CMU Computer Club Talk Series

Spring 2015

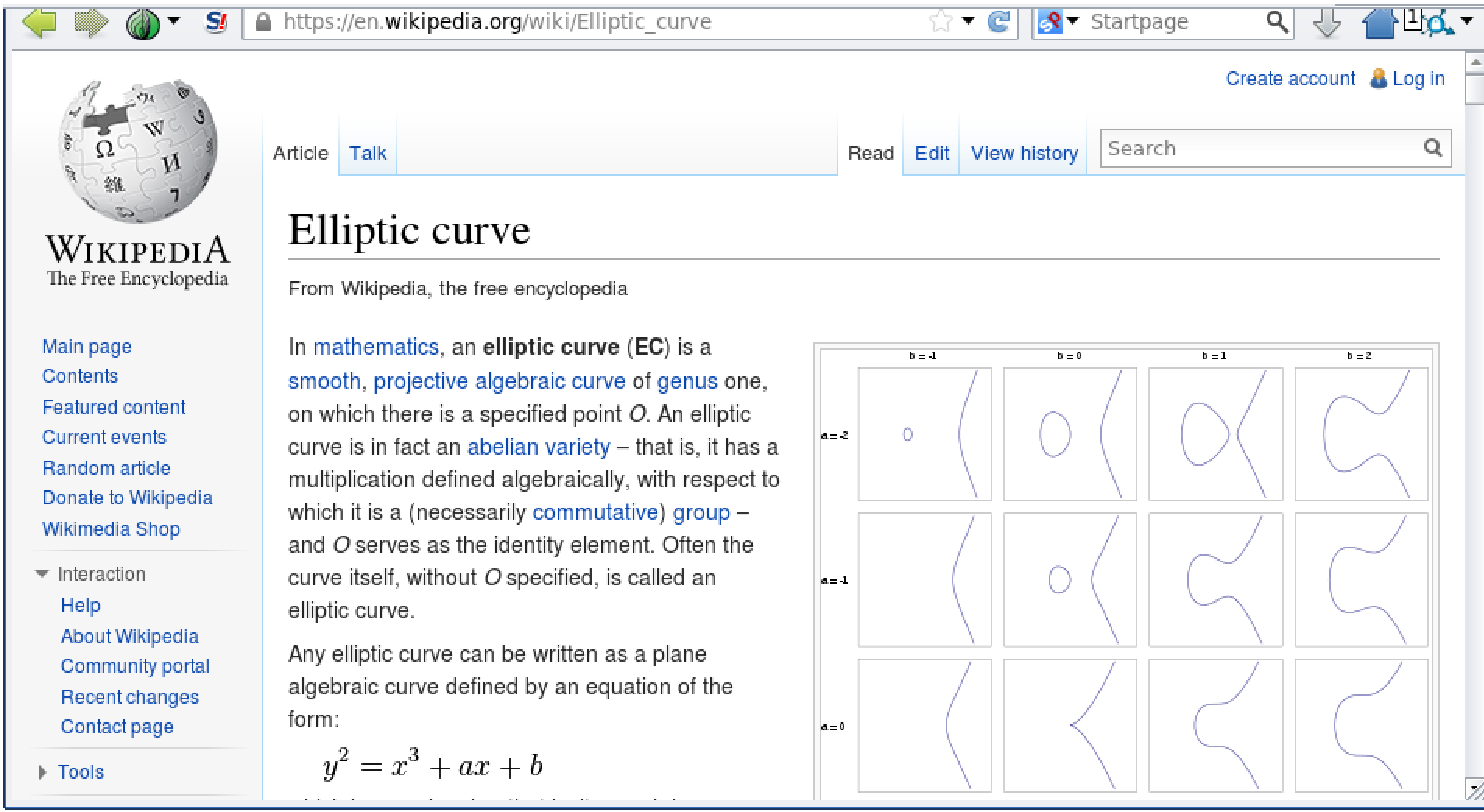
Elliptic Curve Cryptography

We would like to thank Green Hills Software for sponsoring this talk series

Green Hills make the world's highest performing compilers, most secure real-time operating systems, revolutionary debuggers, and virtualization solutions for embedded systems.

What is an elliptic curve?

What is an elliptic curve?


https://en.wikipedia.org/wiki/Elliptic_curve

Startpage

Create account Log in

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

Article Talk Read Edit View history Search

Elliptic curve

From Wikipedia, the free encyclopedia

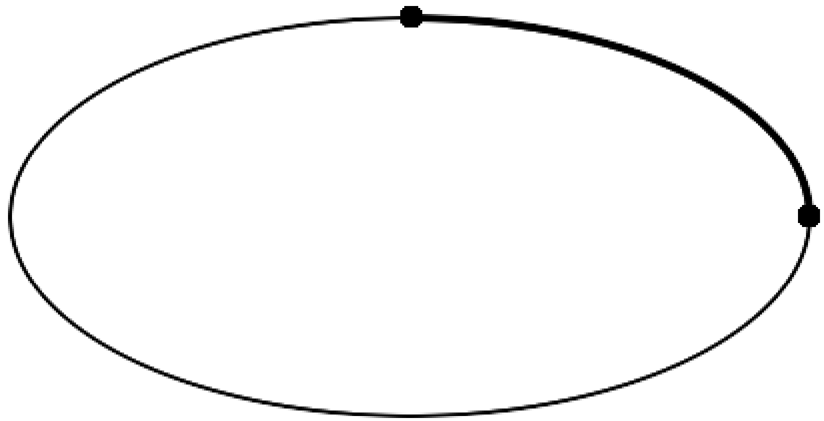
In mathematics, an **elliptic curve** (**EC**) is a smooth, projective algebraic curve of genus one, on which there is a specified point *O*. An elliptic curve is in fact an abelian variety – that is, it has a multiplication defined algebraically, with respect to which it is a (necessarily commutative) group – and *O* serves as the identity element. Often the curve itself, without *O* specified, is called an elliptic curve.

Any elliptic curve can be written as a plane algebraic curve defined by an equation of the form:

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

What is an elliptic curve?

The idea of elliptic curves comes from the problem of finding the arc length of an ellipse.



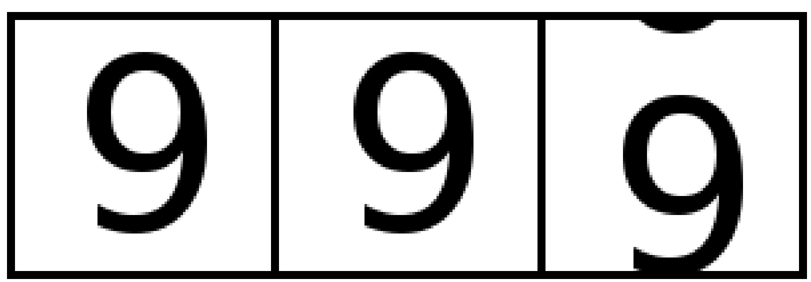
Public-key cryptography

- Proposed by Merkle, 1974
- First practical method described by Diffie & Hellman, 1976

What is modular arithmetic?

What is modular arithmetic?

Mod 1000



This is known as a mathematical ring

Multiplication is also possible

500 * 2 = 000

What about division?

000 ÷ 2 = ?

001 ÷ 2 = ?

Prime number as modulus

Mod 5

1 x 1 = 1	1 x 2 = 2	1 x 3 = 3	1 x 4 = 4
2 x 1 = 2	2 x 2 = 4	2 x 3 = 1	2 x 4 = 3
3 x 1 = 3	3 x 2 = 1	3 x 3 = 4	3 x 4 = 2
4 x 1 = 4	4 x 2 = 3	4 x 3 = 2	4 x 4 = 1

Where division (multiplictive inverse) is defined for every nonzero element, the ring is known as a field

Finite fields are also known as
Galois fields,
after Évariste Galois (1811-1832)

Exponentials and logarithms are also possible in a finite field.

However...

There is no known polynomial-time algorithm for finding logarithms in a finite field.

This is known as the discrete logarithm problem.

Diffie-Hellman Key Exchange

Prearranged generator g, prime modulus p

Alice	Bob
Secret x	Secret y
Calculate g^x	Calculate g^y

Exchange g^x, g^y

Calculate $(g^y)^x = g^{xy}$	Calculate $(g^x)^y = g^{xy}$

Choosing g and p

Ideally, we want g to generate every nonzero element when multiplied with itself.

For example:

mod 5

$2^1 = 2$

$2^2 = 4$

$2^3 = 3$

$2^4 = 1$

The sequence will repeat every p-1 elements (Fermat's little theorem).

But what if...

mod 31

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 1$$

$$2^6 = 2$$

...

The sequence repeats every 5 elements! This makes finding discrete logarithms a little too easy.

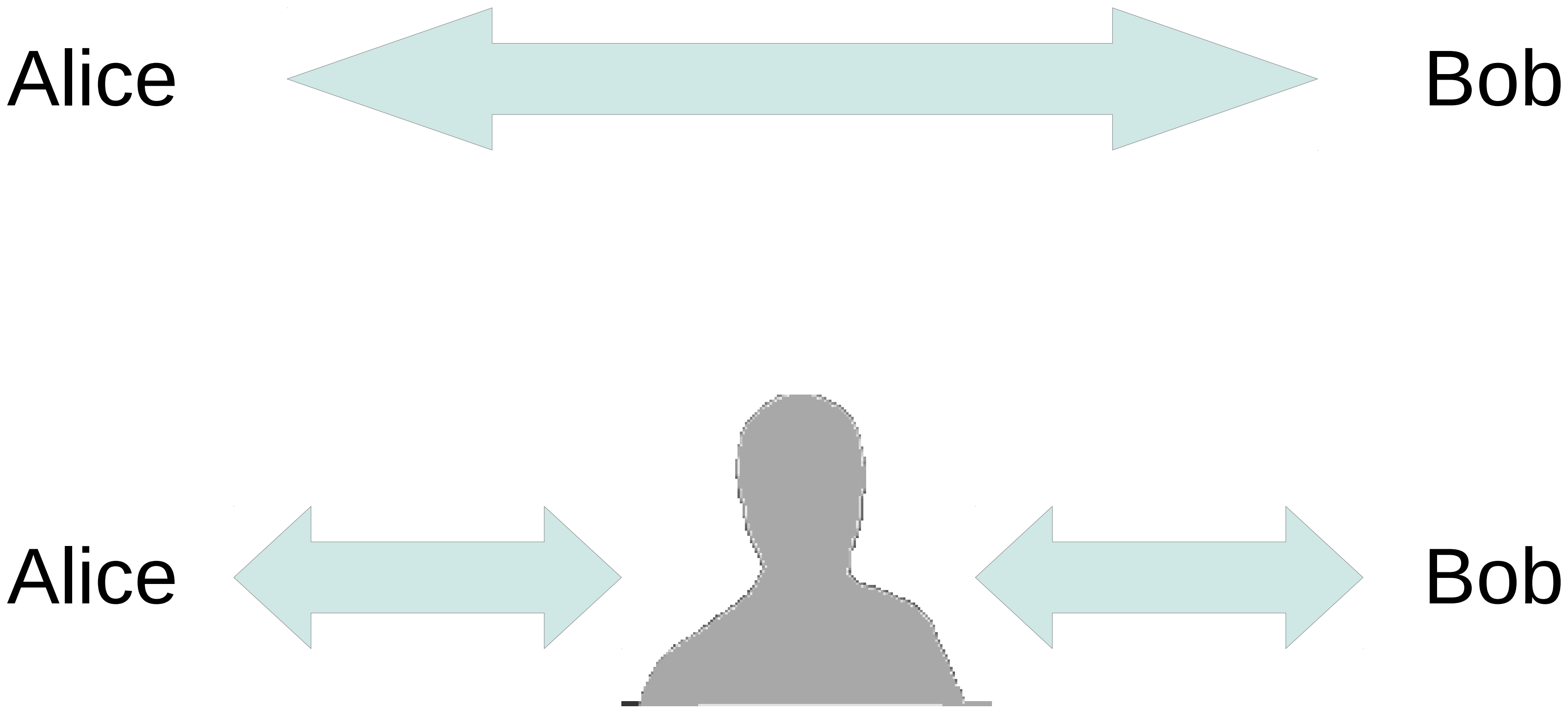
Avoiding multiplicative subgroups

To avoid this problem, choose p such that p-1 does not have many small factors.

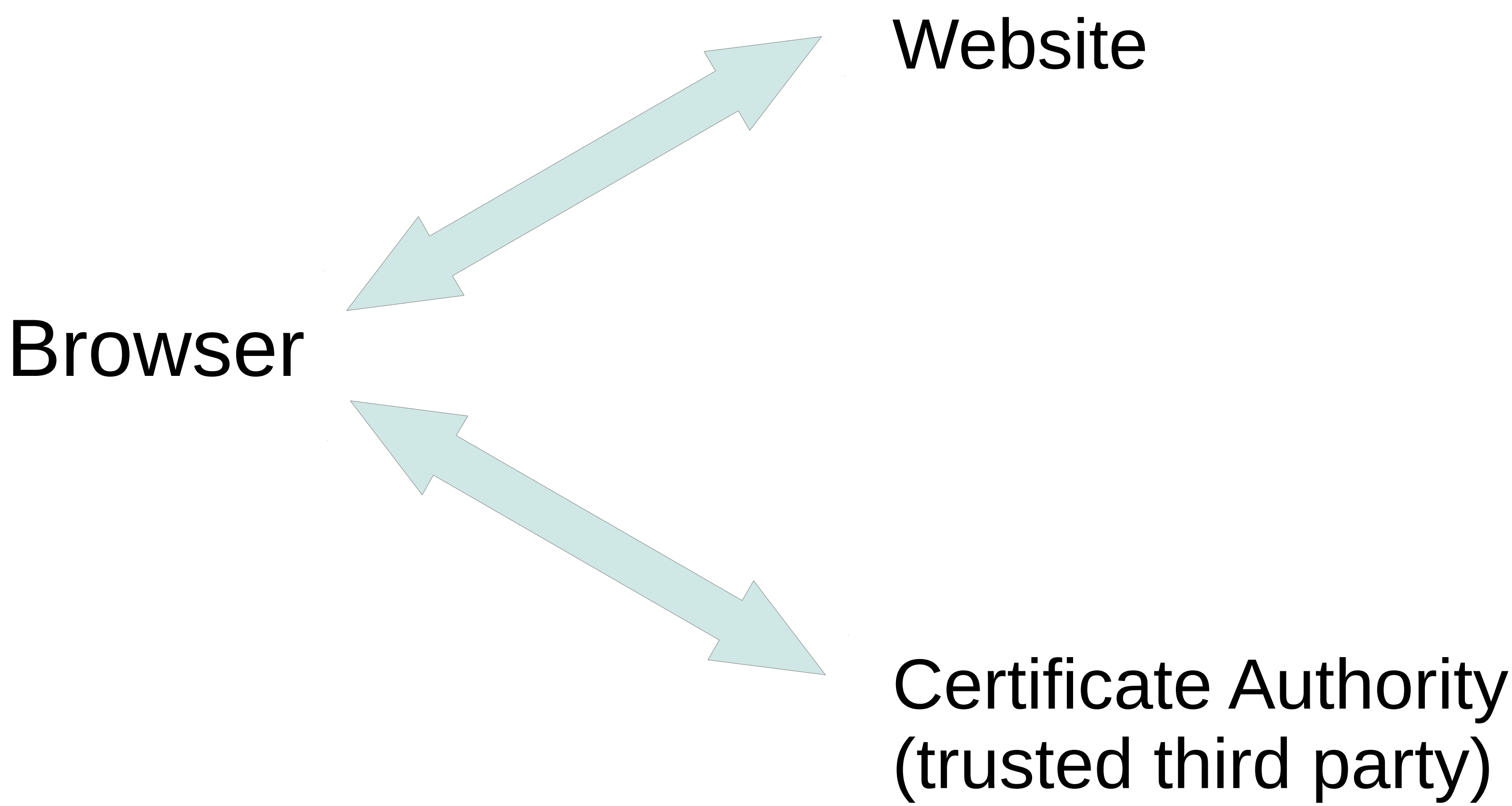
31 is a poor choice because 31-1 = 2 x 3 x 5

Generally it is preferred to choose prime p, such that (p-1)/2 is also prime.

The Man in the Middle



Authenticating Public Keys



ElGamal signature scheme

- Proposed by Tahir ElGamal in 1984
- Prearranged generator g, prime modulus p
- Private key x, public key $y = g^x$
- To sign message m:

  Choose random k

  $r = g^k \pmod{p}$

  $s = (m - xr)k^{-1} \pmod{p-1}$

Signature verification

Verify $g^m = y^r r^s \pmod{p}$

(which is equivalent to g^{xr+ks})

NIST Digital Signature Algorithm (DSA) rearranges the terms a bit:

$g^{m/s} y^{r/s} = r$

What if k is not random?

Signature $g^m = y^r r^s \pmod{p}$

So, $m = xr+ks$, $r = g^k$

If two messages are signed using same k, we have:

$m_1 = xr+ks_1$ and $m_2 = xr+ks_2$

...solve for k, x

Ephemeral Diffie-Hellman (eg SSHv2)

- Diffie-Hellman key exchange with random keys
- Authenticate with known keys
- Erase temporary keys when session ends
- Old sessions can not be decrypted if the authentication keys are later exposed (back traffic protection)
- Future sessions can not be decrypted if the session keys are later exposed (forward secrecy)

The Socialist Millionaires

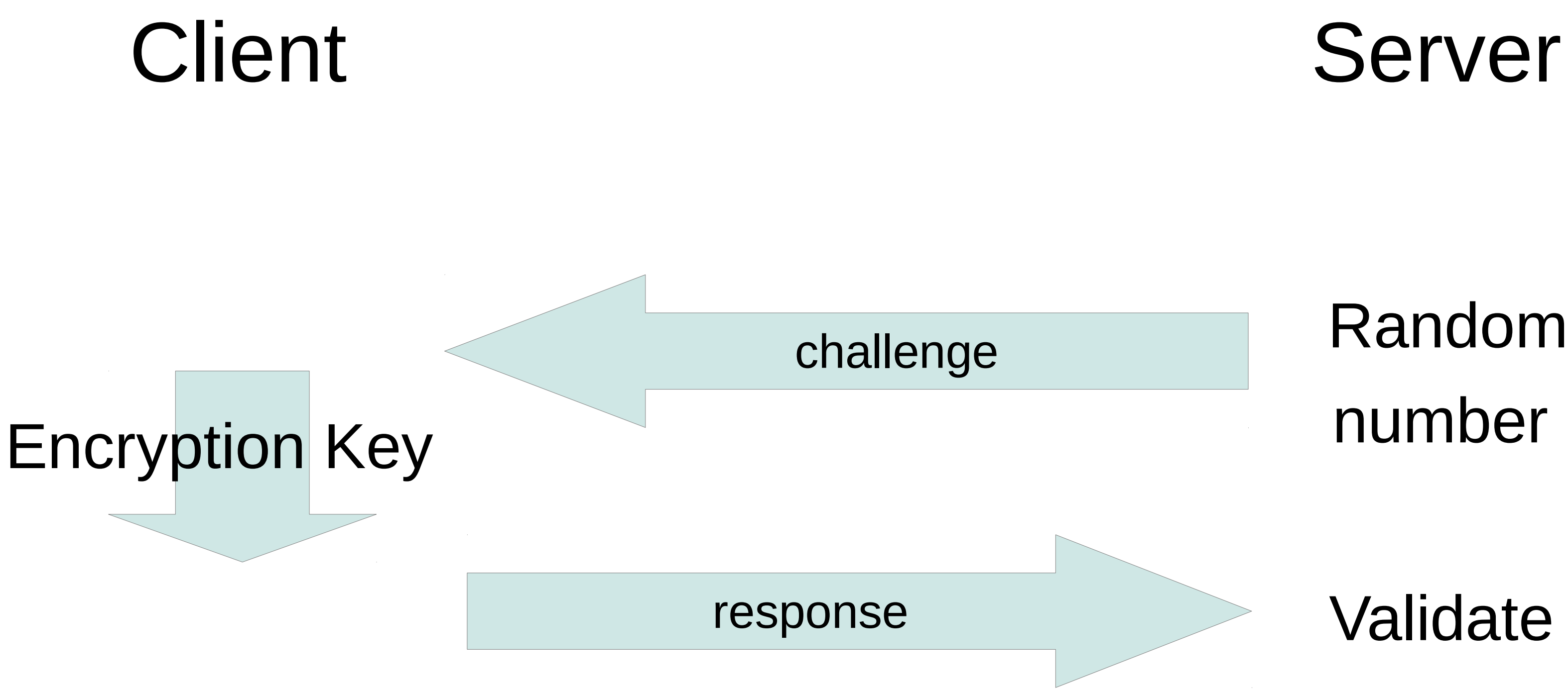
The Socialist Millionaires

Scenario: Two millionaires want to know if they have the same amount of money, but don't want to reveal how much they have.

This is equivalent to the password authentication problem. We want to confirm that both parties have the same password, without revealing the password.

Challenge-Response

(Kerberos, WPA, RADIUS, etc)



Socialist Millionaires Protocol

Random a,b,c,d,e,f   Generator g   Passwords x,y

$$g^{a} \Longleftrightarrow g^{b}$$

$$g^{c} \Longleftrightarrow g^{d}$$

$$g^{e}(g^{ab})^{x} \Longleftrightarrow g^{f}(g^{ab})^{y}$$

$$(g^{e-f}g^{ab(x-y)})^{c} \Longleftrightarrow (g^{e-f}g^{ab(x-y)})^{d}$$

$$g^{cde} \Longleftrightarrow g^{cdf}$$

If x=y then verify $g^{abcd(x-y)}=1$

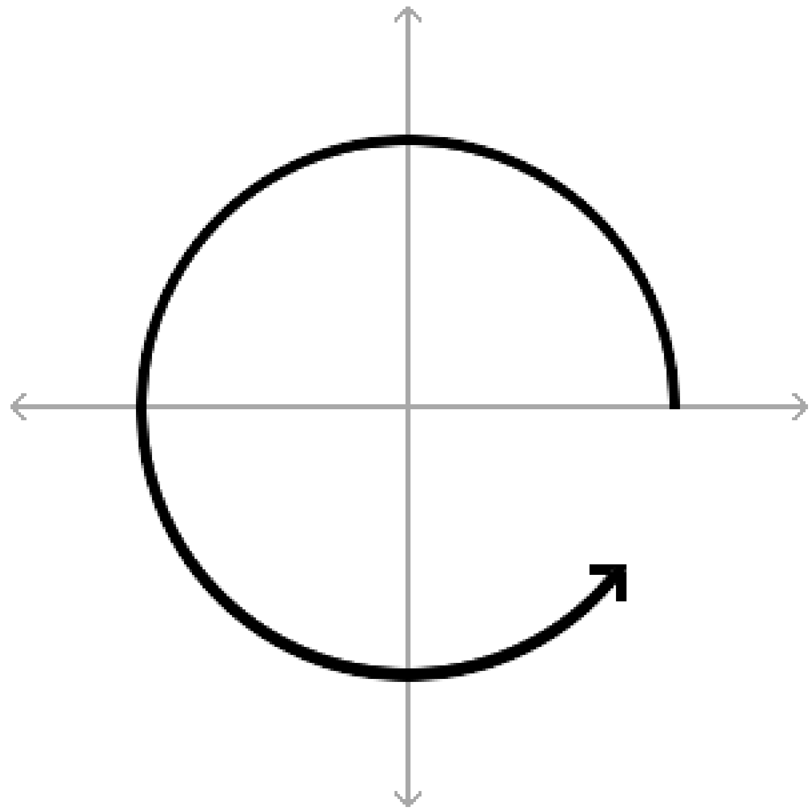
Complex numbers in finite fields

$\sqrt{-1} \bmod 5$?

$-1 \equiv 4$

$\sqrt{-1} \equiv 2$

The circular function

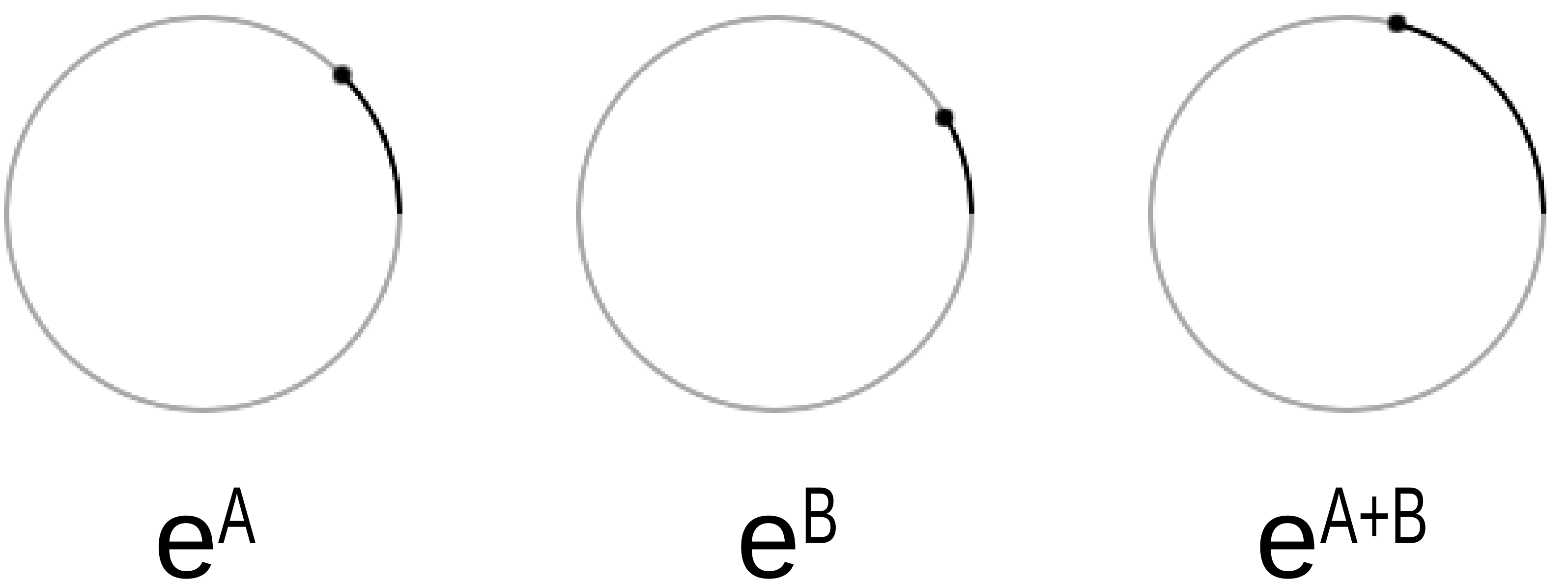


For real (non-discrete) values, $e^{i\theta}$ parameterizes a unit circle in the complex plane.

$$e^{i\theta} = \cos\theta + i \sin\theta$$

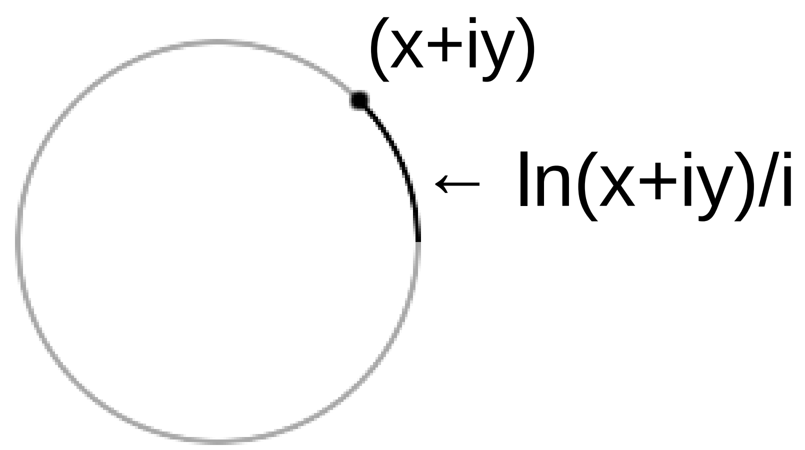
Multiplying complex numbers is equivalent to adding arcs

$$e^A e^B = e^{A+B}$$



Arc length of a circle

The complex natural logarithm gives the arc length, or distance along the circumference of a unit circle



Trigonometric identities

$$\cos(A+B) = (\cos A)(\cos B) - (\sin A)(\sin B)$$

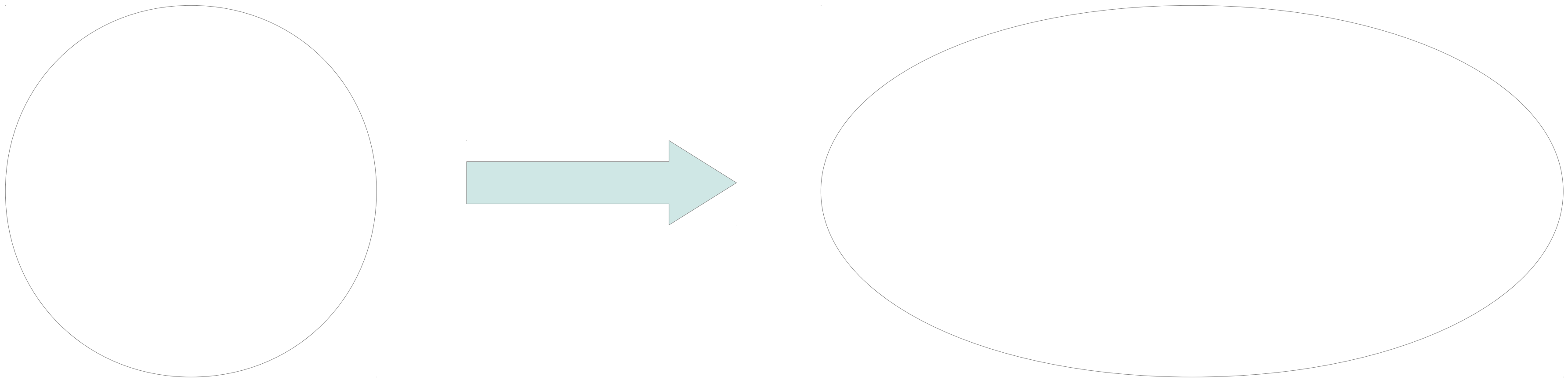
$$\sin(A+B) = (\sin A)(\cos B) + (\cos A)(\sin B)$$

This is equivalent to multiplying complex numbers.

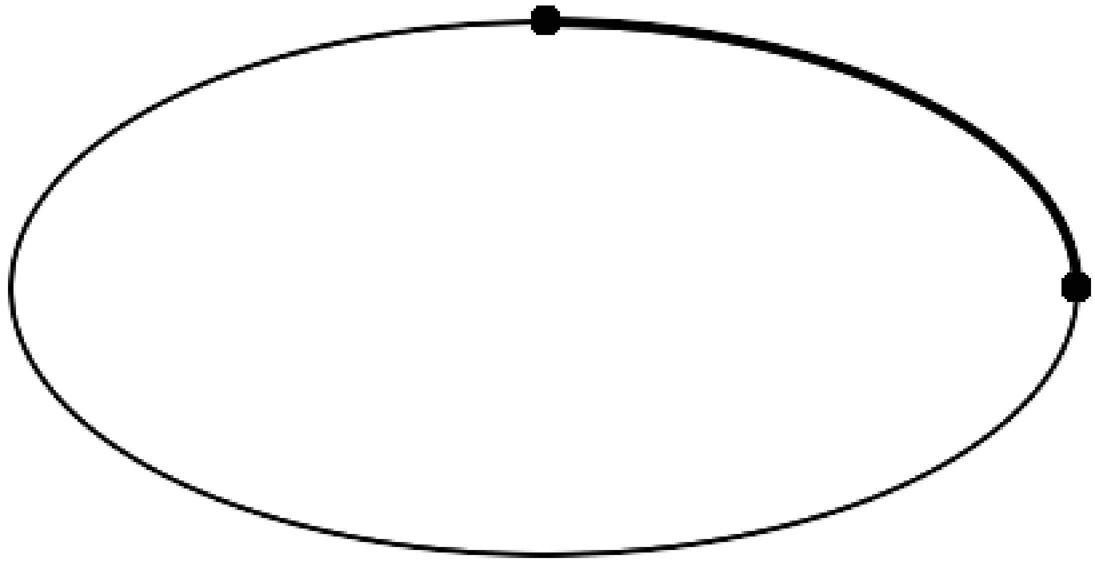
$$\Re(A\times B) = \Re(A)\Re(B) - \Im(A)\Im(B)$$

$$\Im(A\times B) = \Im(A)\Re(B) + \Re(A)\Im(B)$$

What about an ellipse instead of a circle?



Arc length of an ellipse



Unlike a circle, the arc length of an ellipse can not, in general, be expressed in closed form.

What function can be used to parameterize an ellipse using the distance along its circumference?

Weierstrass $\wp$ function

Defined by a differential equation

$$[\wp'(z)]^2 = 4[\wp(z)]^3 - g_2\wp(z) - g_3$$

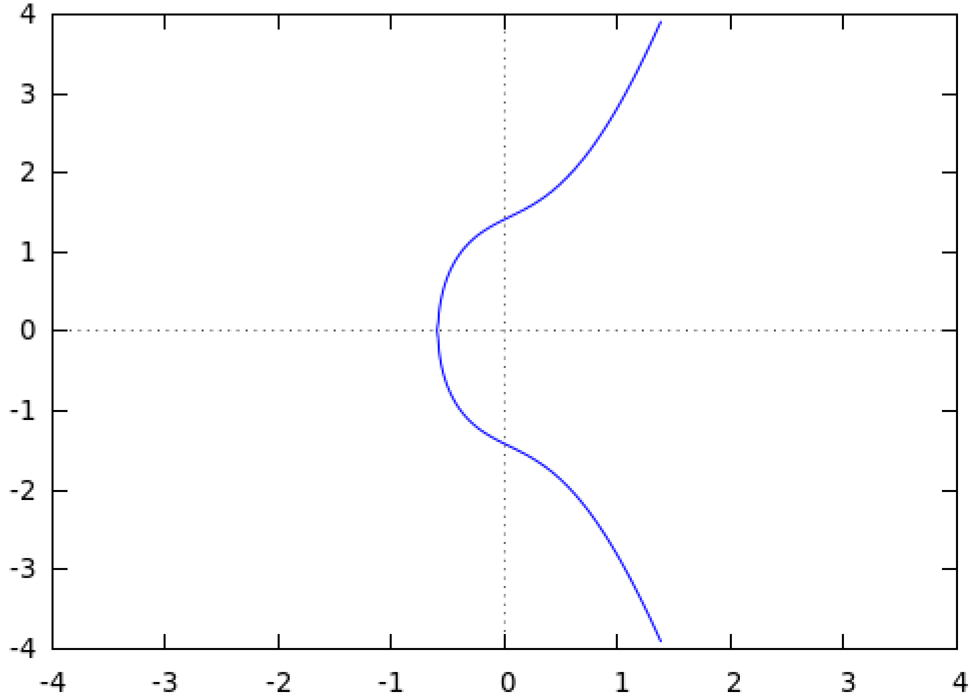
Karl Weierstrass
1815 - 1897

Parameterizing the $\wp$ function

$$X = \wp(z)$$
$$Y = \wp'(z)$$

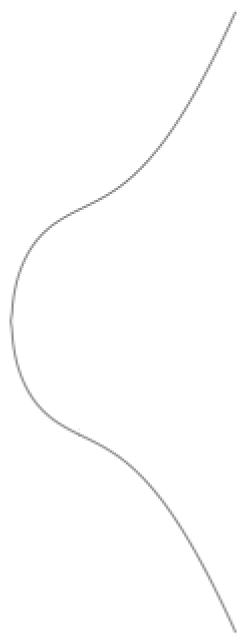
$$y^2 = 4x^3 + ax + b$$

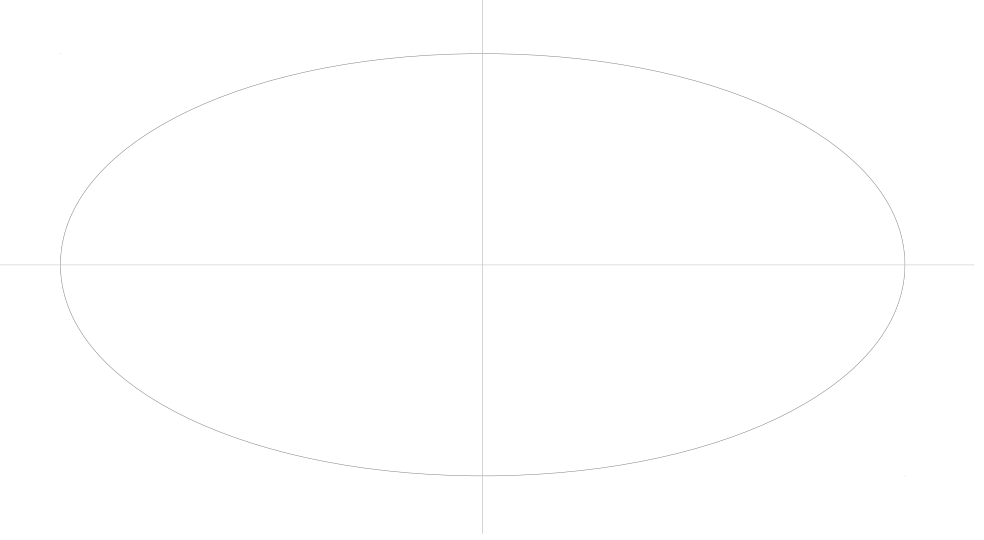
Example elliptic curve

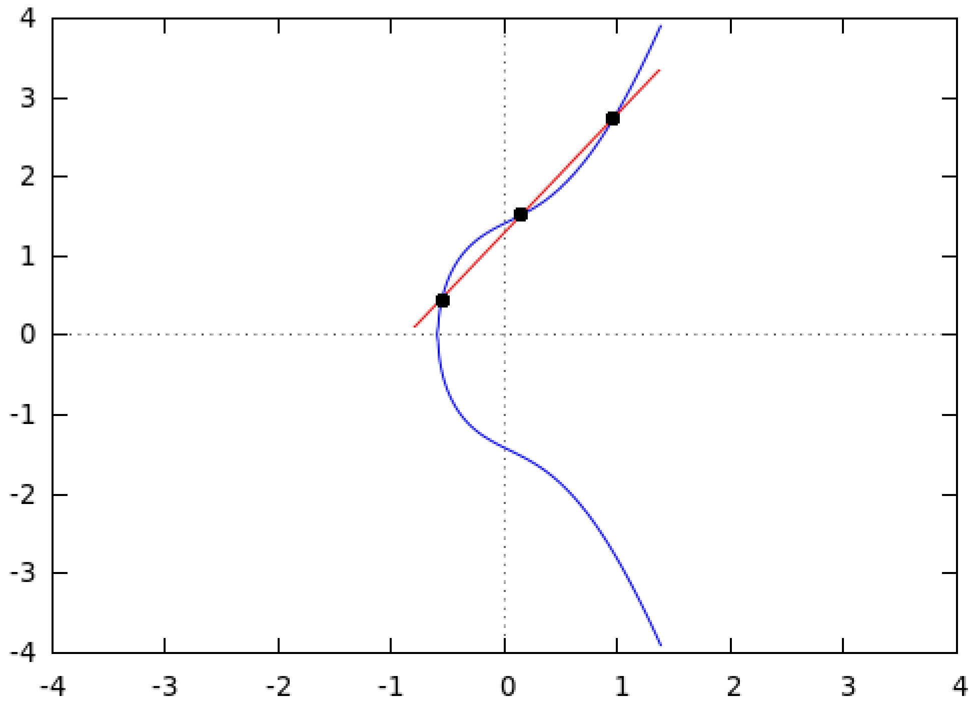


$y^2 = 4x^3 + 2x + 2$

$$\wp \propto \frac{1}{y^2}$$







If $[\wp(a),\wp'(a)]$, $[\wp(b),\wp'(b)]$, $[\wp(c),\wp'(c)]$ are colinear,

$$a + b + c = 0 \pmod{\omega/2}$$

Attacks on Elliptic Curve Crypto

Transform elliptic curve discrete logarithm problem to ordinary discrete logarithm problem

Anomalous elliptic curves with p points in prime field p

Menezes-Okamoto-Vanstone (MOV) attack, for supersingular elliptic curves

Frey-Rueck attack, for non-prime fields (p^n)

SSH Key Exchange

- curve25519-sha256: ECDH over Curve25519 (mod 2^{255}-19) with SHA2
- ecdh-sha2-nistp256: ECDH over NIST P-256 with SHA2
- ecdh-sha2-nistp384: ECDH over NIST P-384 with SHA2
- ecdh-sha2-nistp521: ECDH over NIST P-521 with SHA2
- diffie-hellman-group-exchange-sha256: Custom DH with SHA2
- diffie-hellman-group-exchange-sha1: Custom DH with SHA1
- diffie-hellman-group14-sha1: 2048 bit DH with SHA1
- diffie-hellman-group1-sha1: 1024 bit DH with SHA1

SSL/TLS Key Exchange

TLS_RSA
TLS_DH_DSS
TLS_DH_RSA
TLS_DHE_DSS
TLS_DHE_RSA
TLS_ECDH_ECDSA
TLS_ECDH_RSA
TLS_ECDHE_ECDSA
TLS_ECDHE_RSA