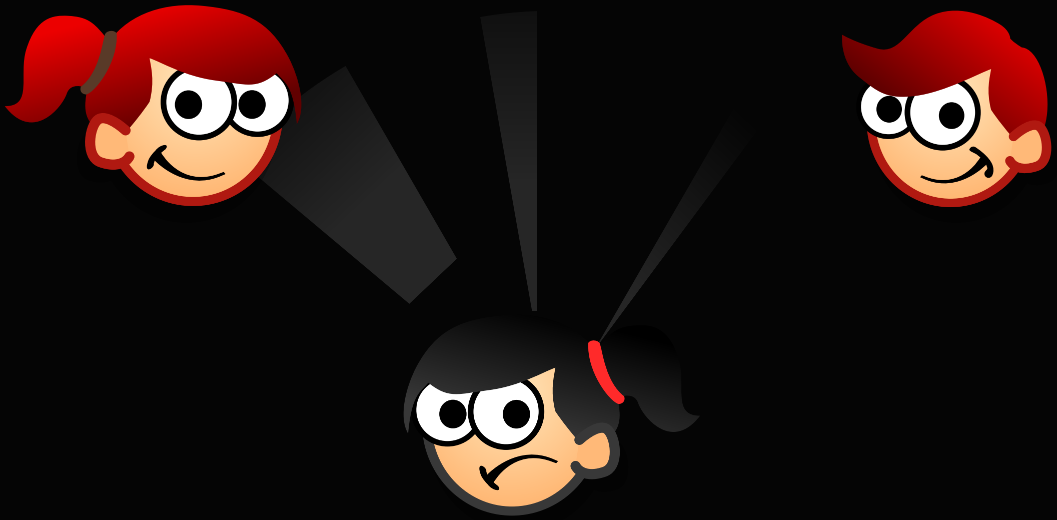
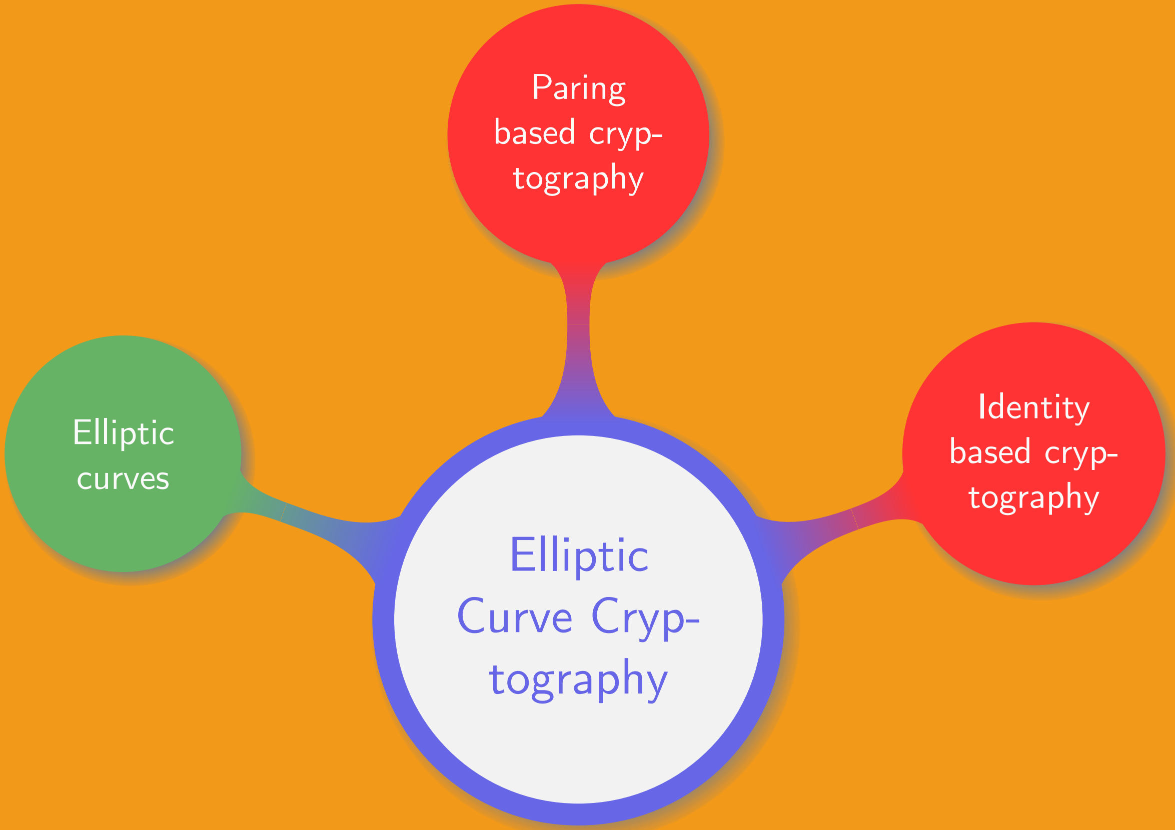


Introduction to Cryptography

8. Elliptic Curve Cryptography

Manuel – Summer 2020

Chapter organisation



Elliptic Curve Cryptography

Elliptic curves

Paring based cryptography

Identity based cryptography

Complexity and security

Main public key cryptography problems:

- RSA problem (3.51)
- Discrete Logarithm Problem (3.71)

Complexity and security

Main public key cryptography problems:

- RSA problem (3.51)
- Discrete Logarithm Problem (3.71)

Both problems can be solved using algorithms with sub-exponential complexity; that is algorithm with complexity neither polynomial nor exponential but somewhere in-between.

Consequence on the key size:

Security level (bits)	80	112	128	192	256
Key size (bits)	1024	2048	3072	7680	15360

Toward Algebra and Geometry

In chapter 3 it was noted that Pollard's rho was a generic algorithm solving the DLP (remark 3.75). In contrast with more efficient algorithms, such as the NFS, Pollard's rho algorithm does not take advantage of the underlying structure of the group.

Therefore a simple idea for the DLP consists in finding a group where no algorithm performs better than Pollard-rho (3.73).

Toward Algebra and Geometry

In chapter 3 it was noted that Pollard's rho was a generic algorithm solving the DLP (remark 3.75). In contrast with more efficient algorithms, such as the NFS, Pollard's rho algorithm does not take advantage of the underlying structure of the group.

Therefore a simple idea for the DLP consists in finding a group where no algorithm performs better than Pollard-rho (3.73).

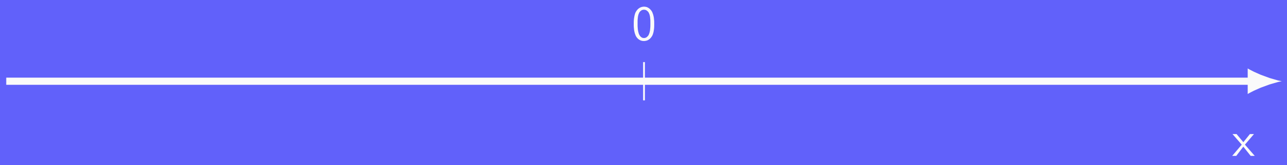
Abstract algebraic structures can be be studied from the perspective of geometry. A simple example is the group structure of the integers $(\mathbb{Z}, +)$ which can be represented on the number line.

Toward Algebra and Geometry

In chapter 3 it was noted that Pollard's rho was a generic algorithm solving the DLP (remark 3.75). In contrast with more efficient algorithms, such as the NFS, Pollard's rho algorithm does not take advantage of the underlying structure of the group.

Therefore a simple idea for the DLP consists in finding a group where no algorithm performs better than Pollard-rho (3.73).

Abstract algebraic structures can be be studied from the perspective of geometry. A simple example is the group structure of the integers $(\mathbb{Z}, +)$ which can be represented on the number line.

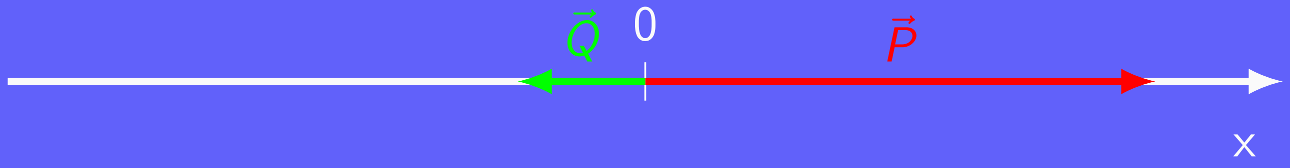


Toward Algebra and Geometry

In chapter 3 it was noted that Pollard's rho was a generic algorithm solving the DLP (remark 3.75). In contrast with more efficient algorithms, such as the NFS, Pollard's rho algorithm does not take advantage of the underlying structure of the group.

Therefore a simple idea for the DLP consists in finding a group where no algorithm performs better than Pollard-rho (3.73).

Abstract algebraic structures can be be studied from the perspective of geometry. A simple example is the group structure of the integers $(\mathbb{Z}, +)$ which can be represented on the number line.

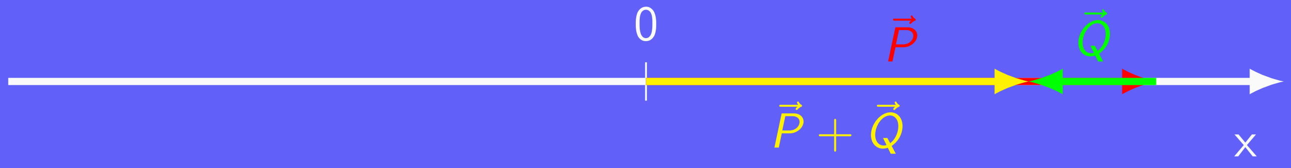


Toward Algebra and Geometry

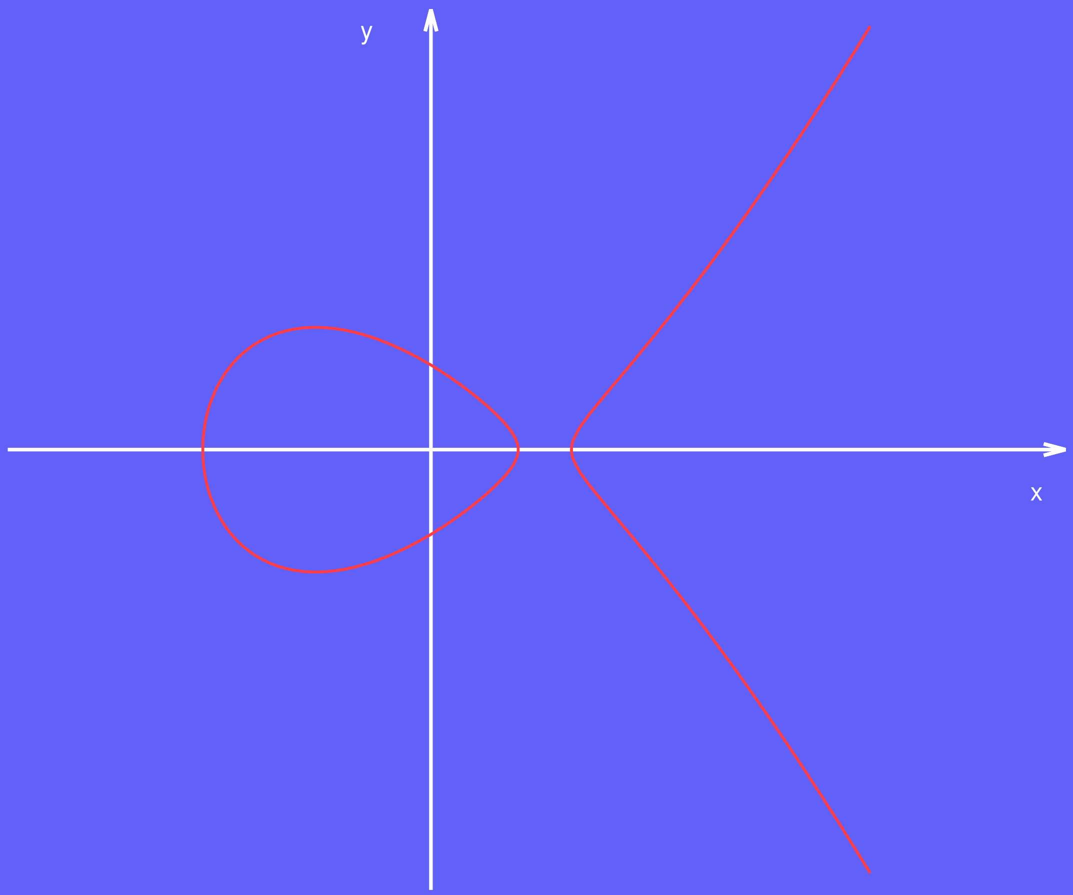
In chapter 3 it was noted that Pollard's rho was a generic algorithm solving the DLP (remark 3.75). In contrast with more efficient algorithms, such as the NFS, Pollard's rho algorithm does not take advantage of the underlying structure of the group.

Therefore a simple idea for the DLP consists in finding a group where no algorithm performs better than Pollard-rho (3.73).

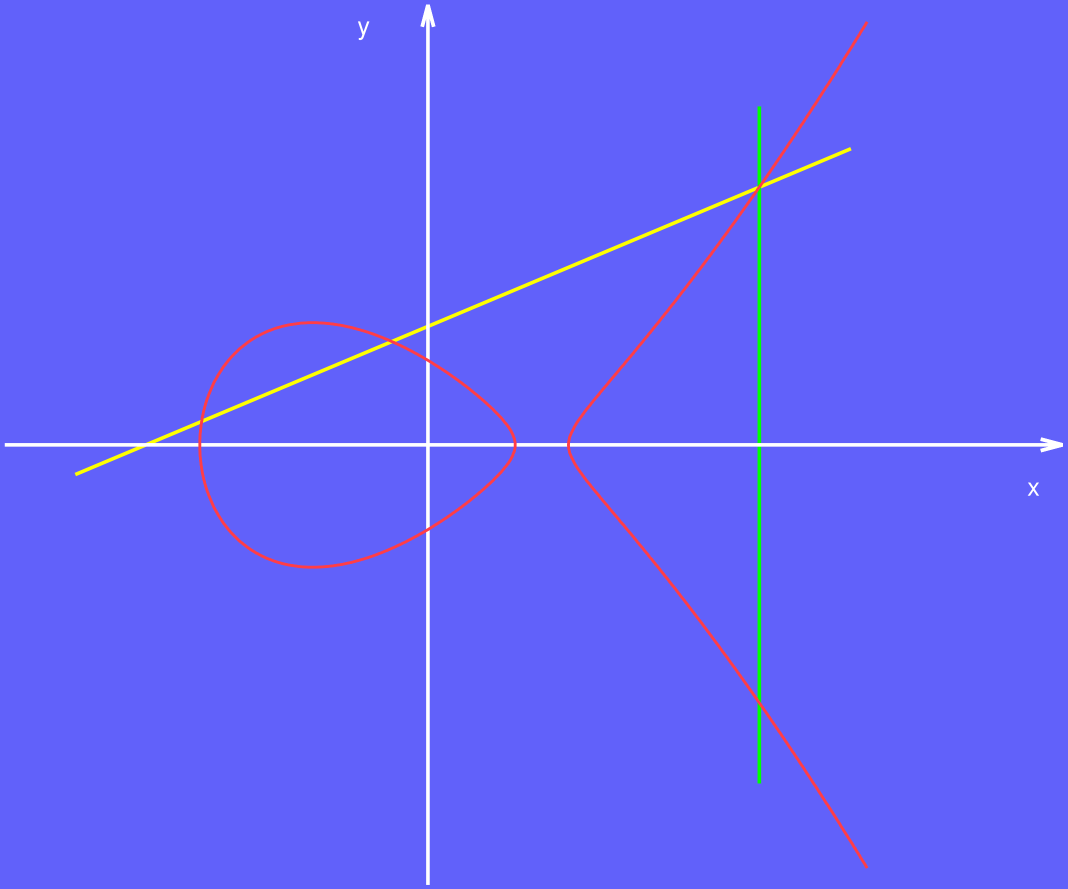
Abstract algebraic structures can be be studied from the perspective of geometry. A simple example is the group structure of the integers $(\mathbb{Z}, +)$ which can be represented on the number line.



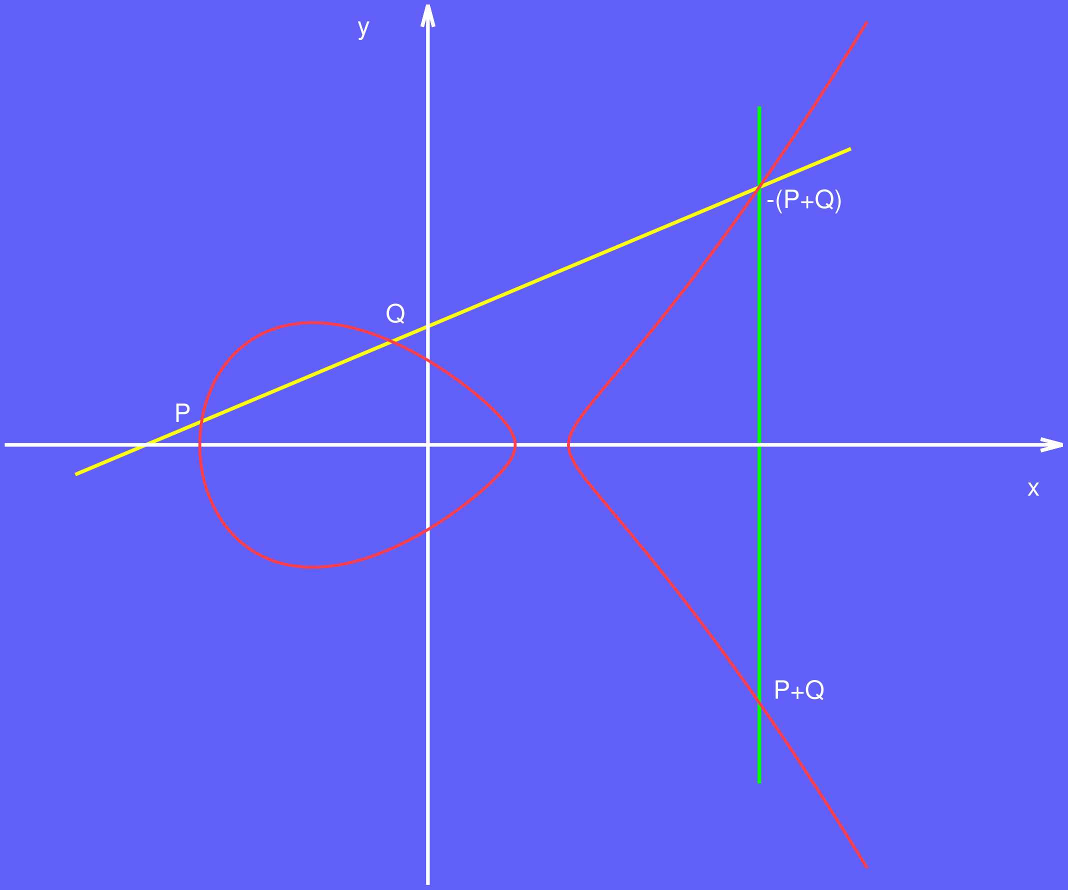
A new shape



A new shape



A new shape



Elliptic curves

The red curve:

- Is called an *elliptic curve*
- Is defined over a field, here the reals
- Can be defined over other fields
- Is given by the equation

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{8.1}$$

In most cases, a change of variable allows to rewrite equation (8.1) in the more simple form

$$y^2 = x^3 + bx + c$$

An elliptic curve as a group

By construction this is almost a group: only a unit element is missing. Therefore we adjoin the point $\mathcal{O}$, called *point at the infinity*. This point can be viewed as the point where all the vertical lines intersect.

Proposition

Let E be an elliptic curve of equation $y^2 = x^3 + bx + c$. Taking two point $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ on E, we define the addition law over E by $P_1 + P_2 = P_3 = (x_3, y_3)$ by

$$x_3 = m^2 - x_1 - x_2, \qquad y_3 = m(x_1 - x_3) - y_1,$$

with

$$m = \begin{cases} (y_2 - y_1)/(x_2 - x_1) & \text{if } P_1 \neq P_2 \\ (3x_1^2 + b)/(2y_1) & \text{if } P_1 = P_2. \end{cases}$$

This addition law is both associative and commutative. If taking $\mathcal{O}$ as unit element, then E is an abelian group.

Elliptic curves modulo p

Example.

Let E be the elliptic curve defined by $y^2 \equiv x^3 + 4x + 4 \bmod 5$. The points on E are all the pairs of elements (x, y) in $\mathbb{F}_5 \times \mathbb{F}_5$ that satisfy the equation.

x mod 5	y^2 mod 5	y mod 5	Points on E
0	4	2 or 3	(0,2) and (0,3)
1	4	2 or 3	(1,2) and (1,3)
2	0	0	(2,0)
3	3		
4	4	2 or 3	(4,2) and (4,3)

The elliptic curve E has eight points: seven calculated from the equation plus the point at the infinity $\mathcal{O}$.

Elliptic curve modulo p

Example.

We now determine the sum of the two points (1,2) and (4,3) on E.

First we note that as 3 is invertible mod 5 then

$$m = \frac{3-2}{4-1} \equiv 2 \bmod 5.$$

Then we compute x_3 and y_3,

$$\begin{aligned} x_3 &\equiv 2^2 - 1 - 4 &&\equiv 4 \bmod 5 \\ y_3 &\equiv 2(1-4) - 2 &&\equiv 2 \bmod 5. \end{aligned}$$

Finally we have $(1,2) + (4,3) = (4,2)$ on E.

Number of points modulo p

Simple strategy to count points on any elliptic curve mod p:

- Compute $t = x^3 + bx + c$ for $0 \leq x \leq p - 1$
- If t is square then $(x, \sqrt{t})$ and $(x, -\sqrt{t})$ are on E
- Approximately one over two values of t are squares
- An elliptic curve mod p has about p points

Number of points modulo p

Simple strategy to count points on any elliptic curve mod p:

- Compute $t = x^3 + bx + c$ for $0 \leq x \leq p - 1$
- If t is square then $(x, \sqrt{t})$ and $(x, -\sqrt{t})$ are on E
- Approximately one over two values of t are squares
- An elliptic curve mod p has about p points

Theorem (Hasse's theorem)

If E is an elliptic curve with n points, then

$$|n - p - 1| < 2\sqrt{p}.$$

Elliptic Curve Discrete Logarithm Problem

Since an elliptic curve gives rise to a group structure it means that we can define a discrete logarithm problem on them.

Problem (Elliptic Curve Discrete Logarithm Problem (ECDLP))

Let E be an elliptic curve over a finite field $\mathbb{F}_q$, $q = p^n$ for some prime p and integer n, and P be a generator of the group. Given a point Q on the E, find k in $\mathbb{N}$ such that $[k]P = Q$, where $[k]P$ represent the operation of adding $k - 1$ times the point P to itself.

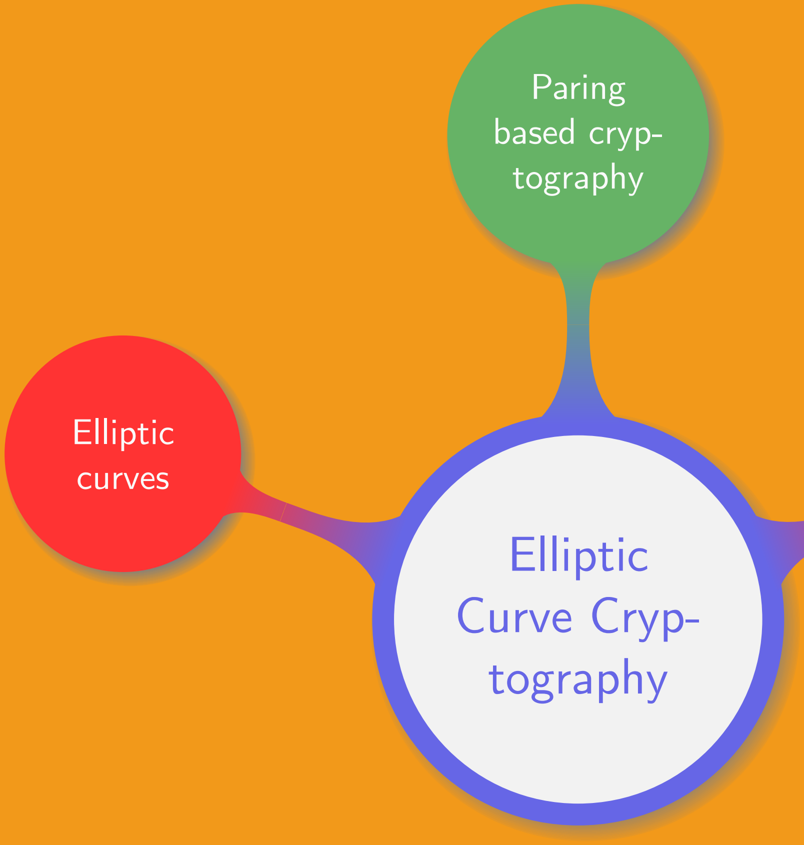
From a geometrical point of view it is clear that given k and P is it easy to find $[k]P$. However given Q and P is it hard to determine k such that $[k]P = Q$. Therefore the ECDLP allows the definition of a 1-way function.

Security of the ECDLP

In the general case, the best known algorithm to solve the ECDLP is Pollard's rho algorithm. From a cryptographic angle it means that the key size, in terms of bits, is only twice the security level.

Security level (bits)	Key size (bits)	
	DLP	ECDLP
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Chapter organisation



Elliptic Curve Cryptography

Elliptic curves

Paring based cryptography

Identity based cryptography

Solving the ECDLP

In remark 3.79 we noted that it was possible to map a hard instance of the DLP into an easier one, for instance in an additive group. Unfortunately this strategy is not practical since computing the map is too time consuming and as such would not provide an speedup.

The question now needs to be reconsidered in the case of elliptic curves. As mentioned earlier (8.12) the best algorithm to solve the ECDLP has exponential complexity. Therefore exhibiting a map from an elliptic curve into a subgroup of a finite field could bring much improvement in solving the ECDLP.

Although such maps exist only a few families of elliptic curves are vulnerable to this attack as in most cases the map is again to hard to compute. In the case where is can be efficiently computed it is called a *cryptographic pairing*.

Pairing

Definition

A *cryptographic paring* is a map e from two additive groups G_1 and G_2 into a multiplicative group G_T. For some given $P_1,\ P_2,\ P \in G_1$ and $Q_1,\ Q_2,\ Q \in G_2$ a pairing has the following properties:

- *Bilinearity:*

$$\begin{aligned} e(P, Q_1 + Q_2) &= e(P, Q_1)e(P, Q_2) \\ e(P_1 + P_2, Q) &= e(P_1, Q)e(P_2, Q), \end{aligned}$$

- *Non-degeneracy:*

$$\begin{aligned} &\forall P \in G_1,\ P \neq \mathcal{O} \quad \exists Q \in G_2 \text{ such that } e(P, Q) \neq 1 \\ &\forall Q \in G_2,\ Q \neq \mathcal{O} \quad \exists P \in G_1 \text{ such that } e(P, Q) \neq 1, \end{aligned}$$

- The map e is efficiently computable.

Pairings in practice

History of elliptic curves in cryptography:

- Discovered in the mid 80es
- In the 90es pairings were used to attack the ECDLP
- Then some families were abandoned since they were insecure
- Around 2000 pairings were used in a “constructive way”

Pairings in practice

History of elliptic curves in cryptography:

- Discovered in the mid 80es
- In the 90es pairings were used to attack the ECDLP
- Then some families were abandoned since they were insecure
- Around 2000 pairings were used in a "constructive way"

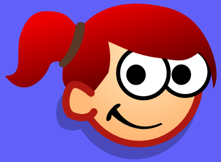
The most useful property of a pairing is bilinearity. It was realised that it could be used to construct new efficient protocols. We now describe one such example, due to Joux, where three parties can construct a common secret key in only one round.

Notations:

- For p a prime and n an integer, $q = p^n$
- An elliptic curve over $\mathbb{F}_q$, $E(\mathbb{F}_q)$
- A subgroup of $E(\mathbb{F}_q)$, $G = G_1 = G_2$

Tripartite key exchange protocol

Initial setup:



Common: G a subgroup of $E(\mathbb{F}_q)$, and P a generator of G

Personal: a secret key x_b (Bob), x_a (Alice), or x_c (Charly)

Tripartite key exchange protocol

Initial setup:

Common: G a subgroup of $E(\mathbb{F}_q)$, and P a generator of G
Personal: a secret key x_b (Bob), x_a (Alice), or x_c (Charly)

Key broadcasting:

Common: broadcast $Q_i = [x_i]P$, $a \leq i \leq c$
Personal: $e(Q_a, Q_c)^{x_b}$ (Bob), $e(Q_b, Q_c)^{x_a}$ (Alice), or $e(Q_a, Q_b)^{x_c}$ (Charly)

Tripartite key exchange protocol

Initial setup:

Common: G a subgroup of $E(\mathbb{F}_q)$, and P a generator of G

Personal: a secret key x_b (Bob), x_a (Alice), or x_c (Charly)

Key broadcasting:

Common: broadcast $Q_i = [x_i]P$, $a \leq i \leq c$

Personal: $e(Q_a, Q_c)^{x_b}$ (Bob), $e(Q_b, Q_c)^{x_a}$ (Alice), or $e(Q_a, Q_b)^{x_c}$ (Charly)

Shared secret key:

Common: $e([x_a x_b x_c]P, P) = e(P, P)^{x_a x_b x_c}$

Security of pairings

Security:

- x_a, x_b, and x_c must remain secret
- $e(P, P)^{x_a x_b x_c}$ must remain secret

Security of pairings

Security:

- x_a, x_b, and x_c must remain secret
- $e(P, P)^{x_a x_b x_c}$ must remain secret

Conclusion: both the DLP and the ECDLP must be secure

Efficiency:

- Pairings become more expensive as $E(\mathbb{F}_q)$ gets larger
- The group G_T is a subgroup of $\mathbb{F}_{q^k}$, for some integer k
- Arithmetic in $\mathbb{F}_{q^k}$ becomes more expensive as p, n, and k grow

Conclusion: balance both security and efficiency

Security of pairings

Security:

- x_a, x_b, and x_c must remain secret
- $e(P, P)^{x_a x_b x_c}$ must remain secret

Conclusion: both the DLP and the ECDLP must be secure

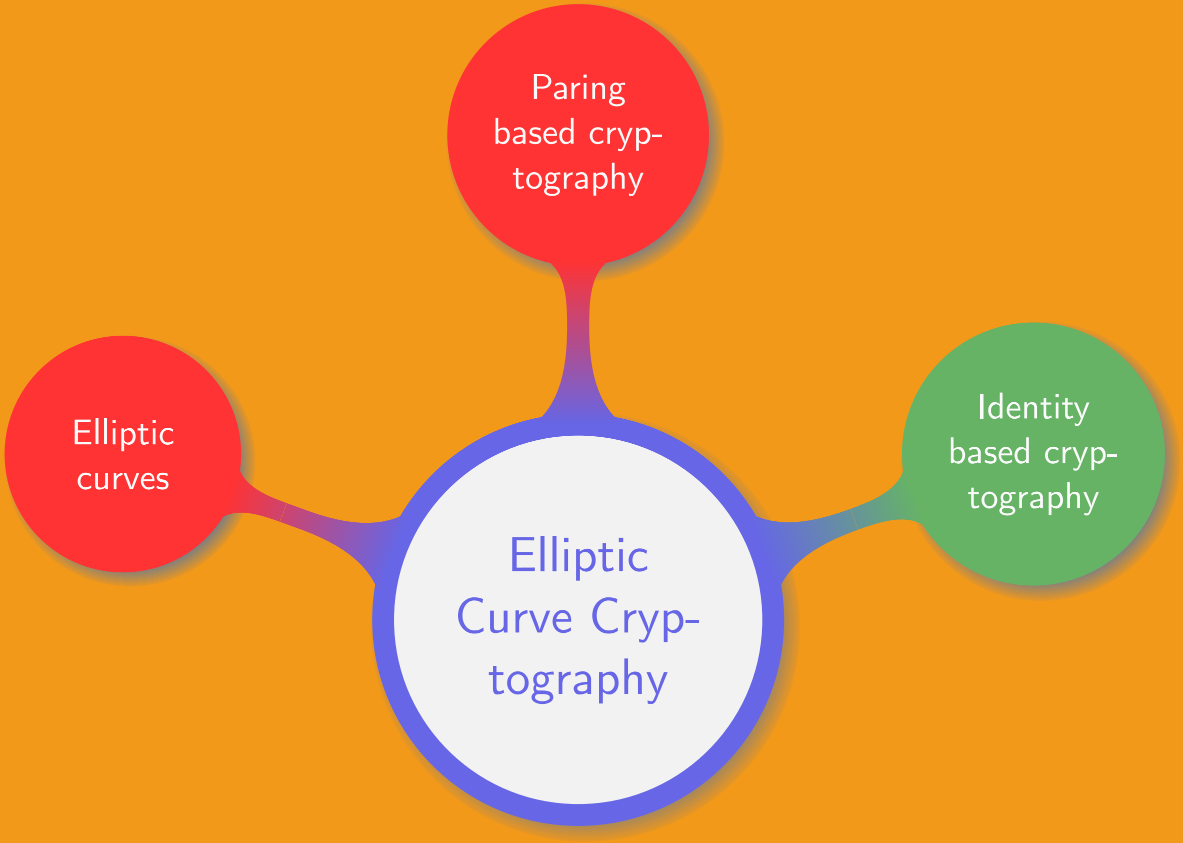
Efficiency:

- Pairings become more expensive as $E(\mathbb{F}_q)$ gets larger
- The group G_T is a subgroup of $\mathbb{F}_{q^k}$, for some integer k
- Arithmetic in $\mathbb{F}_{q^k}$ becomes more expensive as p, n, and k grow

Conclusion: balance both security and efficiency

Always ensure both the ECDLP and DLP have a similar security level

Chapter organisation



Paring based cryptography

Elliptic curves

Elliptic Curve Cryptography

Identity based cryptography

From PKC to IBC

All the protocols presented in chapter 3 require the use of a directory. This implies first that the user must register with a directory provider when he generates his keys but also that any other user who wants to communicate with should must connect to the directory in order to retrieve a public key.

An alternative, and more convenient solution, would be to use the identity of a user to automatically generate his public key. This would in turn eliminate the necessity of a directory.

Two common ways to solve this problem are to use pairings, or lattice based cryptography. We now present the first identity based protocol proposed by Boneh and Franklin in 2001.

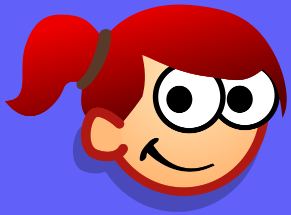
Initial setup

Trusted Authority (TA)

The TA prepares the system:

- Select an elliptic curve $E(\mathbb{F}_q)$
- Choose G, a subgroup of $E(\mathbb{F}_q)$, and P a generator of G
- Pick a random s and set $Q = [s]P$
- Choose a hash function H_1 mapping a string into a point in G
- Choose a hash function H_2
- For each user identity ID compute the secret key
$$s_{ID} = [s]H_1(ID)$$
- Public parameters: $\langle H_1, H_2, G, G_T, P, Q \rangle$

Encryption



Alice sends a message to Bob

Given a message m:

- Get Bob's ID, e.g. `bob@ve475.sjtu.edu.cn`
- Compute $g = e(H_1(\texttt{bob@ve475.sjtu.edu.cn}), Q)$
- Select a random r in $\mathbb{Z}_q^*$
- Compute $t = m \oplus H_2(g^r)$
- Send the ciphertext $C = \langle [r]P, t \rangle$

Decryption

Bob recovers Alice's message

Given a ciphertext $C = \langle [r]P, t \rangle$:

- Set ID to `bob@ve475.sjtu.edu.cn`
- Compute $h = e(s_{ID}, [r]P)$

$$\begin{aligned} h &= e([s]H_1(ID), [r]P) = e(H_1(ID), P)^{sr} \\ &= e(H_1(ID), [s]P)^r = e(H_1(ID), Q)^r \\ &= g^r \end{aligned}$$

- Recover the message as

$$t \oplus H_2(h) = m \oplus H_2(g^r) \oplus H_2(g^r) = m$$

Security

Basic remarks on the security:

- If s can be recovered then all the secret keys can be revealed
- If r can be computed from $[r]P$ then the message can be recovered
- The hash functions must be collision resistant
  e.g. $h = e(H_1(ID), [r]P)^s = g_p^s = g^r$. Neither s nor g^r is known but if we can find s' such that $H_2(g_p^{s'}) = H_2(g^r)$ then m can be recovered
- The TA must be trusted

Attribute based cryptography

Identity Based Cryptography: public key generated from an ID

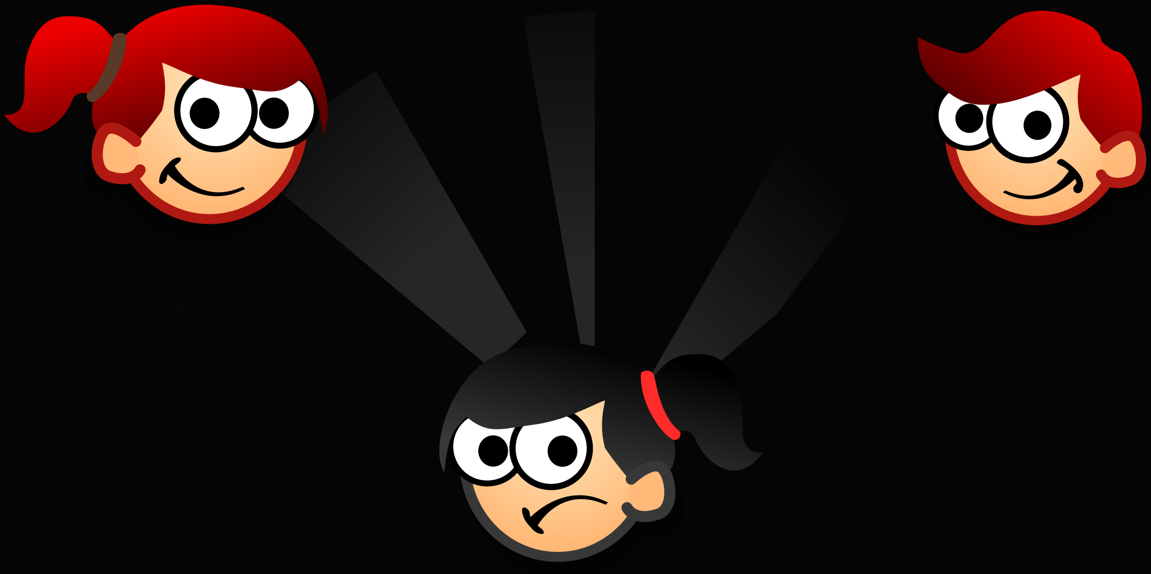
Attribute Based Cryptography: public key generated from attributes

Typical use:

- Company: a class of employees is sent some encrypted information; no need to encrypt using the public key of each employee
- Social media: people can belong to many different groups and want to share information only with a certain group without encrypting a special version for each member of the group
- Broadcast encryption: different class of users have payed for different services

Key points

- What is an elliptic curve?
- What is the main advantage of elliptic curves in cryptography?
- What is the most useful property of a pairing?
- Explain what Identity Based Cryptography and Attribute Based Cryptography are



Thank you!