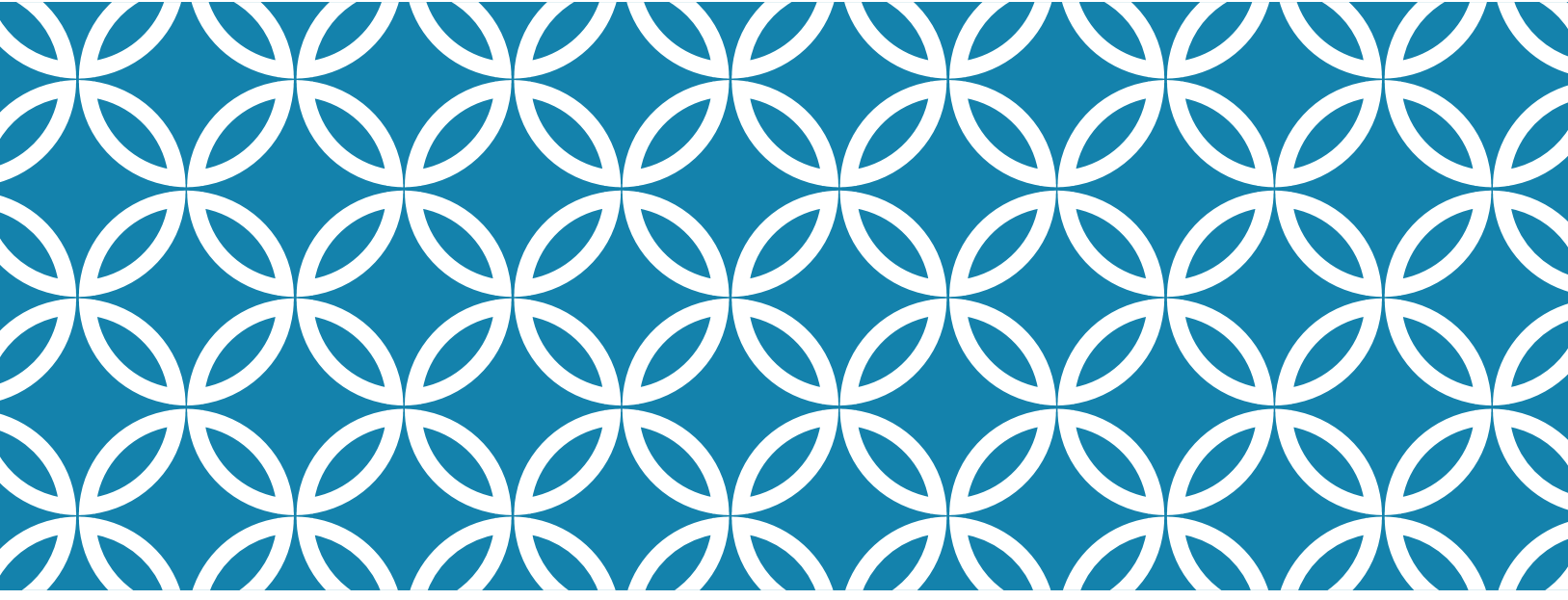


ELLIPTIC CURVES CRYPTOGRAPHY

WHAT ARE ELLIPTIC CURVES?

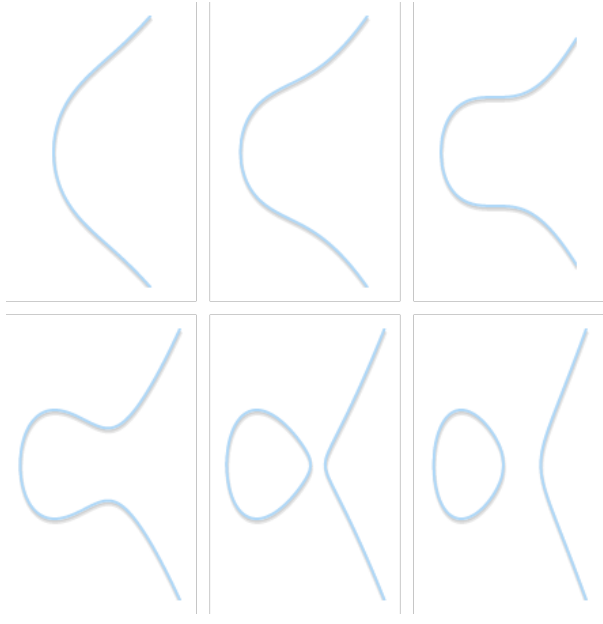
an elliptic curve will simply be **the set of points described by the equation:**

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

where

$$4a^3+27b^2\neq 0$$

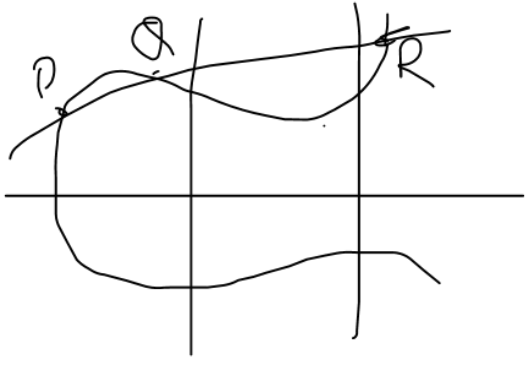
EXAMPLES



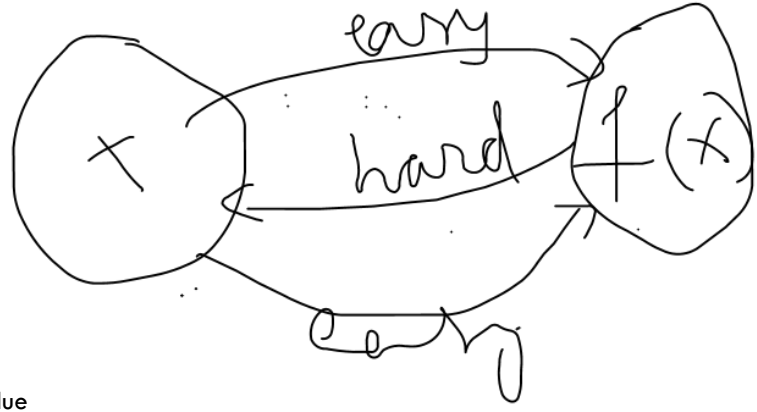
PROPERTIES OF ELLIPTIC CURVES

Symmetric on X-Axis

A line drawn across the curve will meet the curve only at 3 points



TRAP DOOR FUNCTION



Given “t” trapdoor value

GROUPS

A group in mathematics is a set for which we have defined a binary operation that we call "addition" and indicate with the symbol +.

In order for the set GG to be a group, addition must defined so that it respects the following four properties:

GROUP PROPERTIES

closure: if a and b are members of G, then $a+b$ is a member of G;

associativity: $(a+b)+c=a+(b+c)$;

there exists an **identity element** 0 such that $a+0=0+a=a$;

every element has an **inverse,** that is: for every a there exists b such that $a+b=0$.

If we add a fifth requirement:

commutativity: $a+b=b+a$,

then the group is called *abelian group*.

GROUPS OVER ELLIPTIC CURVES

We can define a group over elliptic curves. Specifically:

the elements of the group are the points of an elliptic curve;

the **identity element** is the point at infinity 0;

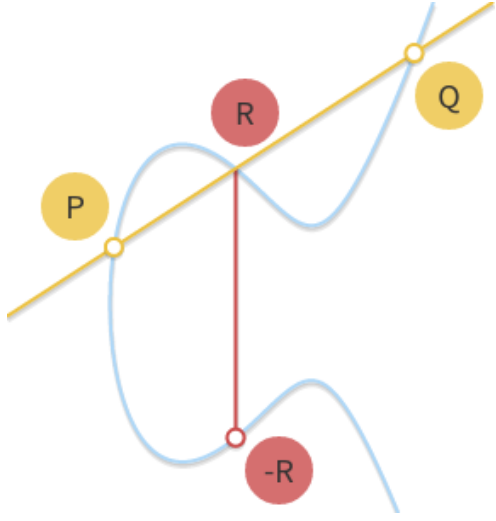
the **inverse** of a point *P* is the one symmetric about the x-axis;

addition is given by the following rule:

given three aligned, non-zero points *P*, Q **and** *R*,

their sum is $P+Q+R=0$

SUM AND INVERSE



FINITE FIELDS

A finite field is a set with a finite number of elements.

An example of finite field is the set of integers modulo p, where p is a prime number.

It is generally denoted as Z/p, $GF(p)$) or F_p We will use the latter notation.

FINITE FIELDS

In fields we have two binary operations: addition (+) and multiplication (·).

Both are closed, associative and commutative.

For both operations, there exist a unique identity element,

and for every element there's a unique inverse element.

Finally, multiplication is distributive over the addition: $x \cdot (y+z) = x \cdot y + x \cdot z$.

MULTIPLICATION OVER POINTS IN ELLIPTIC CURVES

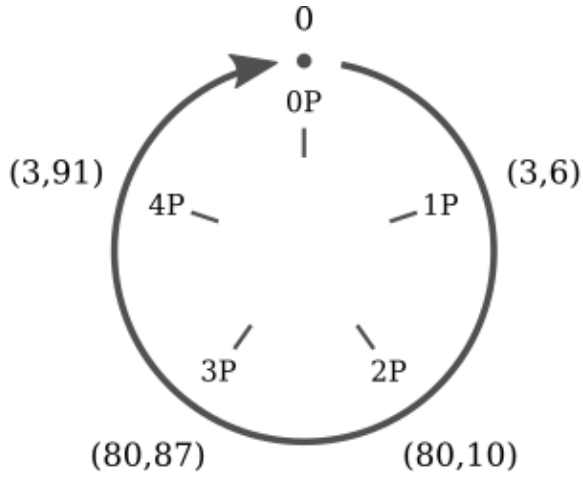
Multiplication over points for elliptic curves in F_p has an interesting property.

Take the curve

$y_2 \equiv x_3 + 2x + 3 \pmod{97}$ and the point $P=(3,6)$.

Now calculate the multiples of P

CYCLIC BEHAVIOR



CYCLIC SUBGROUPS AND BASE POINTS

A "subgroup" is a group which is a subset of another group.

A "cyclic subgroup" is a subgroup which elements are repeating cyclically,

The point *P* is called **generator** or **base point** of the cyclic subgroup.

ORDER

a cyclic subgroup we can give a new, equivalent definition:

the order of P is the smallest positive integer n such that $nP=0$.

In fact, if you look at the previous example, our subgroup contained five points, and we had $5P=0$.

The order of a subgroup is a divisor of the order of the parent group.

In other words, if an elliptic curve contains N points and one of its subgroups contains n points, then n is a divisor of N.

FINDING THE BASE POINT

In the light of this, we can outline the following algorithm:

Calculate the order N of the elliptic curve.

Choose the order n of the subgroup. For the algorithm to work, this number must be prime and must be a divisor of N .

Compute the cofactor $h=N/n$.

Choose a random point P on the curve.

Compute $G=hP$.

If G is 0, then go back to step 4. Otherwise we have found a generator of a subgroup with order n and cofactor h .

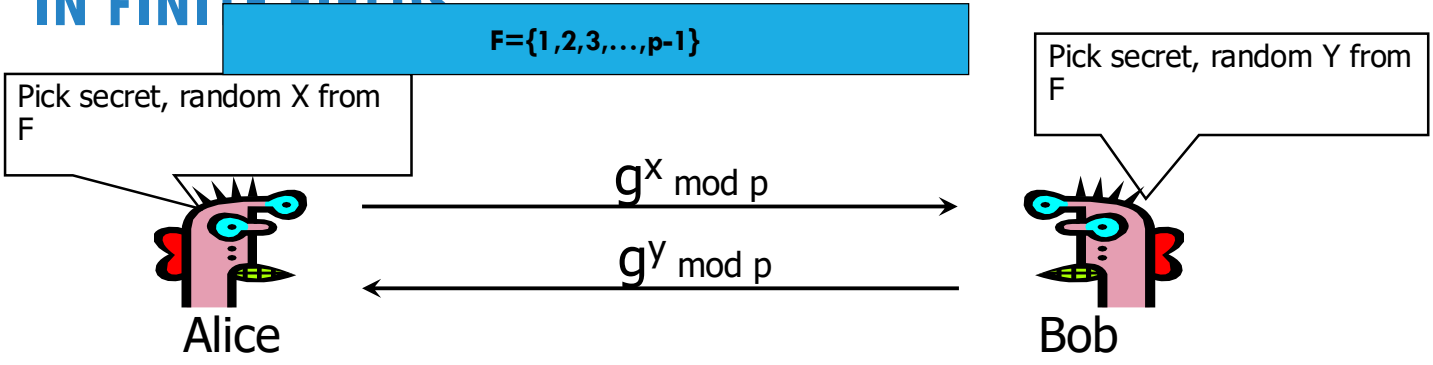
Note that this algorithm only works if n is a prime. If n wasn't a prime, then the order of GG could be one of the divisors of n .

ELLIPTIC CURVES IN CRYPTOGRAPHY

Elliptic Curve (EC) systems as applied to cryptography were first proposed in 1985 independently by Neal Koblitz and Victor Miller.

The discrete logarithm problem on elliptic curve groups is believed to be more difficult than the corresponding problem in (the multiplicative group of nonzero elements of) the underlying finite field.

DISCRETE LOGARITHMS IN FINITE FIELDS



Compute $k=(g^y)^x=g^{xy}$ mod p

Compute $k=(g^x)^y=g^{xy}$ mod p

Eve has to compute g^{xy} from g^x and g^y without knowing x and y…
She faces the Discrete Logarithm Problem in finite fields

DEFINITION OF ELLIPTIC CURVES

An elliptic curve over a field ***K*** is a nonsingular cubic curve in two variables, $f(x,y) = 0$ with a rational point (which may be a point at infinity).

The field ***K*** is usually taken to be the complex numbers, reals, rationals, algebraic extensions of rationals, p-adic numbers, or a **finite field.**

Elliptic curves groups for cryptography are examined with the underlying fields of $\boldsymbol{F_p}$ (*where p>3 is a prime*) and $\boldsymbol{F_{2^m}}$ (*a binary representation with* 2^m *elements*).

WHAT IS ELLIPTIC CURVE CRYPTOGRAPHY (ECC)?

Elliptic curve cryptography [ECC] is a **public-key** cryptosystem just like RSA, Rabin, and El Gamal.

Every user has a **public** and a **private** key.

- Public key is used for encryption/signature verification.
- Private key is used for decryption/signature generation.

Elliptic curves are used as an extension to other current cryptosystems.

- Elliptic Curve Diffie-Hellman Key Exchange
- Elliptic Curve Digital Signature Algorithm

USING ELLIPTIC CURVES IN CRYPTOGRAPHY

The central part of any cryptosystem involving elliptic curves is the **elliptic group**.

All public-key cryptosystems have some underlying mathematical operation.

- RSA has exponentiation (raising the message or ciphertext to the public or private values)
- ECC has point multiplication (repeated addition of two points).

GENERIC PROCEDURES OF ECC

Both parties agree to some publicly-known data items

- The **elliptic curve equation**
  - values of ***a*** and ***b***
  - prime, ***p***
- The **elliptic group** computed from the elliptic curve equation
- A **base point**, B, taken from the elliptic group
  - Similar to the generator used in current cryptosystems

Each user generates their public/private key pair

- Private Key = an integer, x, selected from the interval [1, p-1]
- Public Key = product, Q, of private key and base point
  - (Q = x*B)

EXAMPLE – ELLIPTIC CURVE CRYPTOSYSTEM ANALOG TO EL GAMAL

Suppose Alice wants to send to Bob an encrypted message.

- Both agree on a base point, B.
- Alice and Bob create public/private keys.
  - Alice
    - Private Key = a
    - Public Key = $P_A = a * B$
  - Bob
    - Private Key = b
    - Public Key = $P_B = b * B$
- Alice takes plaintext message, M, and encodes it onto a point, P_M, from the elliptic group

EXAMPLE – ELLIPTIC CURVE CRYPTOSYSTEM ANALOG TO EL GAMAL

- Alice chooses another random integer, k from the interval [1, p-1]
- The ciphertext is a pair of points
  - $P_C = [\ (kB), (P_M + kP_B)\ ]$

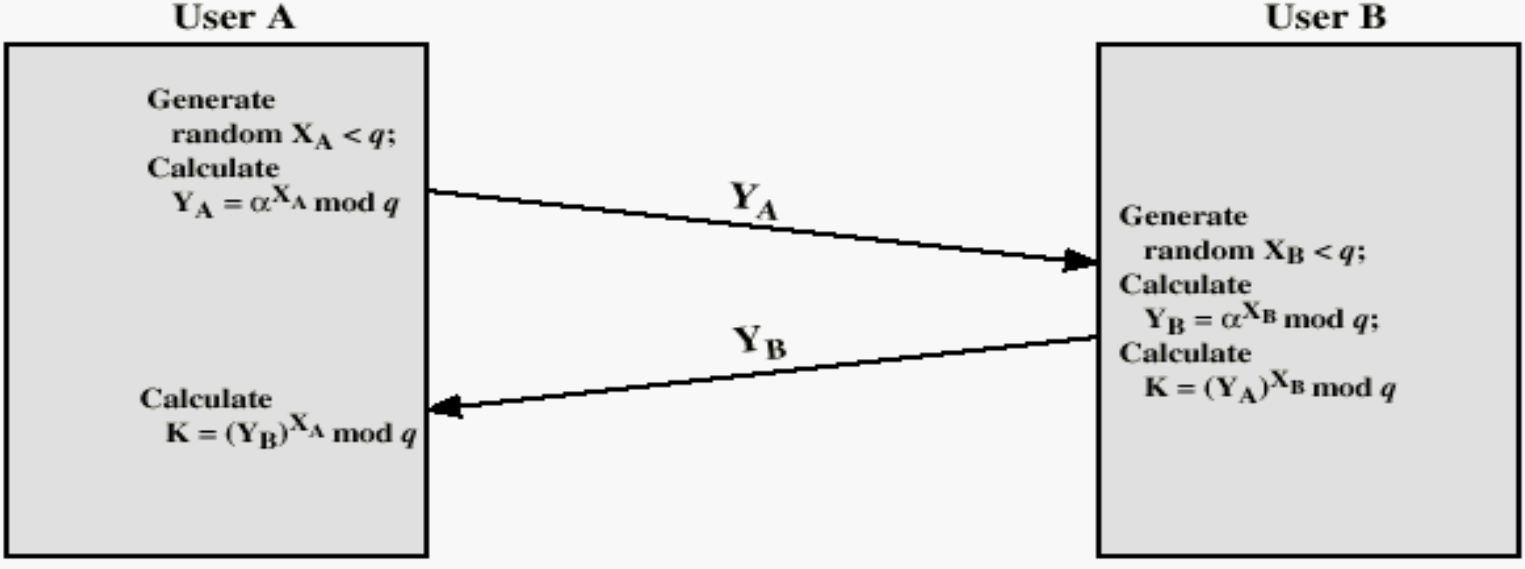
- To decrypt, Bob computes the product of the first point from P_C and his private key, b
  - $b * (kB)$
- Bob then takes this product and subtracts it from the second point from P_C
  - $(P_M + kP_B) - [b(kB)] = P_M + k(bB) - b(kB) = P_M$
- Bob then decodes P_M to get the message, M.

EXAMPLE – COMPARE TO EL GAMAL

- The ciphertext is a pair of points
  - $P_C = [\ (kB), (P_M + kP_B)\ ]$
- The ciphertext in El Gamal is also a pair.
  - $C = (g^k \bmod p, mP_B^k \bmod p)$

- Bob then takes this product and subtracts it from the second point from P_C
  - $(P_M + kP_B) – [b(kB)] = P_M + k(bB) – b(kB) = P_M$
- In El Gamal, Bob takes the quotient of the second value and the first value raised to Bob's private value
  - $m = mP_B^k / (g^k)^b = mg^{k*b} / g^{k*b} = m$

DIFFIE-HELLMAN (DH) KEY EXCHANGE



ECC DIFFIE-HELLMAN

Public: Elliptic curve and point B=(x,y) on curve

Secret: Alice's a and Bob's b

Alice, A

a(x,y)

b(x,y)

Bob, B

- Alice computes a(b(x,y))
- Bob computes b(a(x,y))
- These are the same since ab = ba

EXAMPLE – ELLIPTIC CURVE DIFFIE-HELLMAN EXCHANGE

Alice and Bob want to agree on a shared key.

- Alice and Bob compute their public and private keys.
  - Alice
    - Private Key = a
    - Public Key = $P_A = a * B$
  - Bob
    - Private Key = b
    - Public Key = $P_B = b * B$
- Alice and Bob send each other their public keys.
- Both take the product of their private key and the other user's public key.
  - Alice → $K_{AB} = a(bB)$
  - Bob → $K_{AB} = b(aB)$
  - **Shared Secret Key = $K_{AB} = abB$**

WHY USE ECC?

How do we analyze Cryptosystems?

- How difficult is the underlying problem that it is based upon
  - RSA – Integer Factorization
  - DH – Discrete Logarithms
  - ECC - Elliptic Curve Discrete Logarithm problem
- How do we measure difficulty?
  - We examine the algorithms used to solve these problems

SECURITY OF ECC

To **protect** a 128 bit AES key it would take a:

- RSA Key Size: 3072 bits
- ECC Key Size: 256 bits

How do we strengthen RSA?

- Increase the key length

Impractical?

NIST guidelines for public key sizes for AES

ECC KEY SIZE (Bits)	RSA KEY SIZE (Bits)	KEY SIZE RATIO	AES KEY SIZE (Bits)
163	1024	1 : 6	
256	3072	1 : 12	128
384	7680	1 : 20	192
512	15 360	1 : 30	256

Supplied by NIST to ANSI X9F1

APPLICATIONS OF ECC

Many devices are small and have limited storage and computational power

Where can we apply ECC?

- **Wireless communication devices**
- Smart cards
- Web servers that need to handle many encryption sessions
- **Any application where security is needed but lacks the power, storage and computational power that is necessary for our current cryptosystems**

BENEFITS OF ECC

Same benefits of the other cryptosystems: confidentiality, integrity, authentication and non-repudiation but…

Shorter key lengths

- Encryption, Decryption and Signature Verification speed up
- Storage and bandwidth savings

SUMMARY OF ECC

"**Hard problem**" analogous to discrete log

- **Q=kP, where Q, P belong to a prime curve**
  given k, P → "easy" to compute Q
  given Q, P → "hard" to find k
- **known as the elliptic curve logarithm problem**
  - **k must be large enough**

ECC security relies on elliptic curve logarithm problem

- compared to factoring, can use much smaller key sizes than with RSA etc

→ **for similar security ECC offers significant computational advantages**