Elliptic Curve Cryptography

David Wolever
@wolever

what is a cryptography

@wolever

that’s easy

@wolever

133498471 9824

@wolever

133498471982⁴

⊕ hello, world!

@wolever

133498471982 4

⊕ hello, world!

blaiiaieuaoaf

@wolever

otp: one true ciPher

@wolever

otp: one time pad

@wolever

problem

numbers are hard

@wolever

~~numbers are hard~~

@wolever

sharing numbers is hard

@wolever

enter: public key cryptography

@wolever

pick a secret number

while you're listening

@wolever

dh: Diffie–Hellman

@wolever

$$A^b \% p = g^{ab}$$

@wolever

We agree on:

p: a *prime modulus* (ex: 23)

g: a *primitive root base* (ex, 5)

You (Alice) and I (bob)
Pick secret numbers:
a: Alice's number (ex, 32)
b: Bob's number (ex, 16)

We each calculate:

$$A = g^a \% p$$
$$A = 5^{32} \% 23$$

$$B = g^b \% p$$
$$B = 5^{16} \% 23$$

We share A and B
(that’s right, you can hear
them)

We each calculate:

$$s = A^b \% p$$

$$s = B^a \% p$$

and because *math*

@wolever

We get the same number:

$$
\begin{aligned}
s &= A^b \,\%\, p \\
&= (g^a \,\%\, p)^b \,\%\, p \\
&\equiv g^{ab} \,\%\, p \\
&\equiv (g^b \,\%\, p)^a \,\%\, p \\
&\equiv B^a \,\%\, p
\end{aligned}
$$

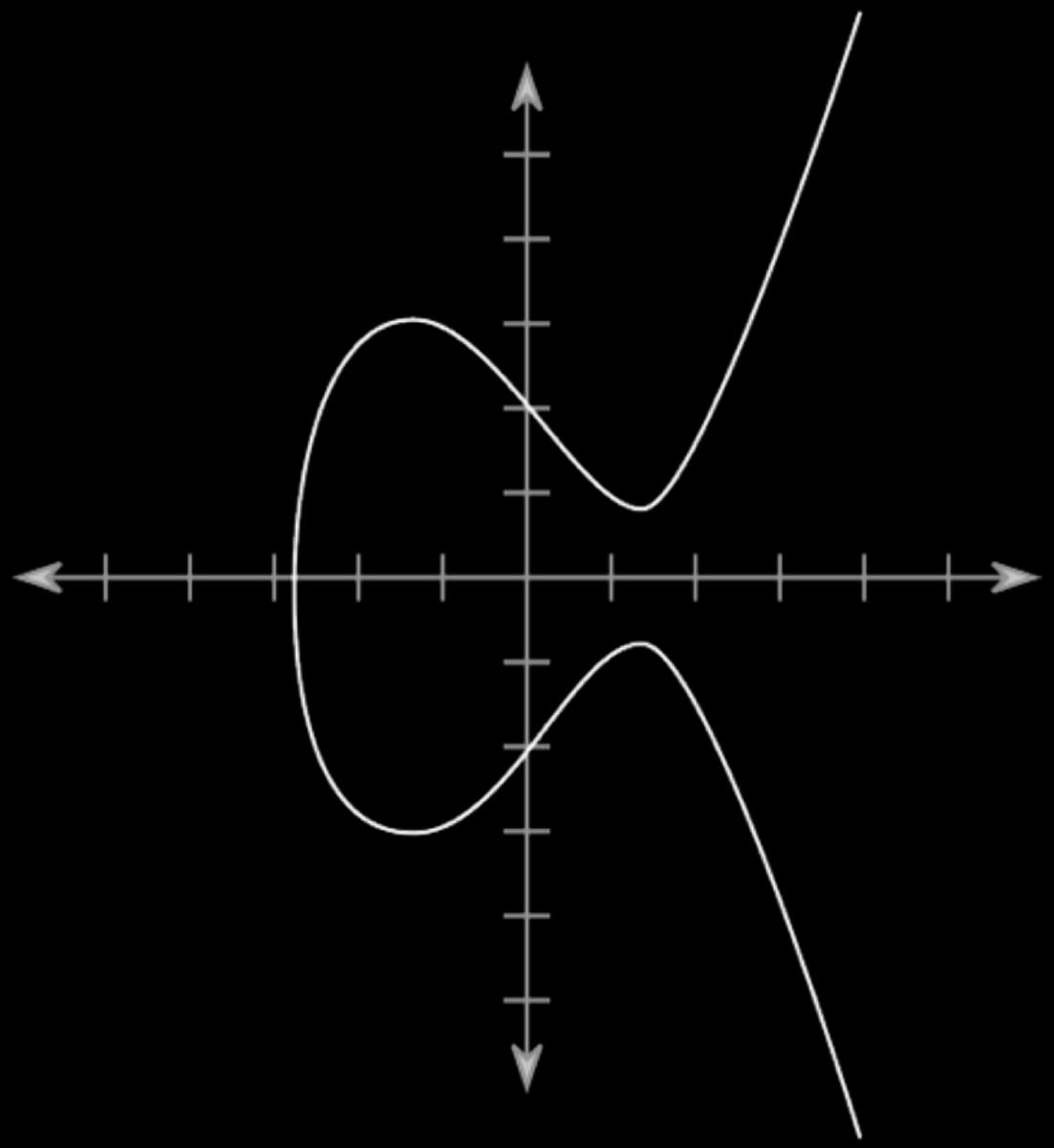
@wolever

magic

@wolever

enter: elliptic curves

@wolever



@wolever

300 bit ec key ≈

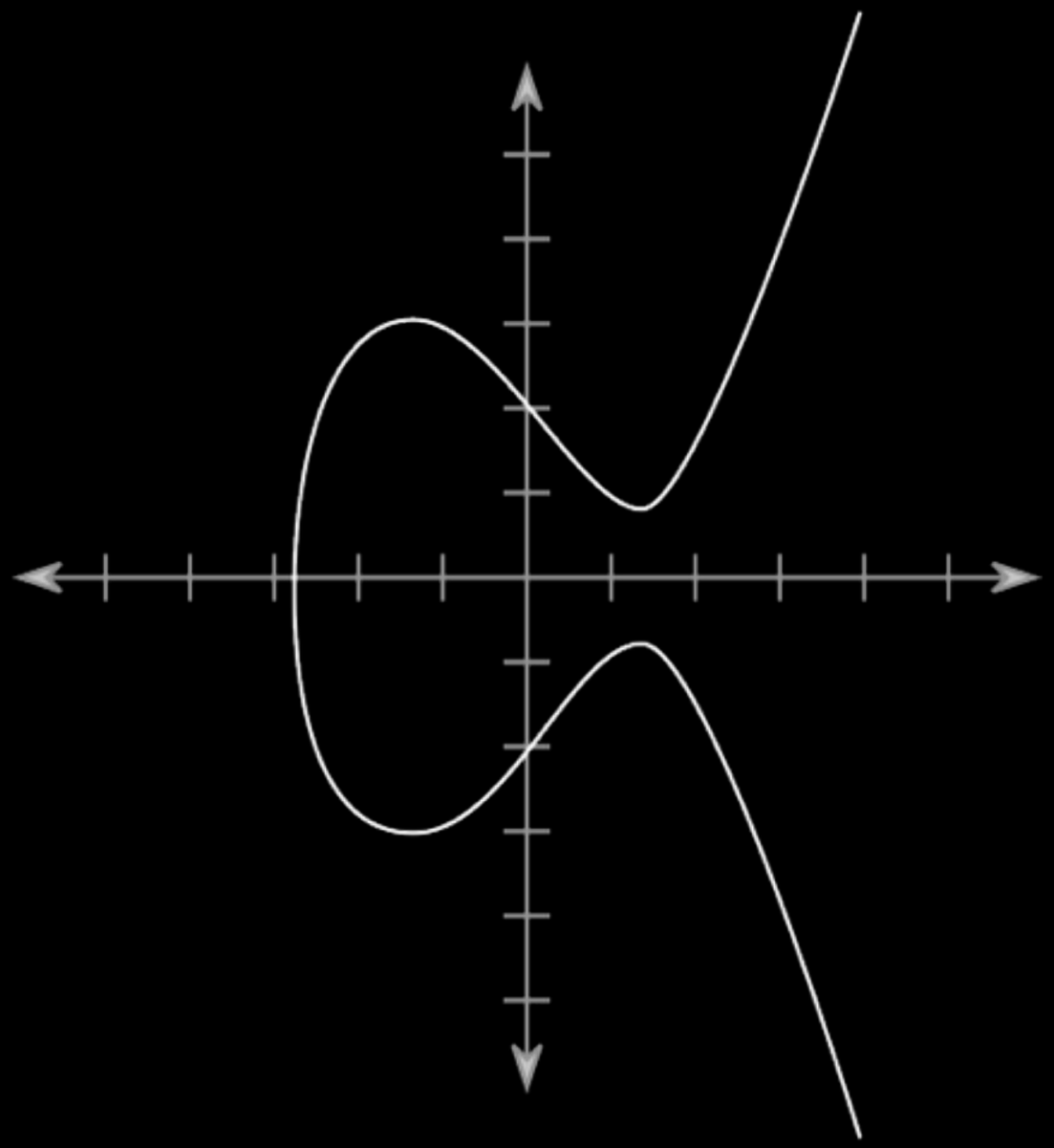
3000 bit dh key

@wolever

1. pick a curve

@wolever

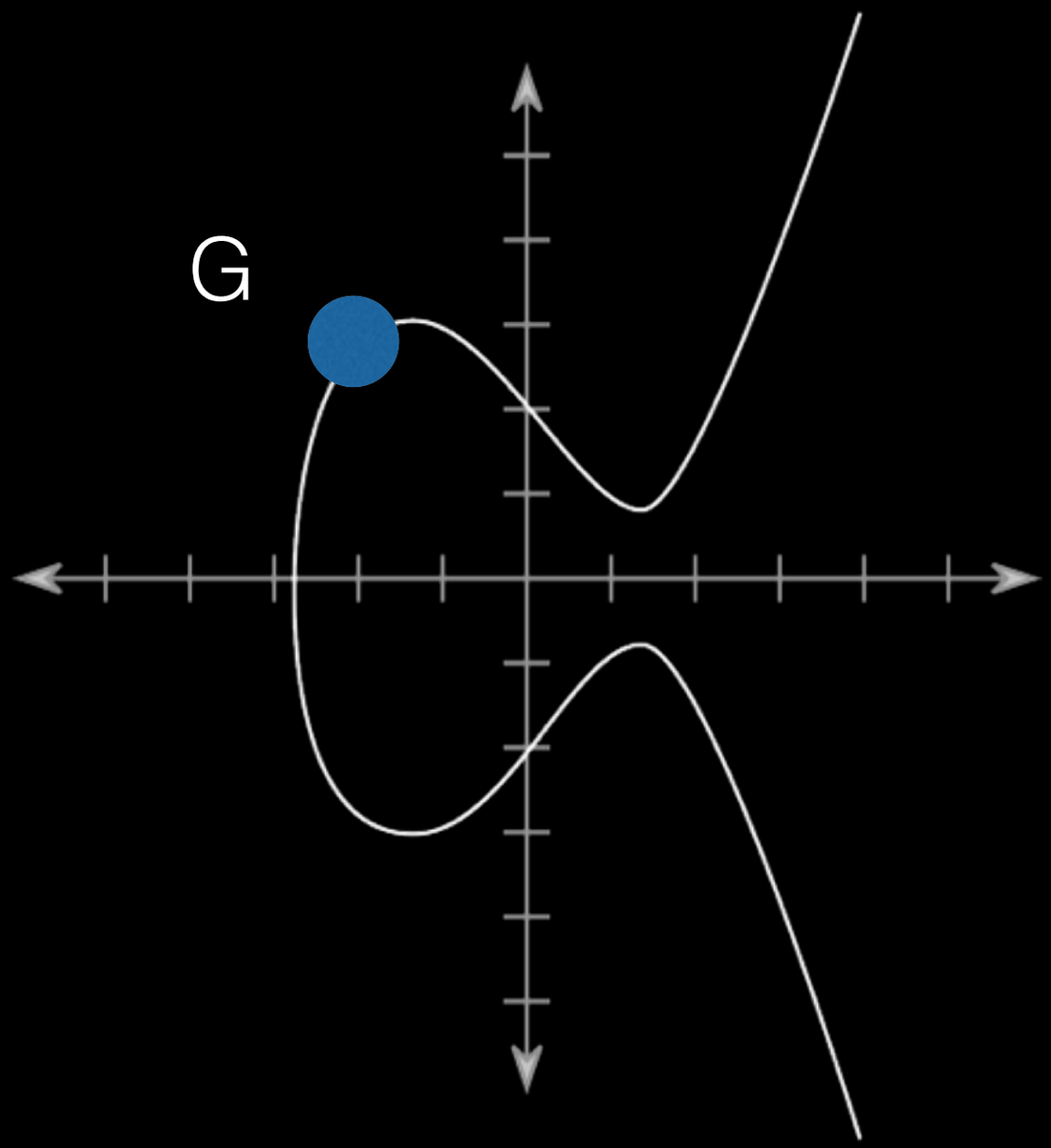
$$y^2 = x^3 + 7$$



(not actually $y^2 = x^3 + 7$)

@wolever

2. pick a base point



@wolever

3. we each pick a secret, random number (*a* and *b*)

@wolever

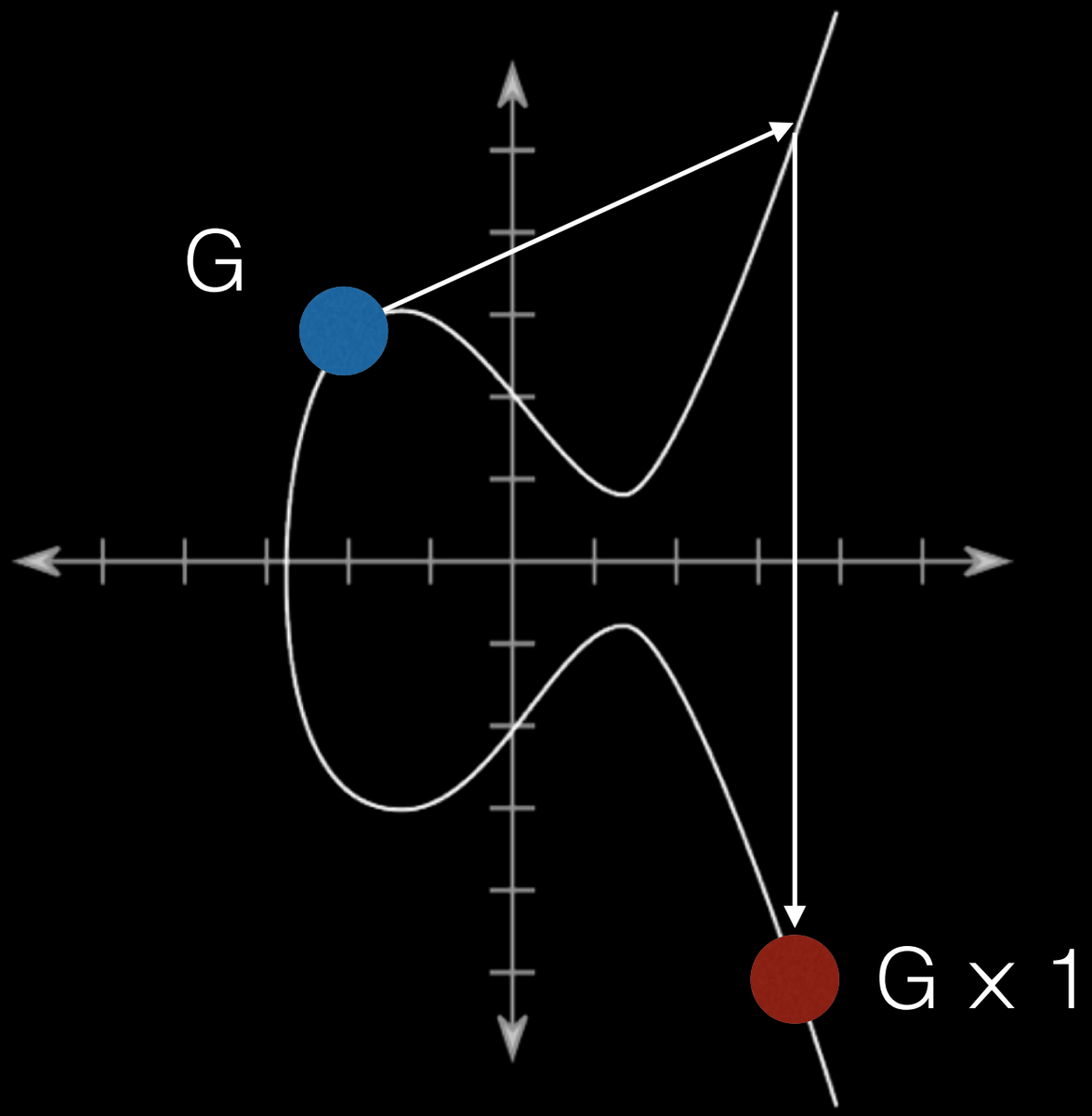
3. multiply the base point by that number

$$A = G \times a$$

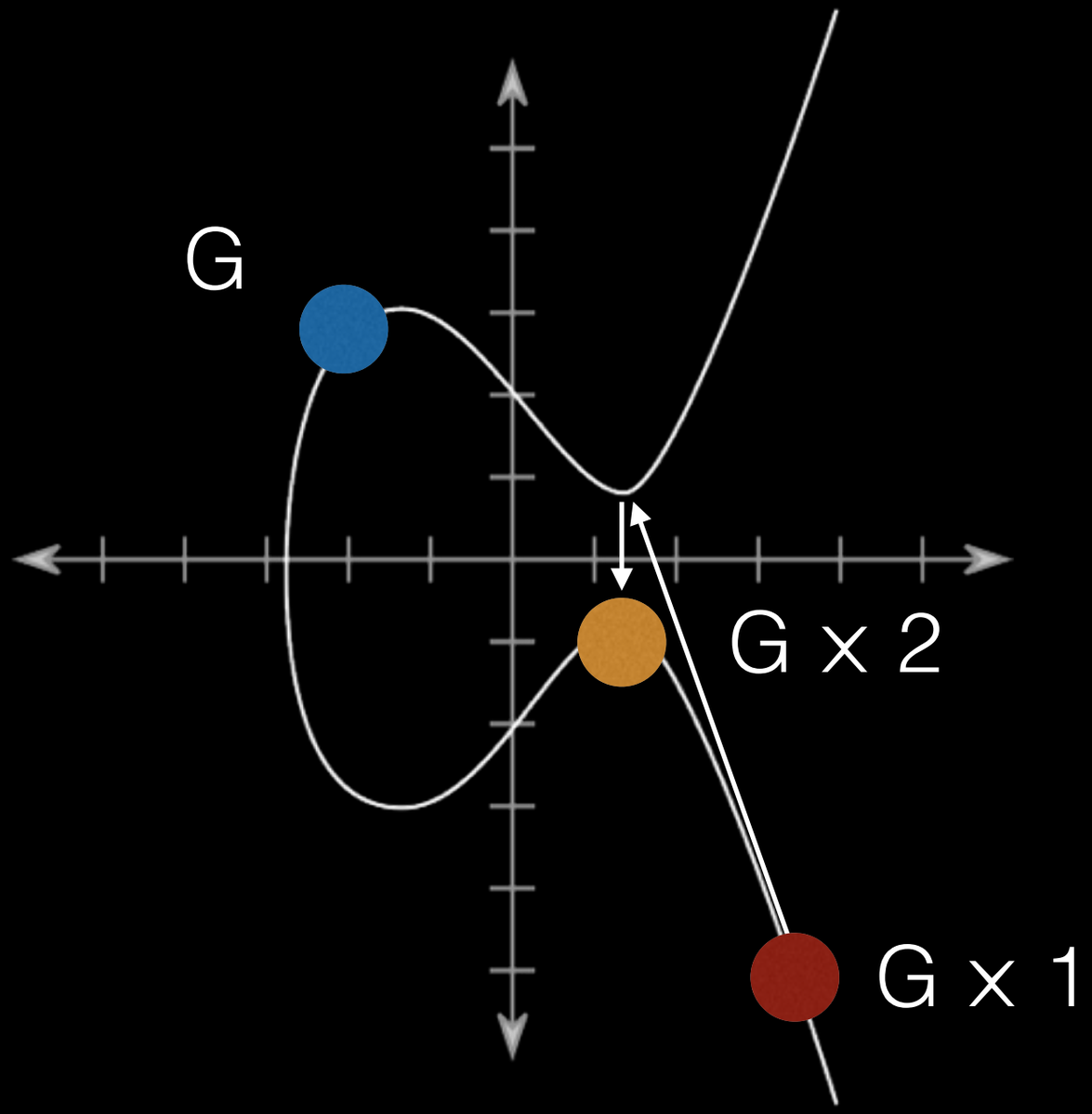
$$B = G \times b$$

… multiply a point?

@wolever



@wolever



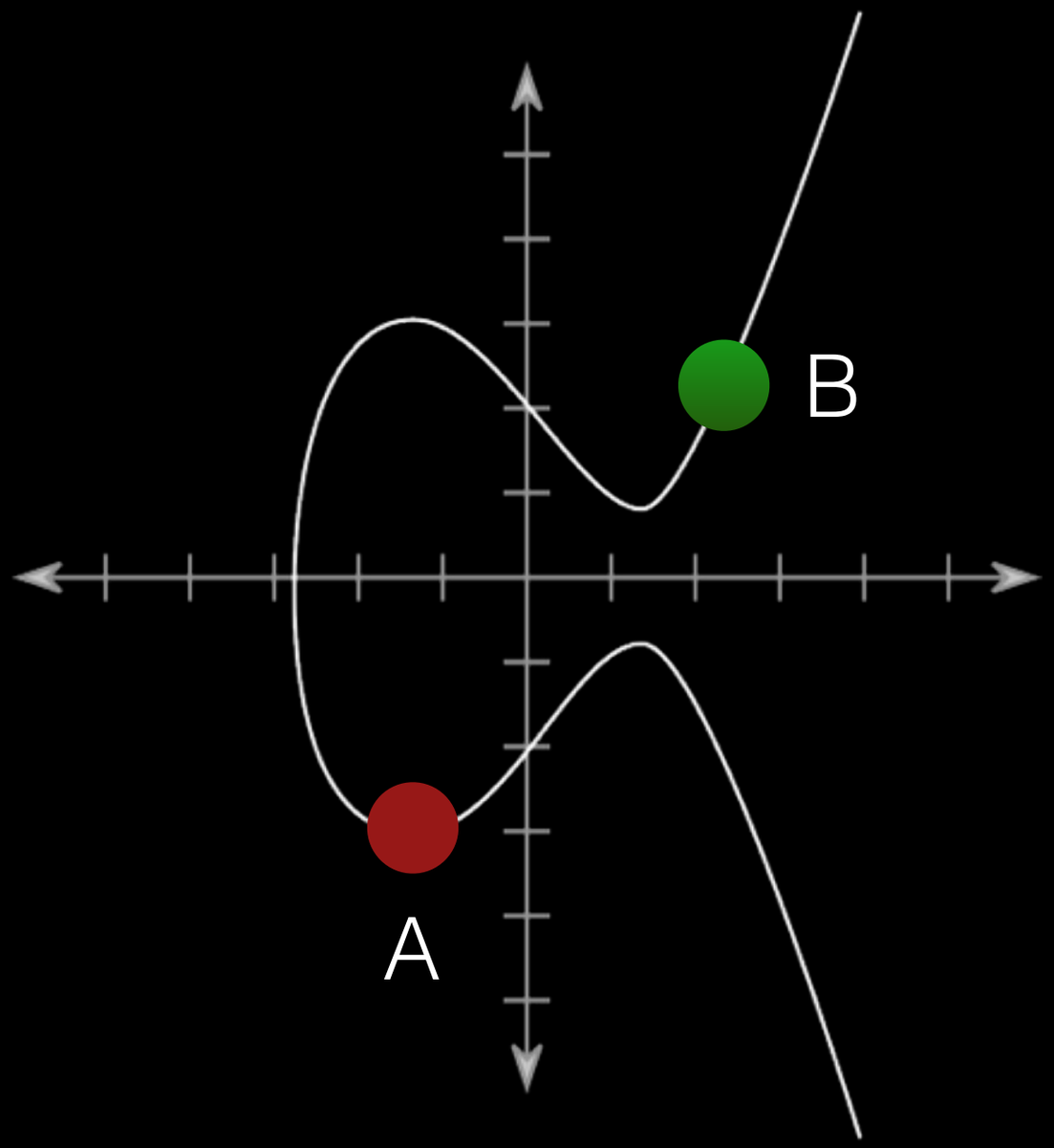
@wolever

3. multiply the base point by that number

$$A = G \times a$$

$$B = G \times b$$

@wolever



@wolever

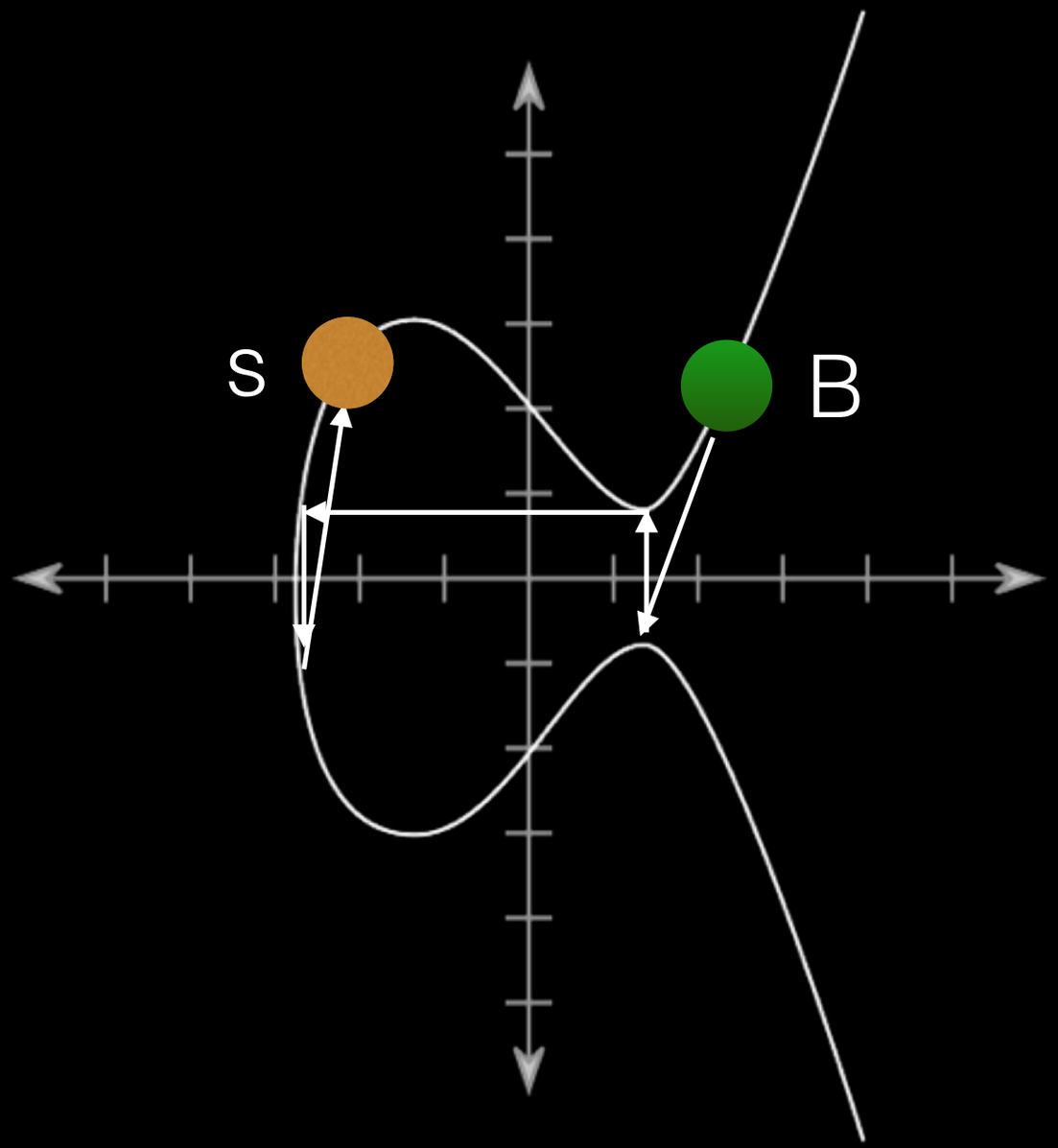
4. share those points

@wolever

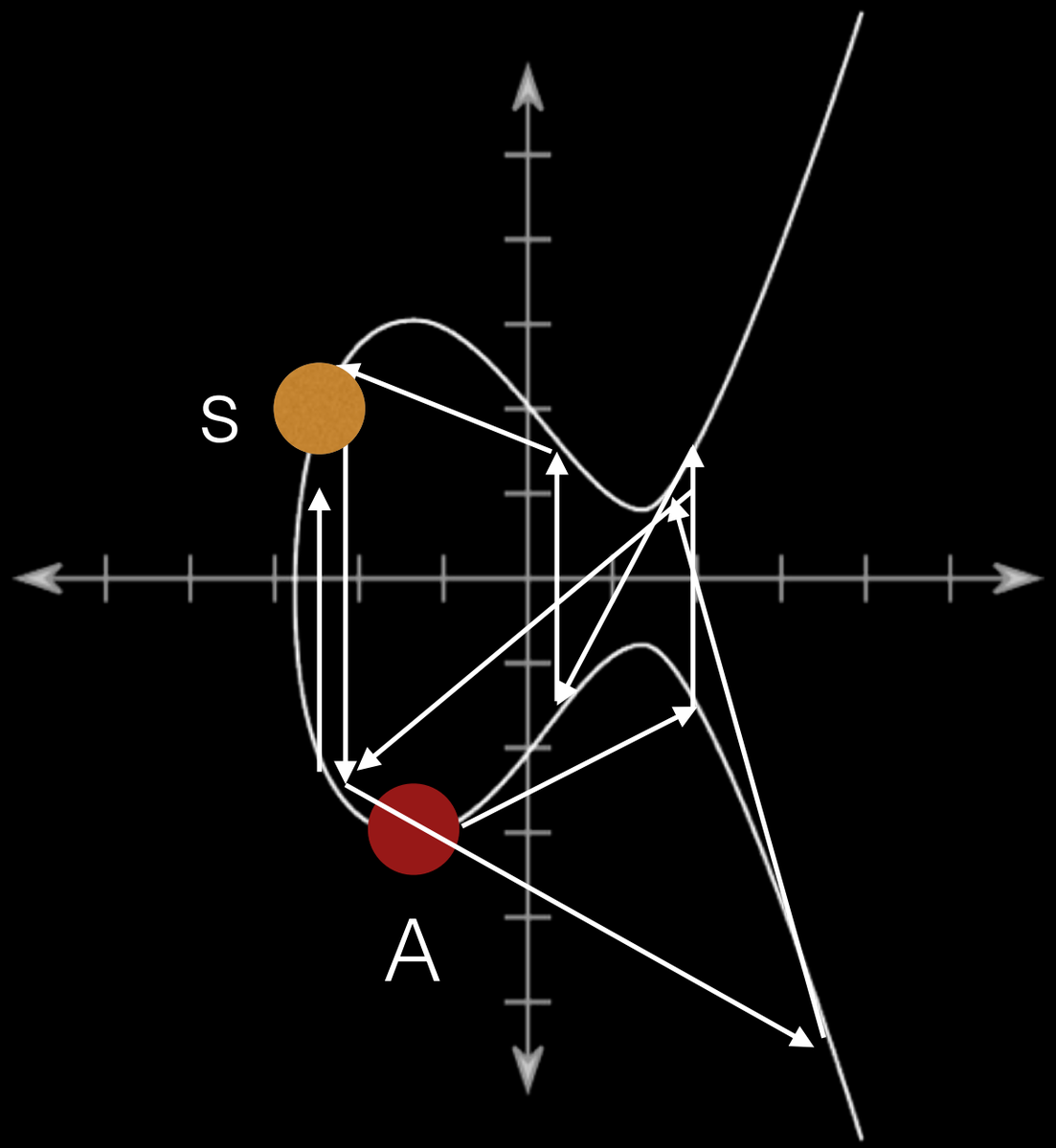
5. multiply the points by our secret numbers

$$s = A \times b$$

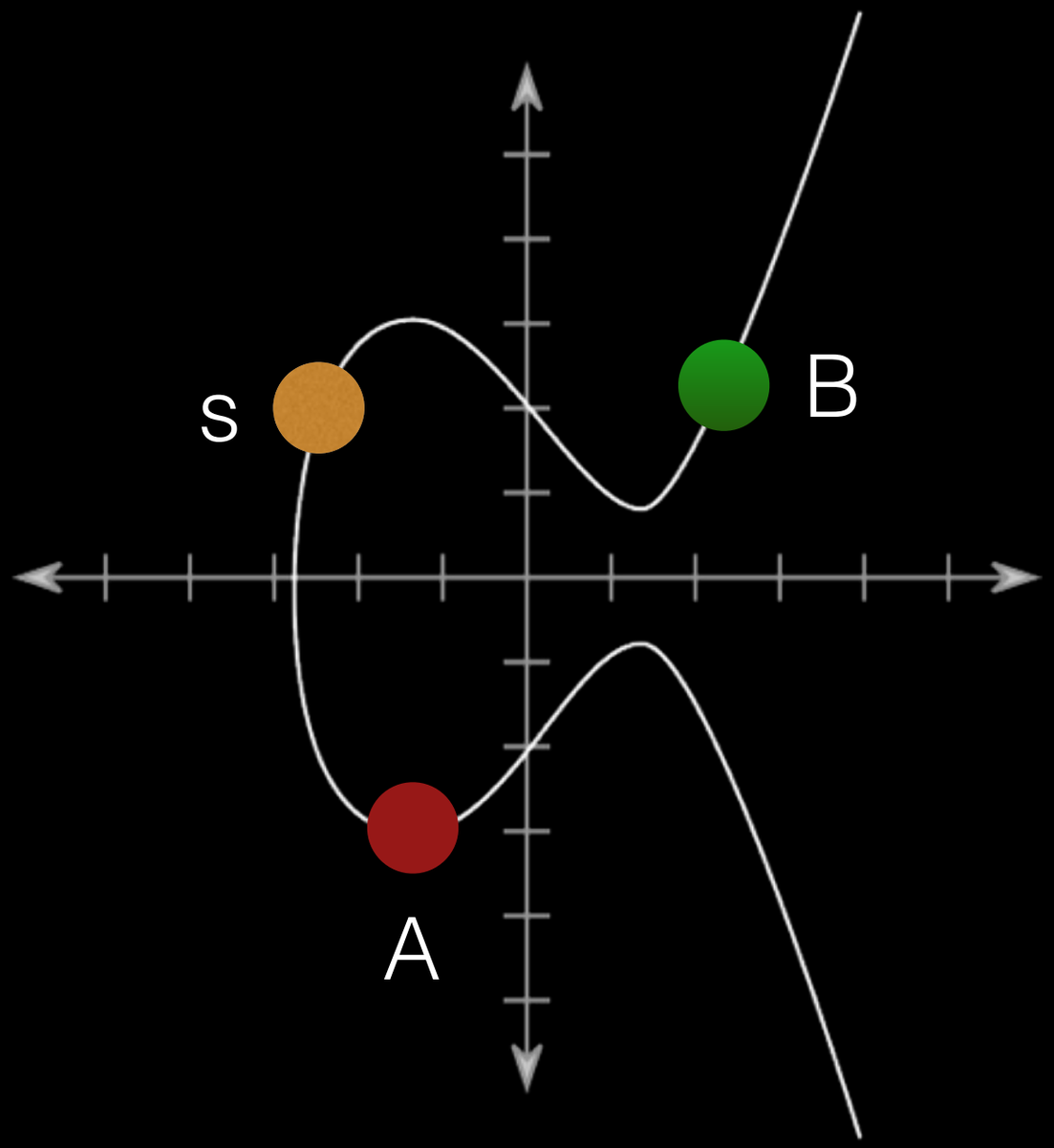
$$s = B \times a$$



@wolever



@wolever



@wolever

really not *that* hard!

@wolever

... but

@wolever

DO ***NOT*** DO THIS YOURSELF

@wolever

libsodium cryptography

@wolever

```
>>> from cryptography.fernet import Fernet
>>> # Put this somewhere safe!
>>> key = Fernet.generate_key()
>>> f = Fernet(key)
>>> token = f.encrypt(b"A really secret message. Not for
prying eyes.")
>>> token
'...'
>>> f.decrypt(token)
'A really secret message. Not for prying eyes.'
```

@wolever

AKINDI

PRICING SIGN IN

Grading that gives you actionable data. Instantly.

Grade multiple choice assessments using regular paper and any scanner!

TRY IT FREE FOR 30 DAYS!

Work with me: wolever@akindi.com

https://akindi.com/pages/jobs