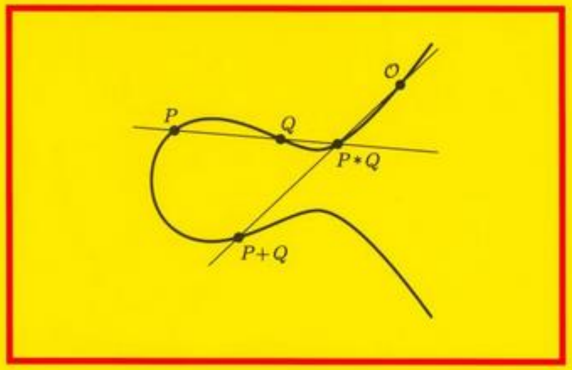
# Elliptic Curve Cryptography
## for those who are afraid of maths

Martijn Grooten, Virus Bulletin
@martijn_grooten
BSides London, 3 June 2015

Undergraduate Texts in Mathematics

Joseph H. Silverman
John Tate

# Rational Points on Elliptic Curves




Springer

# Disclaimer:

This talk will be useless.

I am not a cryptographer.

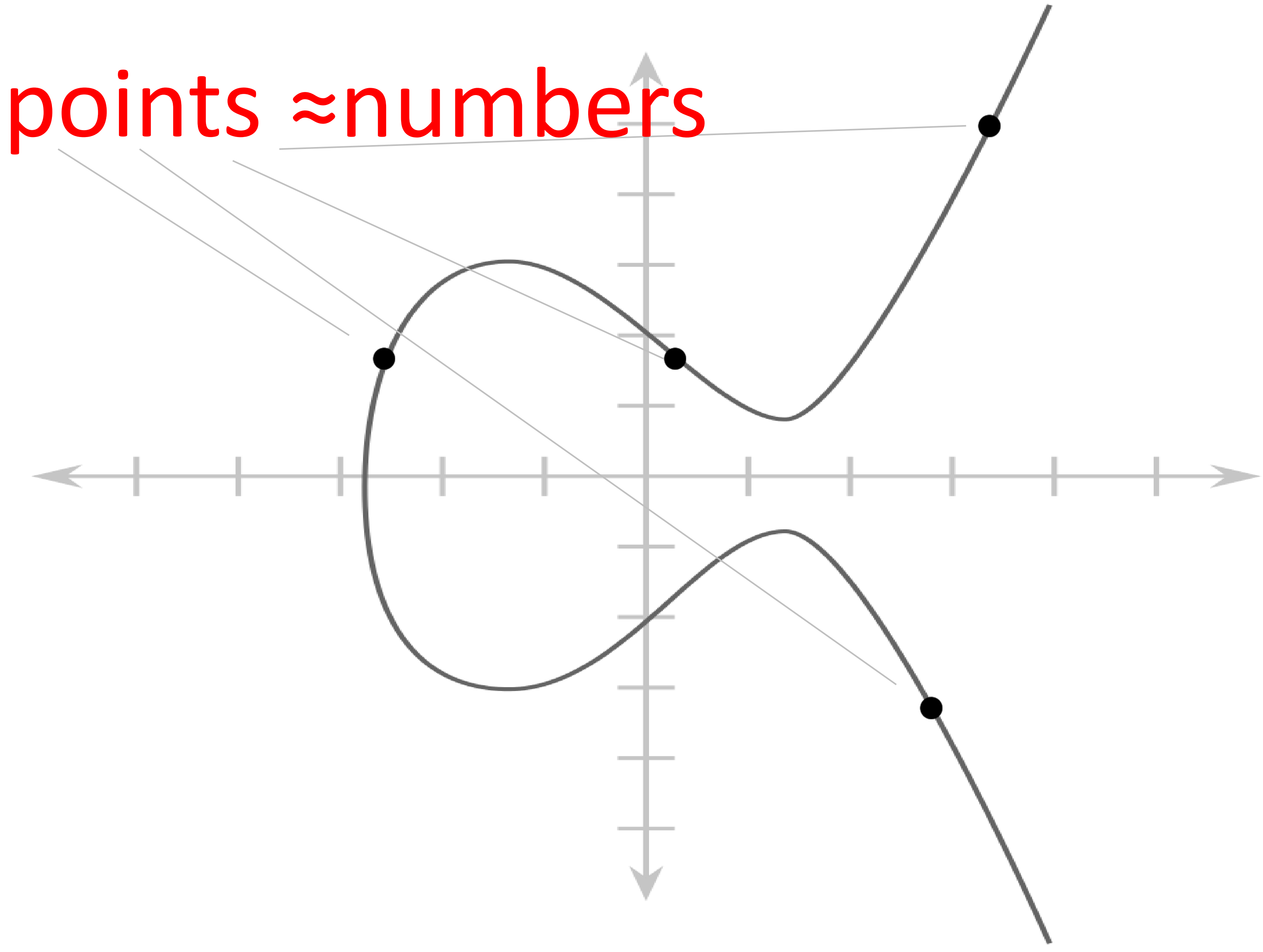
Some things are wrong.

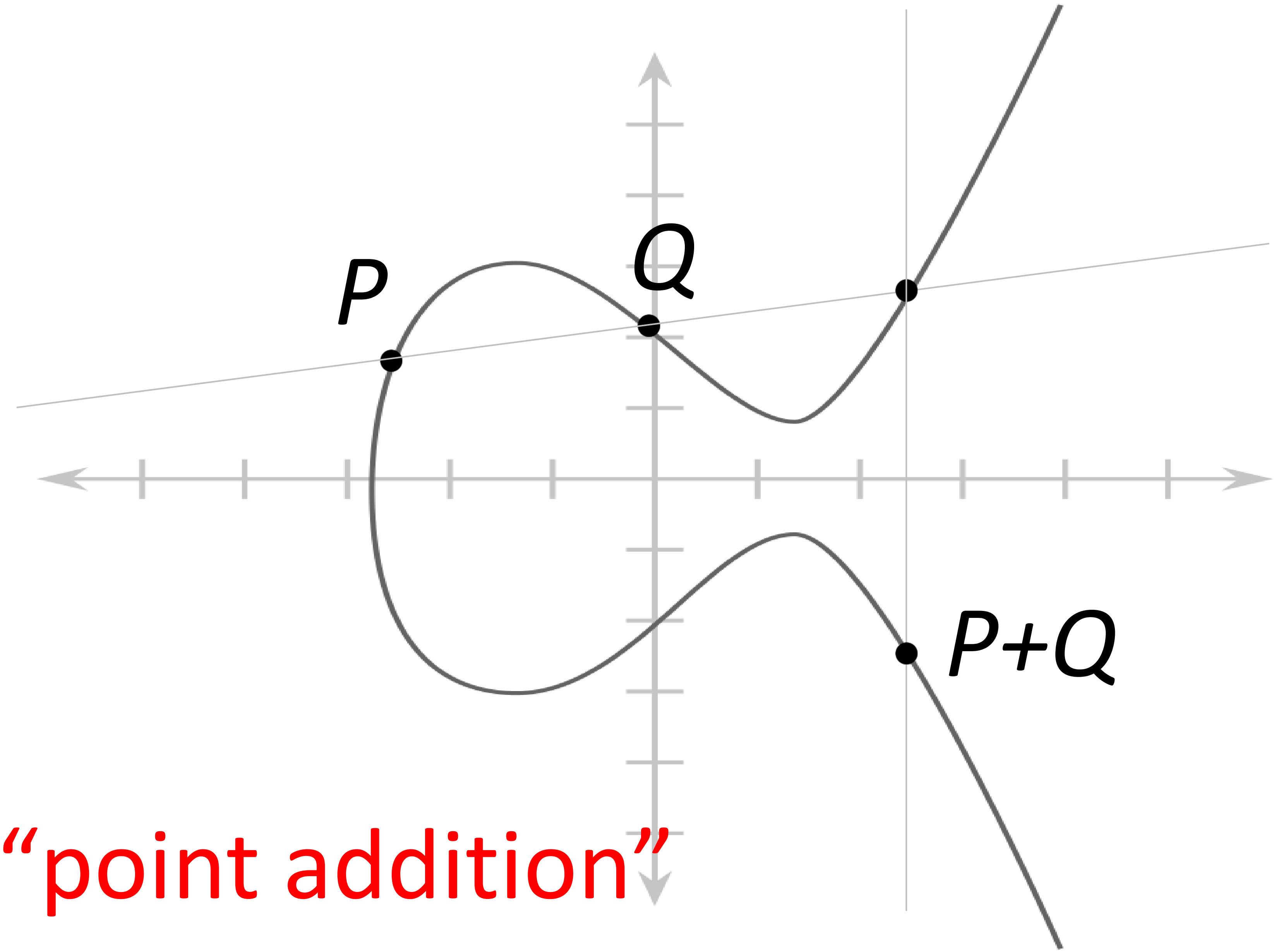
# Elliptic curves

$$y^2 = x^3 + a \cdot x + b$$

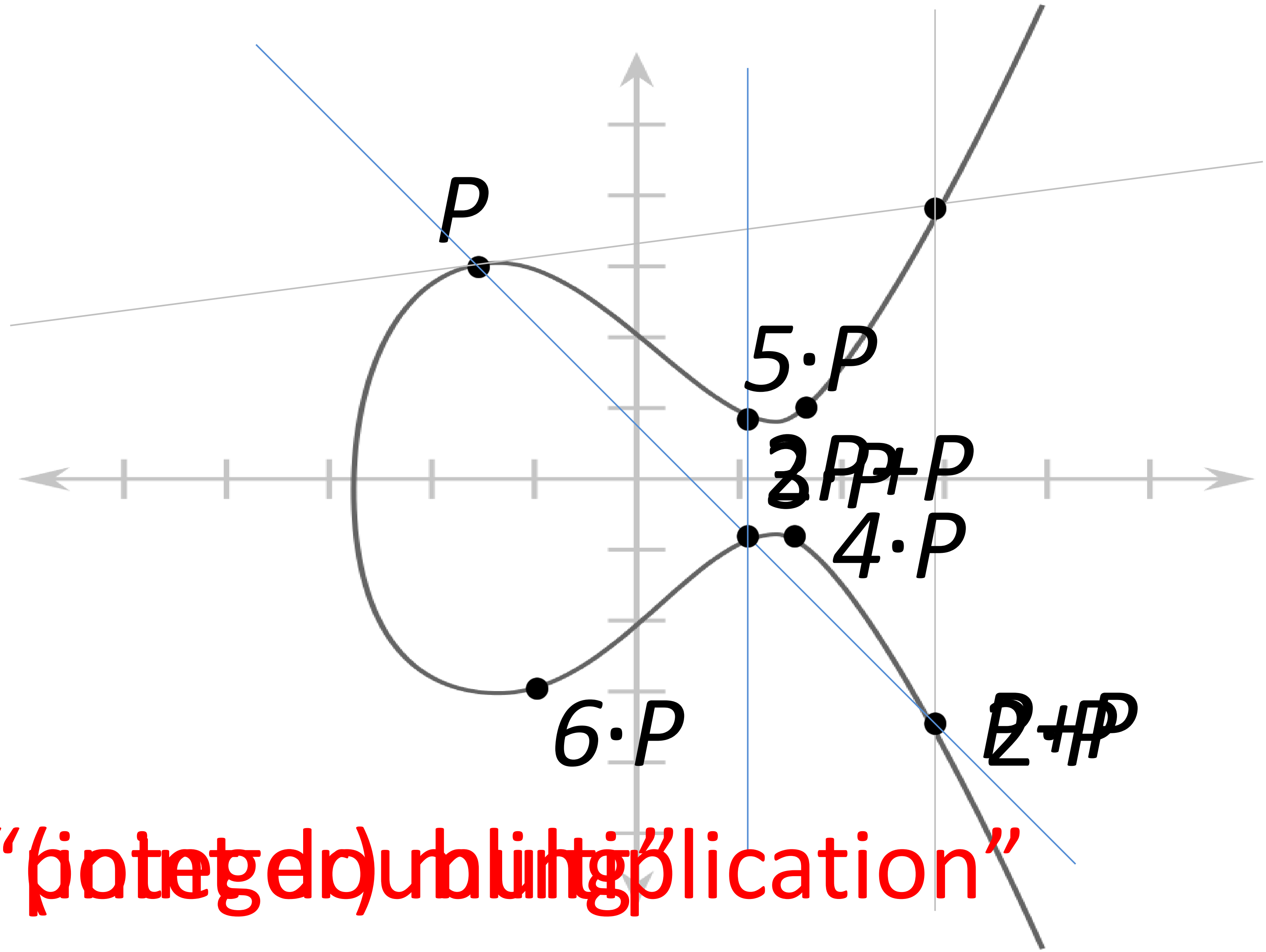
...and a prime number $p$.

**choice!**


points ≈numbers




“point addition”




“point doubling”

“(integer) multiplication”

# So:

We can "add" points to each other.

We can "multiply" points by an integer.

Nice: $P + Q = Q + P$

$3 \cdot P + P = 2 \cdot P + 2 \cdot P = 4 \cdot P$

$5 \cdot (7 \cdot P) = 7 \cdot (5 \cdot P)$

etc.

The points on a curve form an *Abelian Group* (very exciting!).

# Multiplication is very fast

To go from a point $P$ to $100 \cdot P$:

$P \to 2 \cdot P$

$2 \cdot P \to 3 \cdot P$

$3 \cdot P \to 6 \cdot P$

$6 \cdot P \to 12 \cdot P$

$12 \cdot P \to 24 \cdot P$

$24 \cdot P \to 25 \cdot P$

$25 \cdot P \to 50 \cdot P$

$50 \cdot P \to 100 \cdot P$

Only eight steps!

# “Division” is very slow

Given points $P$ and $Q$, where $Q=n\cdot P$, the best way to find the number $n$ is to try $P$, $2\cdot P$, $3\cdot P$, etc. That is very slow.

The *Discrete Logarithm Problem* for elliptic curves.

# ECDH (Elliptic Curve Diffie Hellman)

The challenge: Alice and Bob want to agree on a secret key over a public channel.

For example: Alice is a web server, Bob a browser and they want to exchange a key to encrypt a TLS session.

# ECDH (Elliptic Curve Diffie Hellman)

Alice and Bob have agreed on an elliptic curve and a “base point” $P$ on the curve.

Alice chooses secret large random number $a$.

Bob chooses secret large random number $b$.

# ECDH (Elliptic Curve Diffie Hellman)

Alice computes $a{\cdot}P$ ($a$ times the point $P$) and shares the answer with Bob.

Bob computes $b{\cdot}P$ and shares this too.

Alice computes $a{\cdot}(b{\cdot}P)$ ($a$ times the point Bob gave her).

Bob computes $b{\cdot}(a{\cdot}P)$.

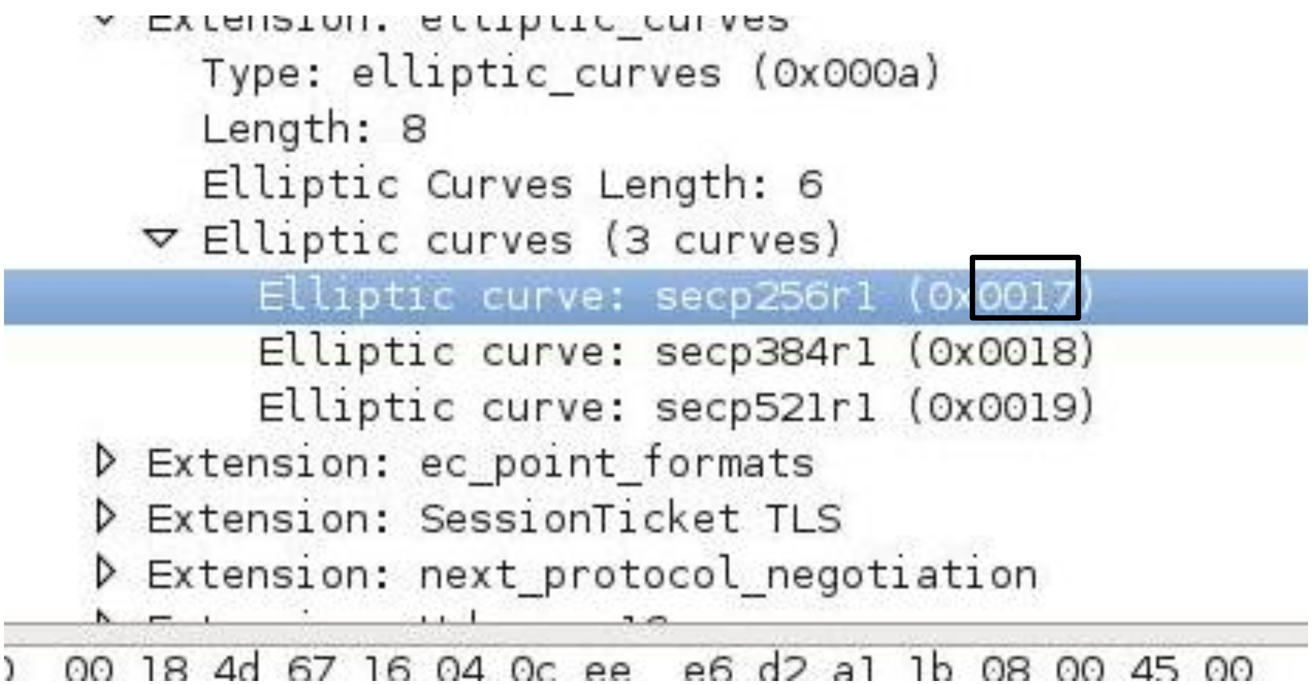
Secret key: $a{\cdot}(b{\cdot}P) = b{\cdot}(a{\cdot}P)$.

# Wireshark (client to server)

```
  Session ID Length: 0
  Cipher Suites Length: 22
▽ Cipher Suites (11 suites)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
    Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
    Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
    Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
  Compression Methods Length: 1
▷ Compression Methods (1 method)
```

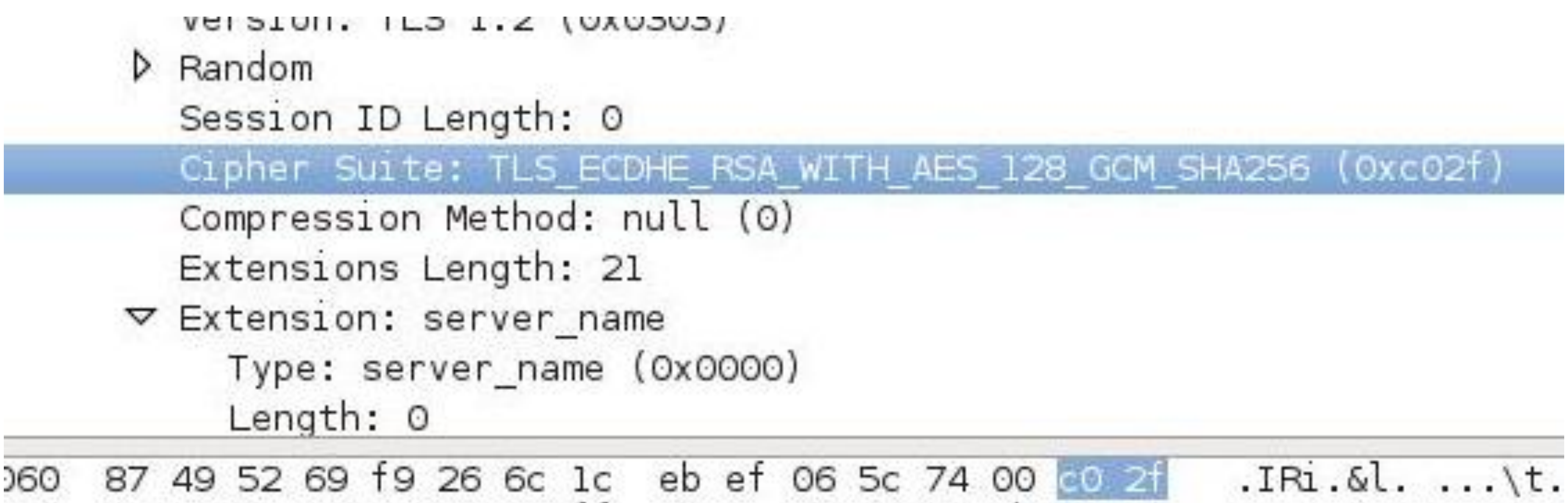
"11 cipher suites you didn't know I supported"

# Wireshark (client to server)




“These are my three favourite curves.”

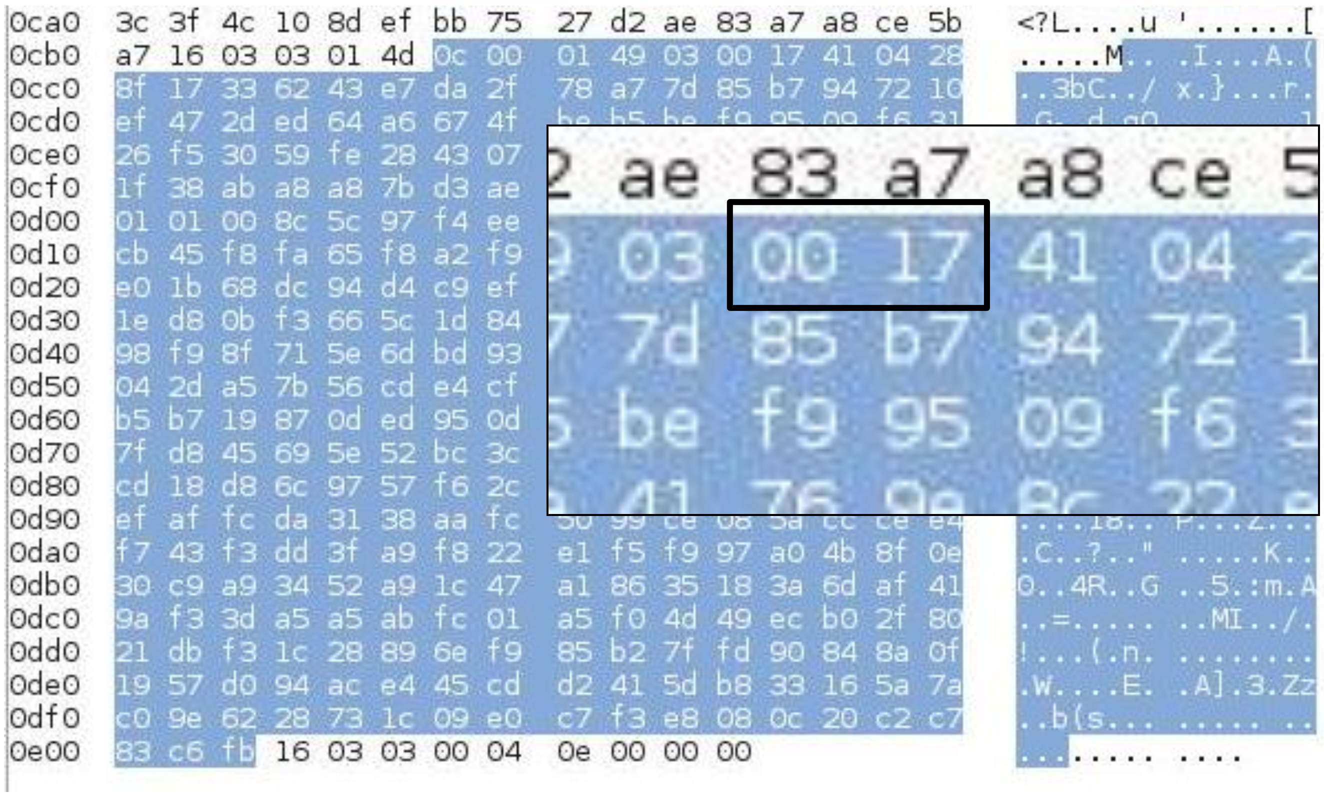
# Wireshark (server to client)




“OK, let’s go for

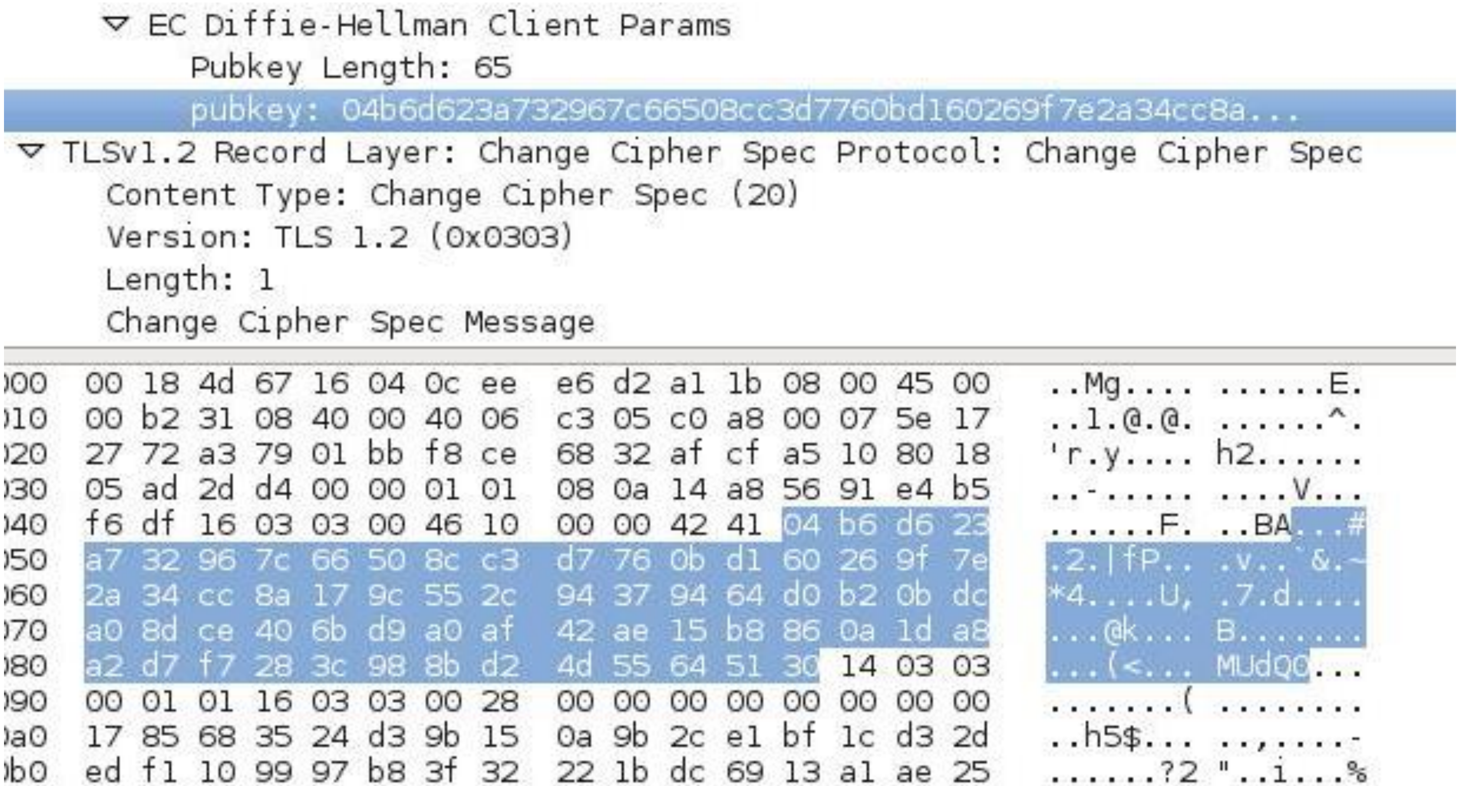
TLS_ECDHE_RSA_WITH_AES_128_CGM_SHA256 .”

# Wireshark (server to client)




"And curve NIST P-256. And this is my point."

# Wireshark (client to server)




"Cheers – here's mine!"

# What could possibly go wrong?

What if there is a ‘loop’?

If $1001 \cdot P = P$, then there are only 1000 possible values for $n \cdot P$, **no matter how large *n* is!**

Loops can be avoided. Other (known and unknown!) weaknesses remain possible.

# Are we using 'weak' curves?

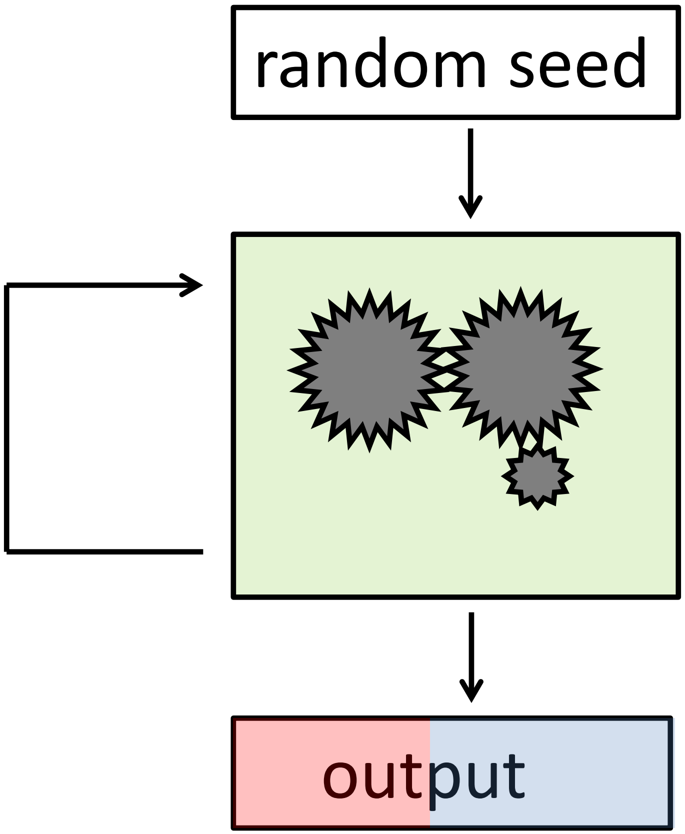
NIST P-256:

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

4105836372515214212932612978004
7268409114441015993725554835256
314039467401291

**WHAT???**

# Random number generators

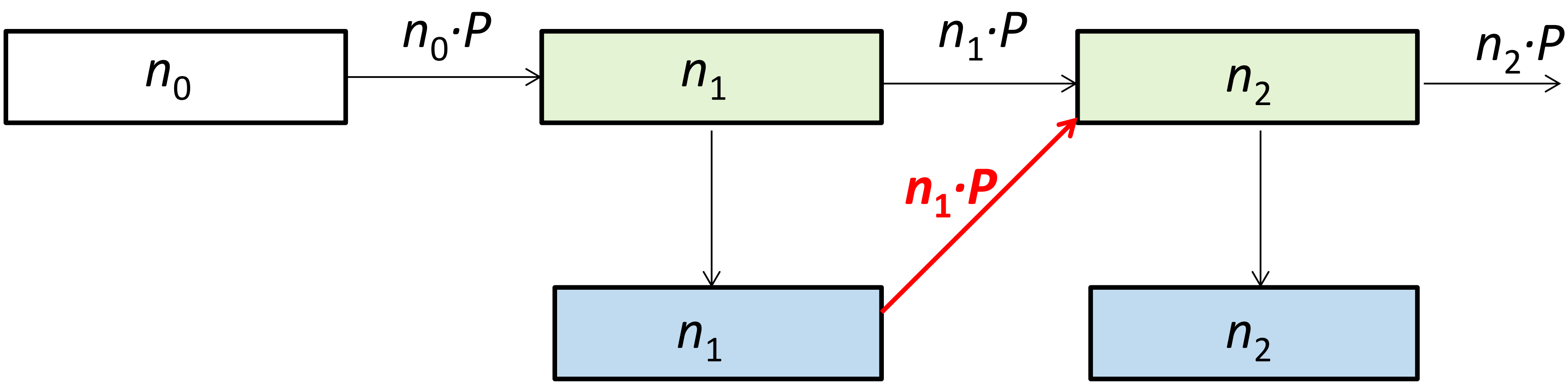


# Random number generators

Discrete Logarithm Problem:

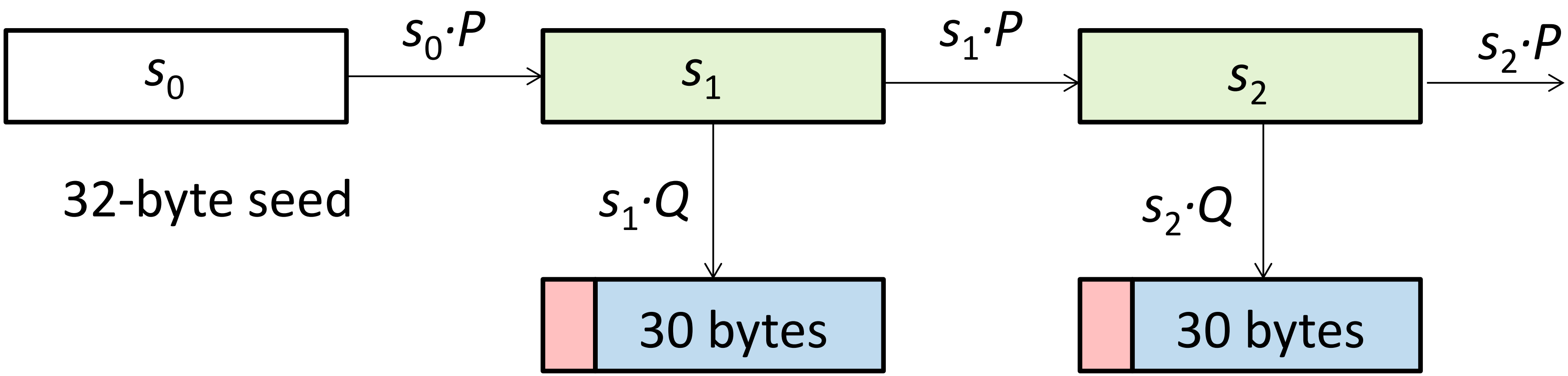
$$n \rightarrow n \cdot P$$

gives "random" points/numbers.



# Random number generators

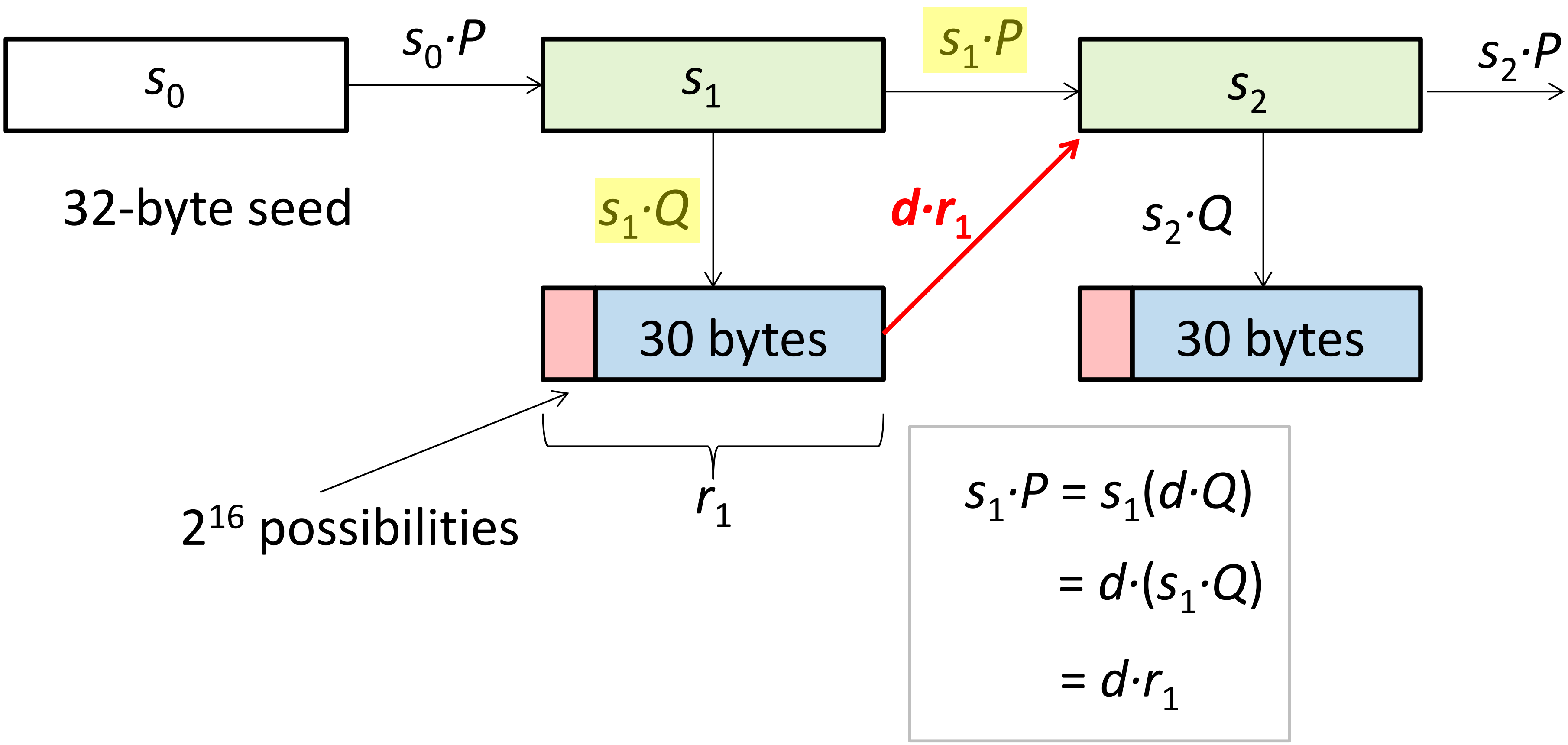
Given: elliptic curve with two points *P* and *Q*.




*Note: ideas from this slide and the next are borrowed from Bernstein, Heninger and Lange (NCSC '14).*

# Random number generators

Fact: $P = d \cdot Q$ for some (large) number $d$.



# So who, if anyone, knows *d*?

**NIST SP 800-90A** **January 2012**

## Acknowledgements

The National Institute of Standards and Technology (NIST) gratefully acknowledges and appreciates contributions by Mike Boyle and Mary Baish from NSA for assistance in the development of this Recommendation. NIST also thanks the many contributions by the public and private sectors.

"Dual_EC_DRBG"

# Conclusion

Elliptic curve cryptography is a good idea because we can do with much smaller keys.

256-bit ECC ≈ 3072-bit RSA.

Elliptic curve crypto uses complicated maths. **That is its biggest weakness**.

# Thank you!

@martijn_grooten

martijn.grooten@virusbtn.com

www.virusbtn.com

PS VB2015, Prague 30 Sep-2 Oct