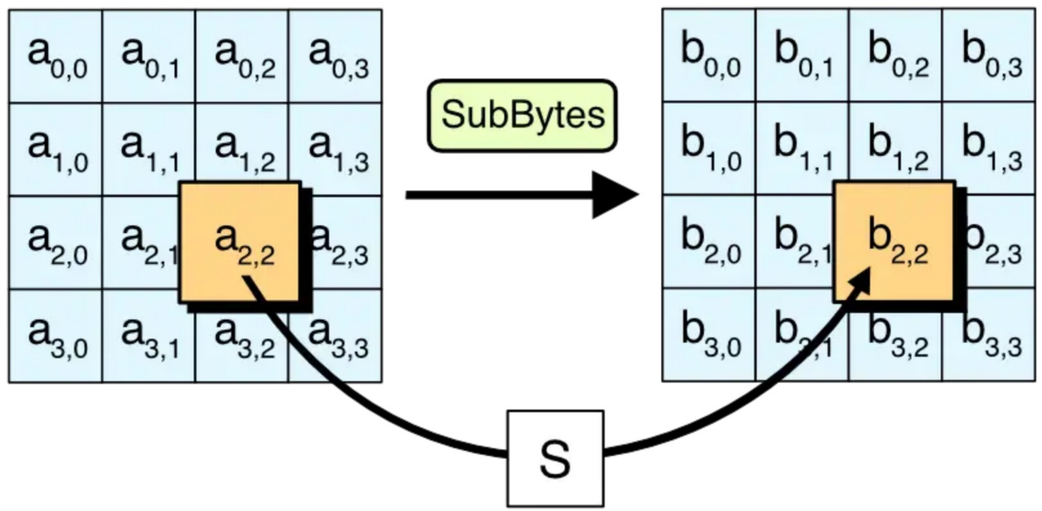
The algorithm must be public
Public algorithms can be continuously analyzed and validated from everyone.

The security is in the parameters
The parameters must be chosen so that brute forcing is unfeasible.



What are our choices today?

The **Advanced Encryption Standard** (AES) is a secure symmetric block algorithm.



Let's make the communication secure

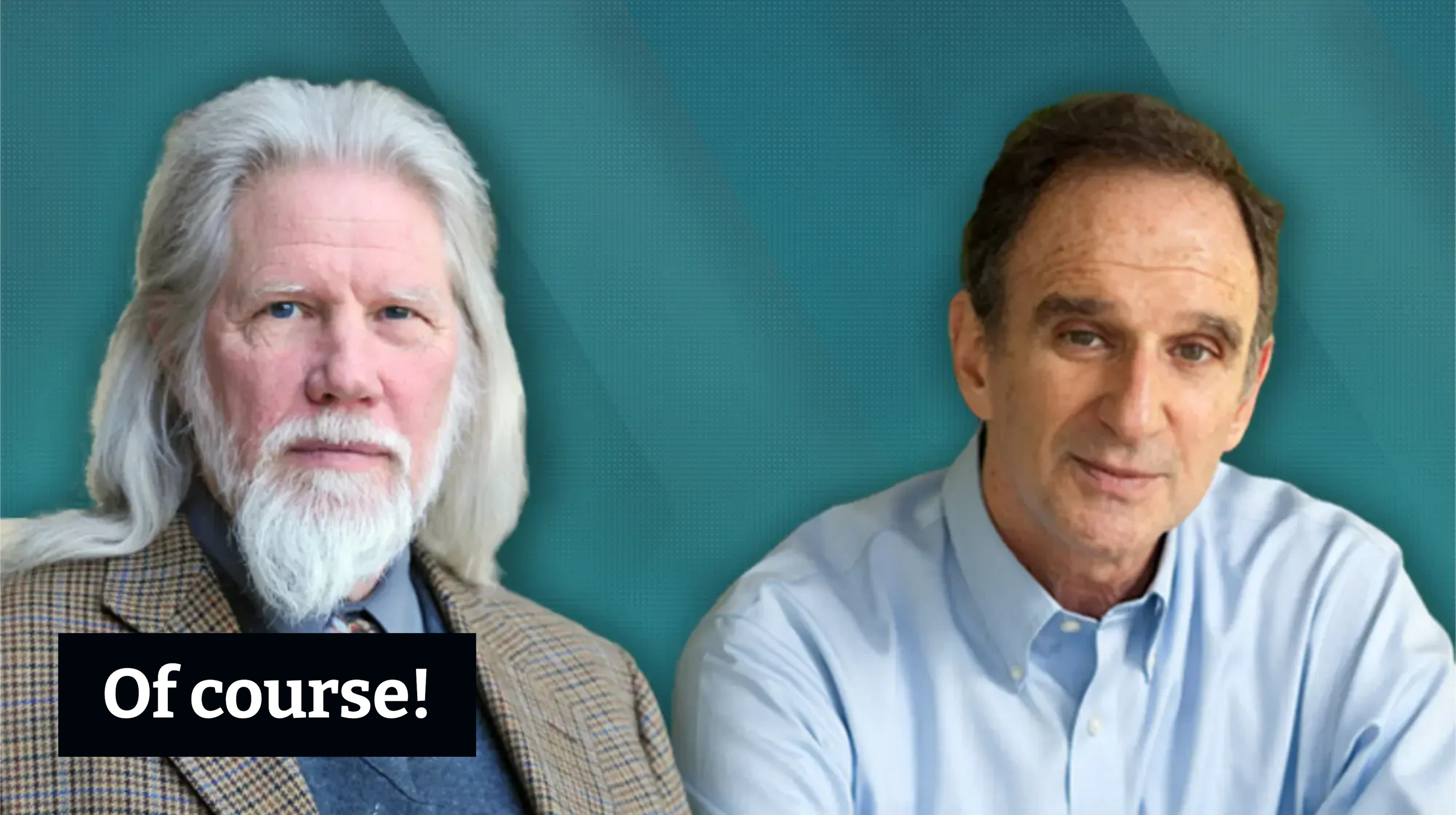
Before starting encrypting the traffic, the peers have to choose a shared key.

Shared key handshaking is not and cannot be encrypted.

Anybody could steal the key.

Do we have a solution?




Of course!

The Diffie-Hellman algorithm



Brings security in the insecurity

It allows to securely exchange cryptographic keys over a public channel.



Strong mathematical basis

It uses asymmetric keys to build a shared key.



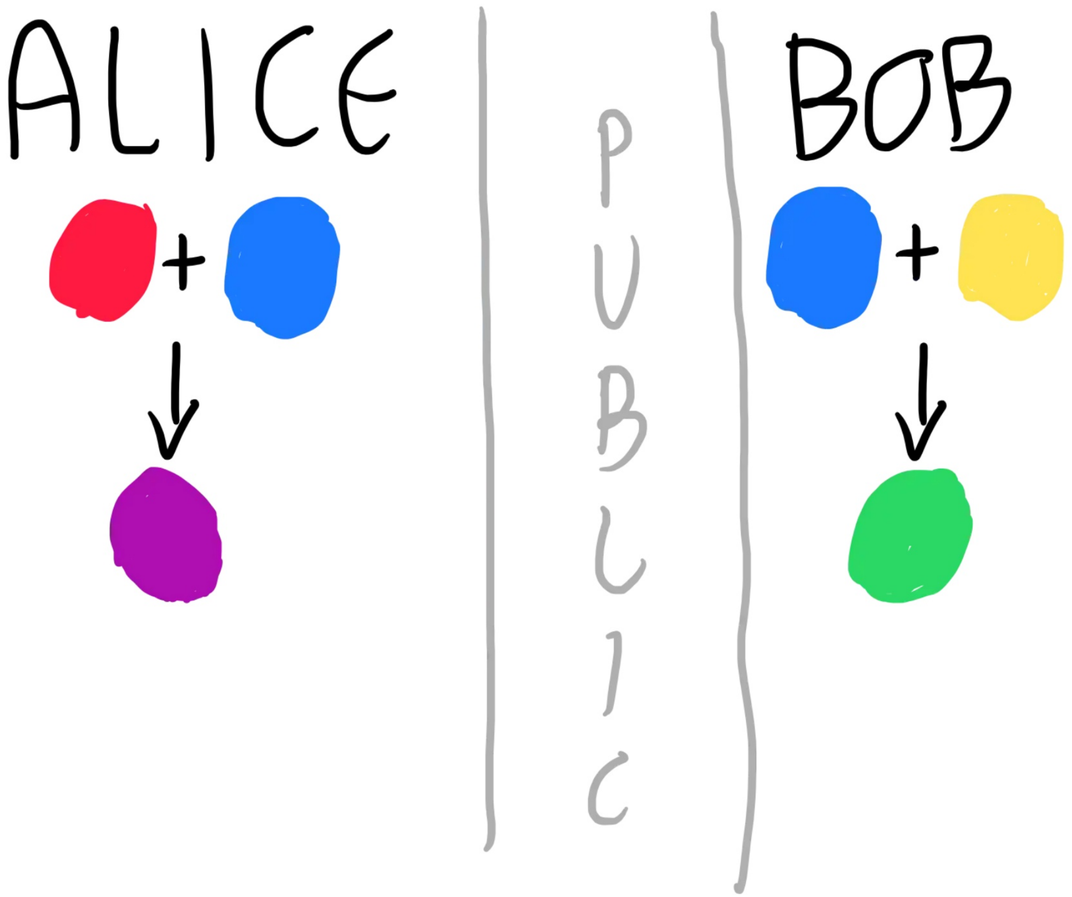
It is fast and easy to implement

Keys can (and should) be rotated with each message to improve security.

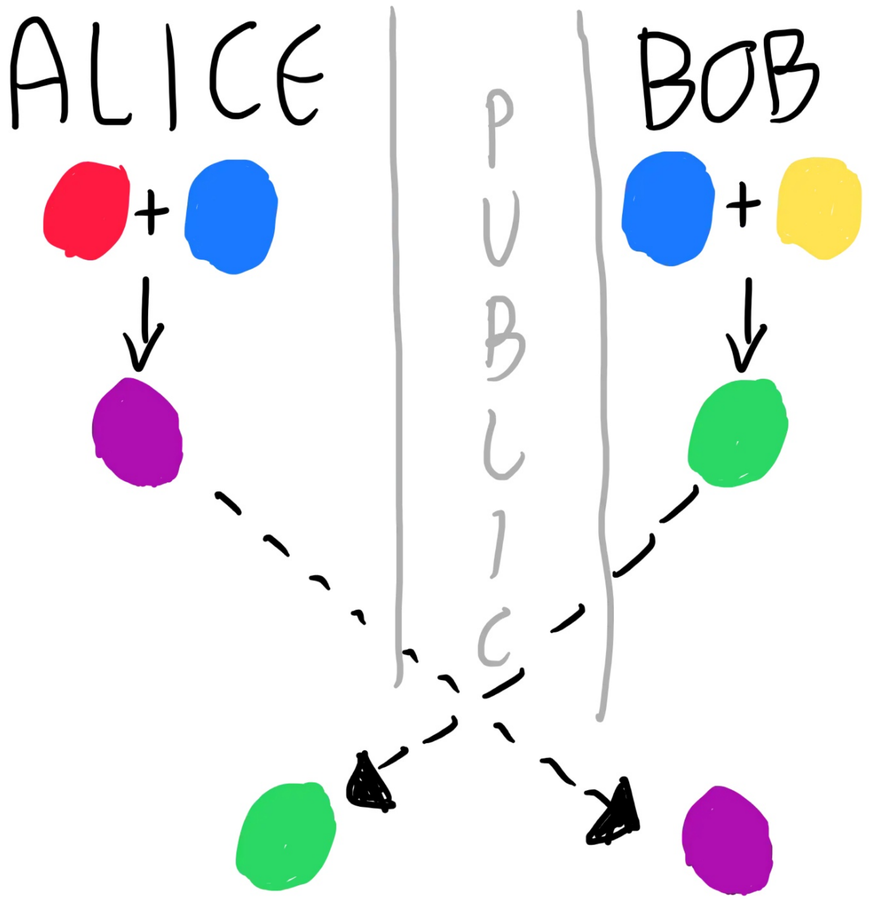
It's time to paint!



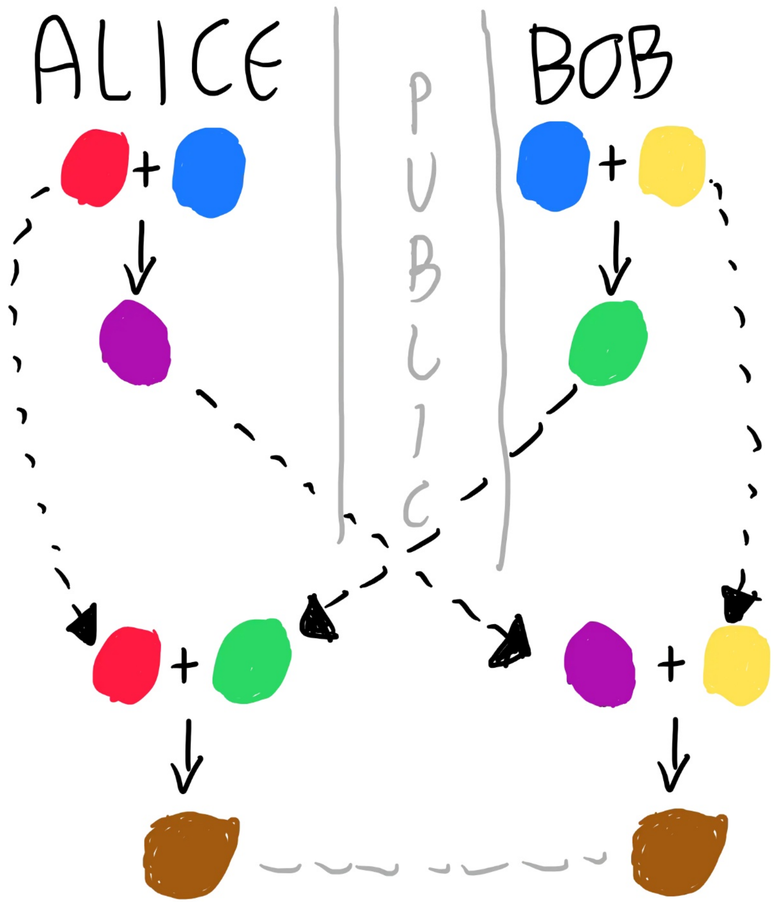
Generate the private and public keys



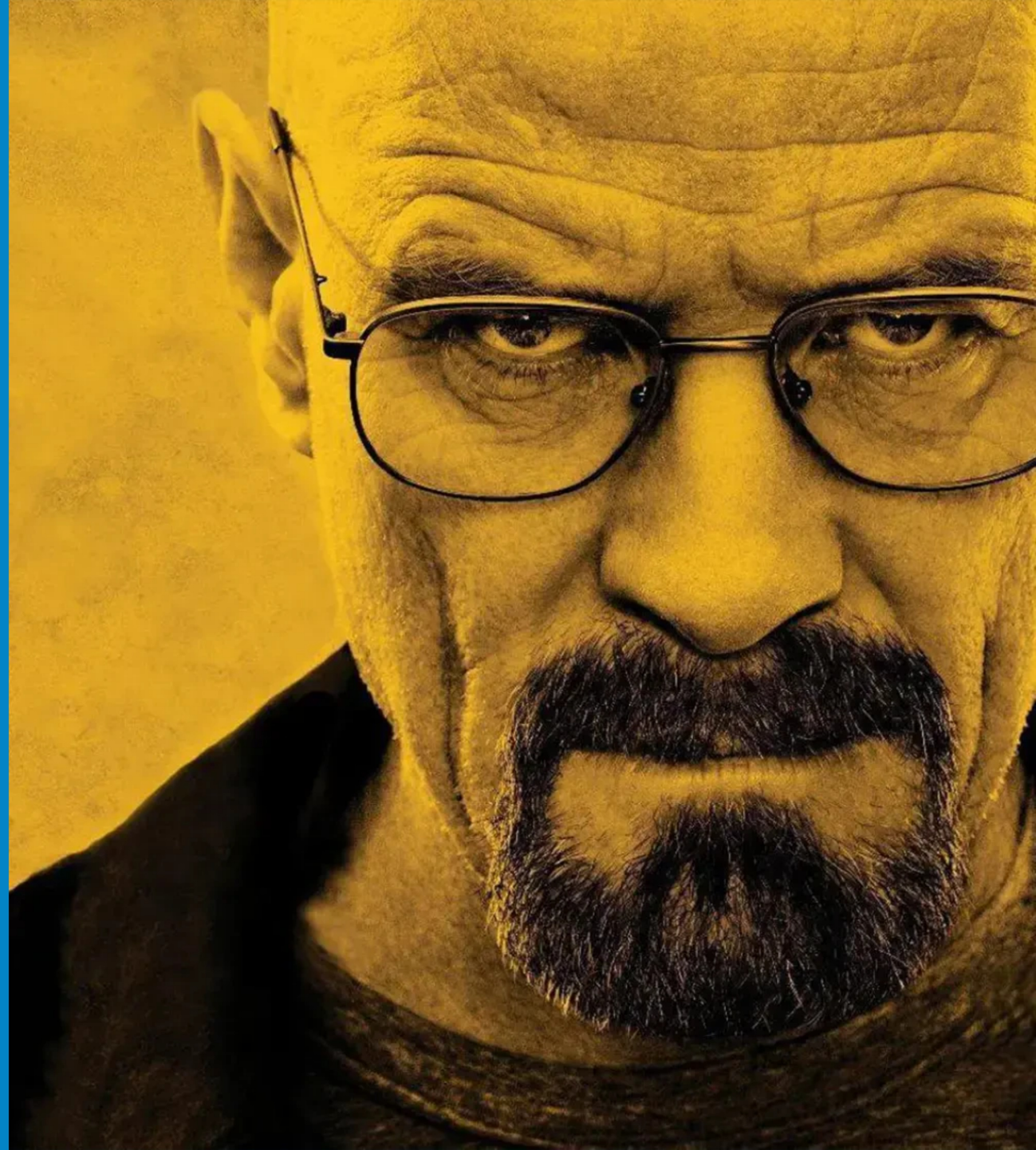
Exchange the public keys



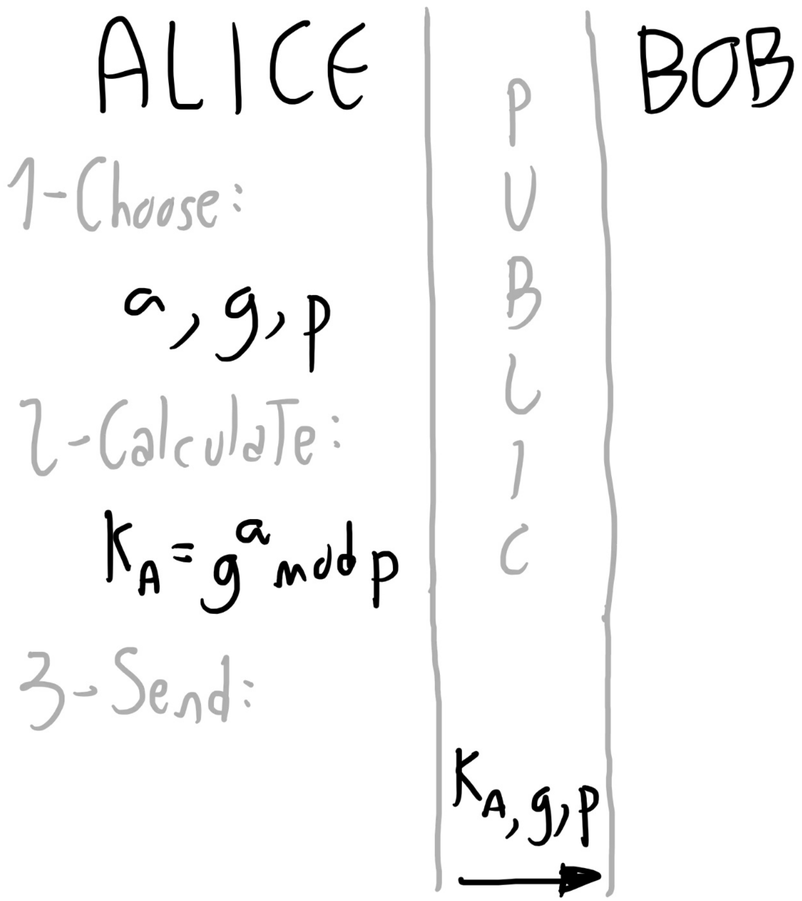
Generate the shared keys



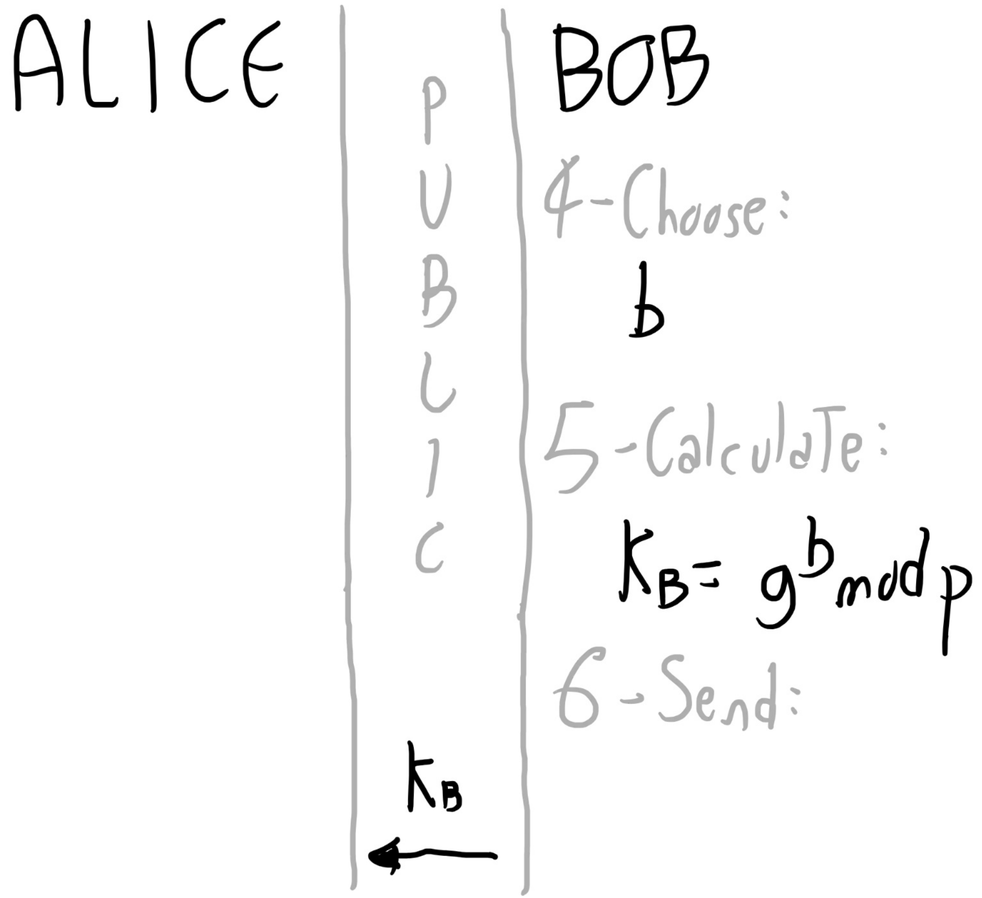
Let's make some math!



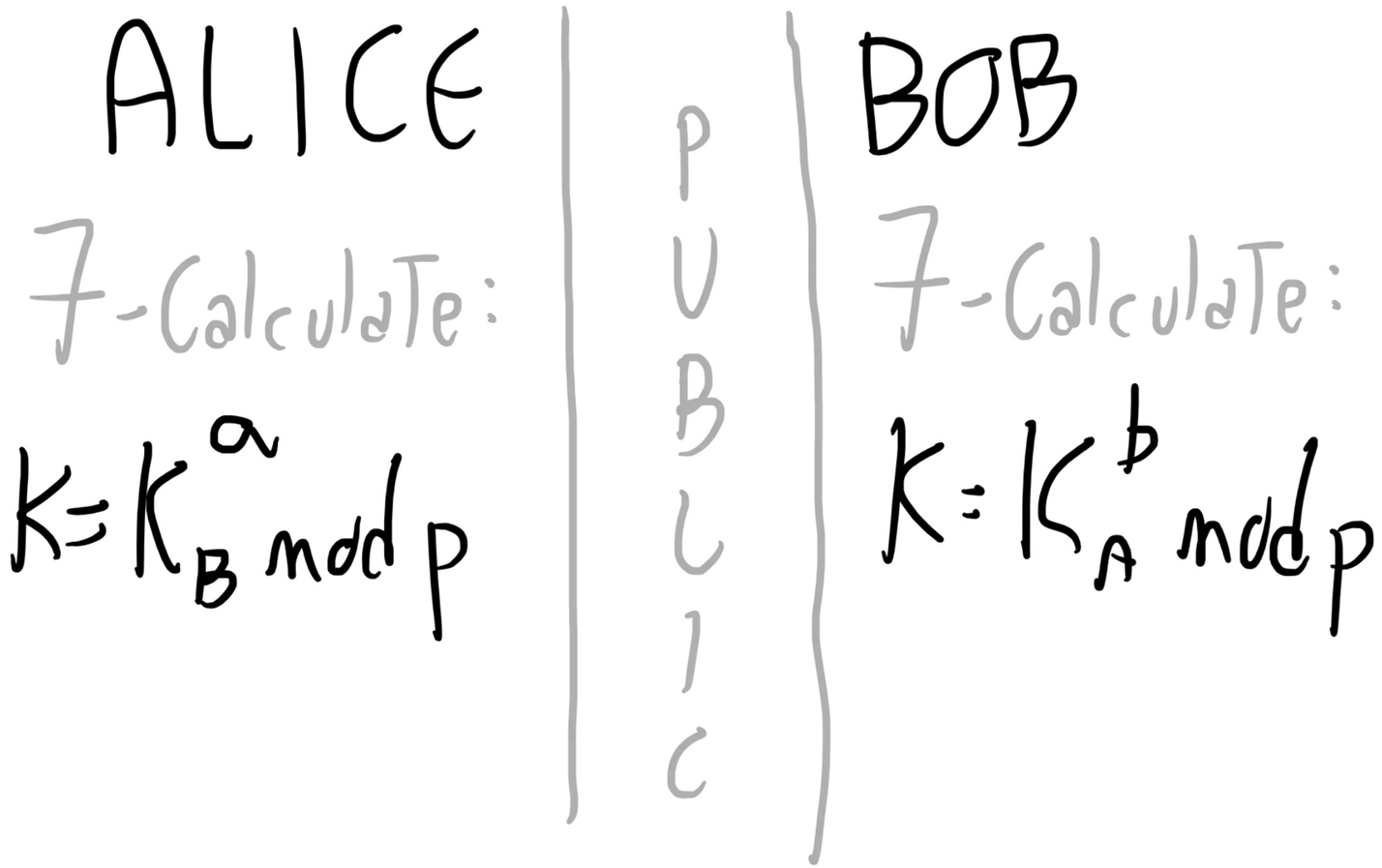
How is this possible?



How is this possible?



How is this possible?



How is this possible?

$$
\begin{aligned}
K &= K_A^b \bmod p = \\
&= \left(g^a \bmod p\right)^b \bmod p = \\
&= g^{ab} \bmod p = \\
&= g^{ba} \bmod p = \\
&= \left(g^b \bmod p\right)^a \bmod p = \\
&= K_B^a \bmod p
\end{aligned}
$$

Let's prove it!

$$a=5 \quad b=6 \quad g=2 \quad p=10$$

$$K_A = g^a \bmod p = 2^5 \bmod 10 = 32 \bmod 10 = 2$$

$$K_B = g^b \bmod p = 2^6 \bmod 10 = 64 \bmod 10 = 4$$

$$K = K_A^b \bmod p = 2^4 \bmod 10 = 16 \bmod 10 = 6$$

"

$$K = K_B^a \bmod p = 4^6 \bmod 10 = 4096 \bmod 10 = 6$$

"

One last thing™

“The decisions we make about communication security today will determine the kind of society we live in tomorrow.”

Whitfield Diffie



Thank you!

Paolo Insogna

Node.js TSC, Principal Engineer

@p_insogna

paolo.insogna@platformatic.dev

Platformatic