Elliptic Curve Cryptography

Le chiffrement asymétrique version moderne

Sommaire

Préambule mathématique

- Fonction à trappe:
  - Multiplication entière != Factorisation -> RSA
  - Multiplication dans Z/nZ != Problème du logarithme discret -> Diffie-Hellman
  - Bases du chiffrement asymétrique

Préambule mathématique

- La théorie des groupes G=(E,•):
  - • loi de composition interne à E
  - Associativité de • : a•(b•c) = (a•b)•c
  - Existence d'un élément neutre e tel que (x•e)=x pour tout x dans E
  - Existence d'un unique inverse: (a•b) = (b•a) = e
  - Commutativité non obligatoire mais présente ici (a•b)=(b•a)

Préambule mathématique

- La théorie des groupes G=(E,•) (pt2):
  - Peuvent être fini ou non
  - Existence de sous-groupes
  - Soit f dans E, il existe un sous groupe minimal F de G qui contient f => f engendre F
  - |F| divise |G| si G fini
  - G est cyclique ⇔ il existe g dans G tel que g engendre G

- Un exemple très classique: Z/nZ aka F_n

Les courbes elliptiques

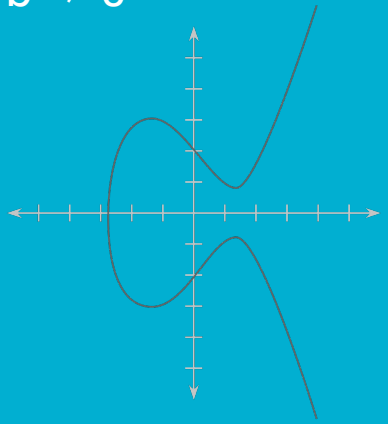
En tant qu'objet mathématique

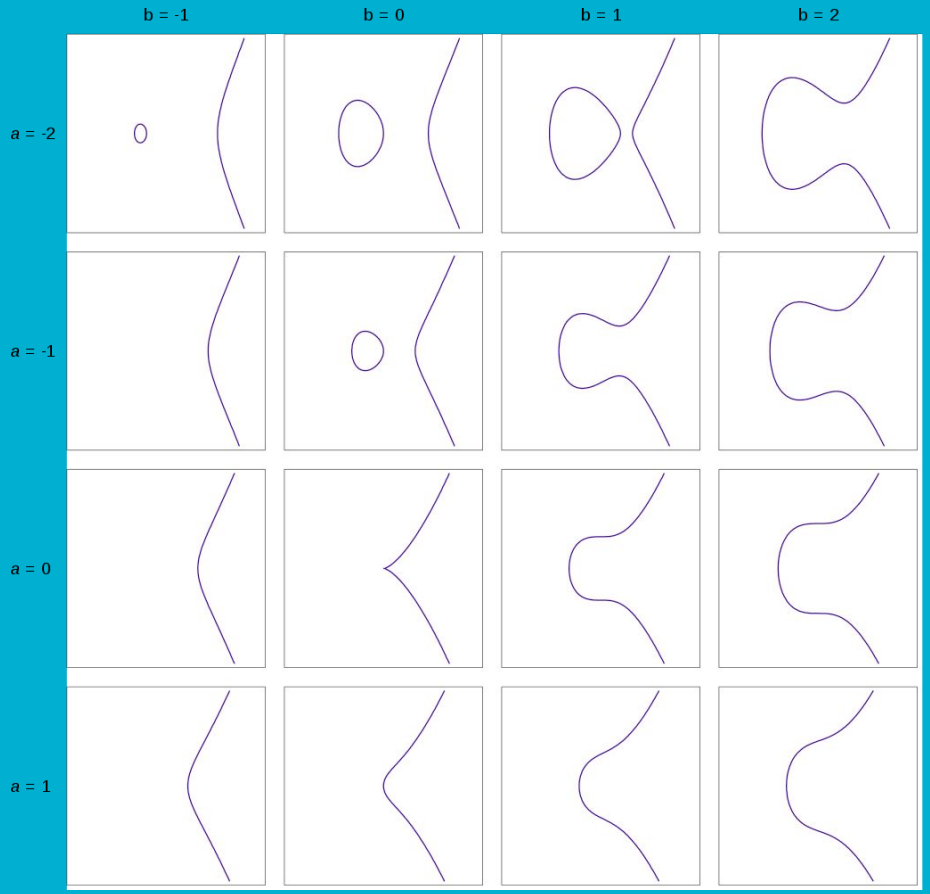
Un sujet d'actualité tant en cryptographie qu'en mathématiques et dans la théorie des nombres (th. de Fermat)

Les courbes

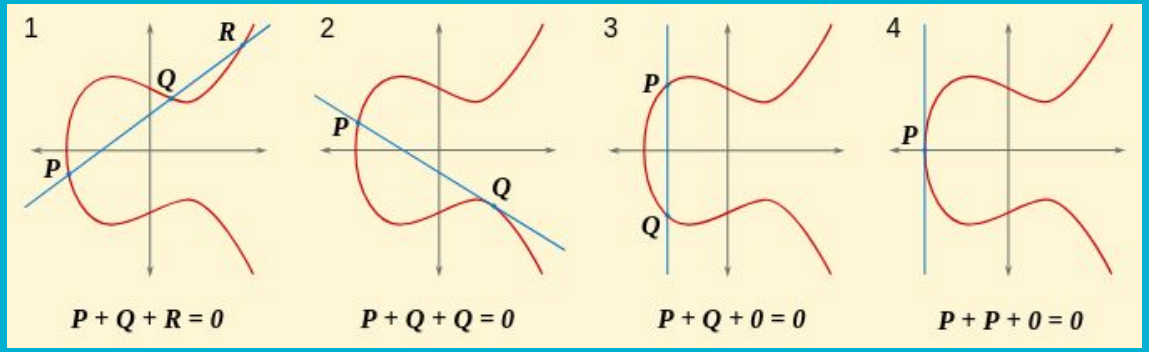
- $Y^2 = X^3 + aX + b$

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





L’addition



(a) If P = O, then P + Q = Q.

(b) Otherwise, if Q = O, then P + Q = P.

(c) Otherwise, write P = (x_1, y_1) and Q = (x_2, y_2).

(d) If $x_1 = x_2$ and $y_1 = -y_2$, then P + Q = O.

(e) Otherwise:

(e1) if P ≠ Q: $\lambda = (y_2 - y_1) / (x_2 - x_1)$

(e2) if P = Q: $\lambda = (3x_1^2 + a) / 2y_1$

(f) $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$

(g) P + Q = (x_3, y_3)

Un set de points

$E = \{(x,y) : x,y \in R \text{ tq } y^2 = x^3 + a\,x + b\}$ => les réels c'est pas pratique

$E = \{(x,y) : x,y \in N \text{ tq } y^2 = x^3 + a\,x + b\}$ => les entiers trop grands c'est pas pratique

$E(F_n) = \{(x,y) : x,y \in F_n \text{ tq } y^2 = x^3 + a\,x + b\}$ => Il manque le neutre

$E(F_n) = \{(x,y) : x,y \in F_n \text{ tq } y^2 = x^3 + a\,x + b\} \cup O$

Donc si $P \in E(F_n)$, on peut calculer $n*P$ avec $n*P \in E(F_n)$ de manière efficace (double and add)

Le protocole Diffie-Hellman

(version facile pour commencer)

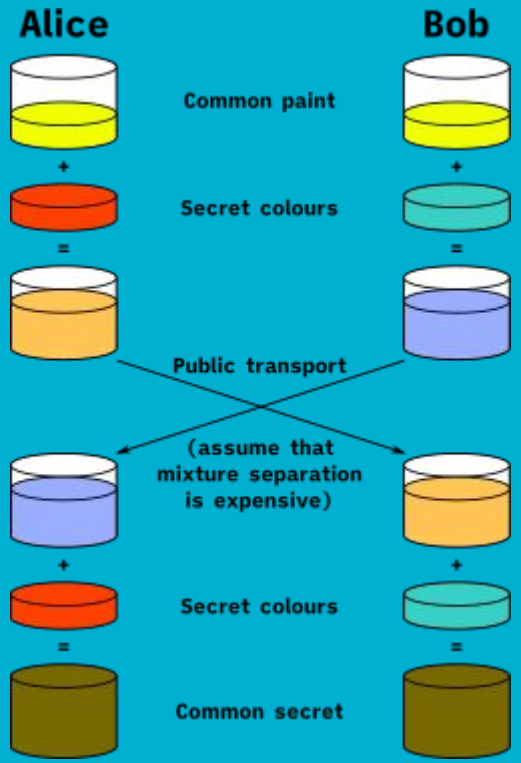
Révélé dans “New Directions in Cryptography”, W.Diffie, M.E.Hellman, 1976 => Prix Turing en 2015

L’échange de clé de Diffie-Hellman (DHKE)

Principe: établir une connection sécurisé sur un canal compromis

Fonction à trappe: Problème du logarithme discret

Principe


Alice
Bob
Common paint
Secret colours
Public transport
(assume that mixture separation is expensive)
Secret colours
Common secret

Concrètement

1) Alice et Bob se mettent d'accord et choisissent (g,n)=(3,17)
2) Alice choisit 9 comme secret et Bob 6
3) Alice calcule sa clé publique 3^9 mod 17 = 14. De même, Bob obtient 15
4) Alice envoie 14 à Bob, Bob envoie 15 à Alice
5) Alice calcule 15^9 mod 17 = 15. Bob calcule 14^6 mod 17 = 15
6) Alice et Bob possèdent un secret commun.

Elliptic Curve Diffie-Hellman Exchange

TLSv1.3

C’est l’heure de fusionner les deux parties précedentes

DHKE but different

On adapte l'algorithme qu'on vient de voir:

Problème du log discret : Trouver n / k = G^n mod p => Trouver n / k = nG dans $E(F_p)$

Le groupe: [[1,p]] => sous-groupe engendré par G

Complexité de l'inverse en O(√n)

Dans la pratique

1) Alice et Bob se mettent d'accord pour choisir une courbe E sur un champ F_p et un générateur G
2) Alice choisit sa clé privée n_A, et Bob la sienne (n_B)
3) Alice calcule sa clé publique $n_A G$, et l'envoie à Bob, qui fait de même et envoie $n_B G$ à Alice
4) Alice calcule $n_A * (n_B G)$, et Bob $n_B * (n_A G)$ qui sont égaux
5) Alice et Bob possèdent maintenant un secret commun

Standardisation

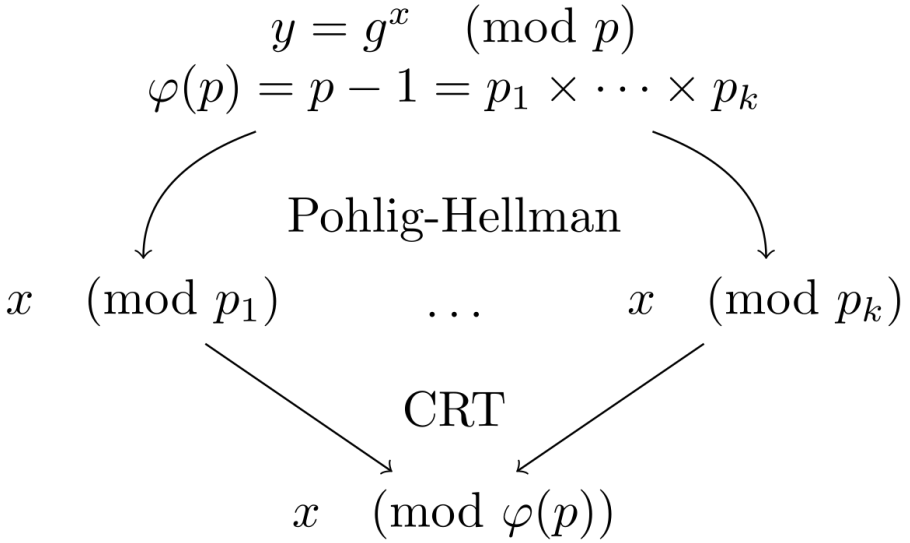
Les courbes ayant de “bonnes propriétés” sont nommées pour être identifié:

Ex: prime192v1

```
p    = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFFFFFFFFFF
a    = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFFFFFFFFFC
b    = 64210519E59C80E70FA7E9AB72243049FEB8DEECC146B9B1
s    = 3045AE6FC8422F64ED579528D38120EAE12196D5
x    = 188DA80EB03090F67CBF20EB43A18800F4FF0AFD82FF1012
y    = 07192B95FFC8DA78631011ED6B24CDD573F977A11E794811
q    = FFFFFFFFFFFFFFFFFFFFFFFF99DEF836146BC9B1B4D22831
h    = 1
```

Pohlig-Hellman

Attaquer DHKE



Baby Step Giant Step

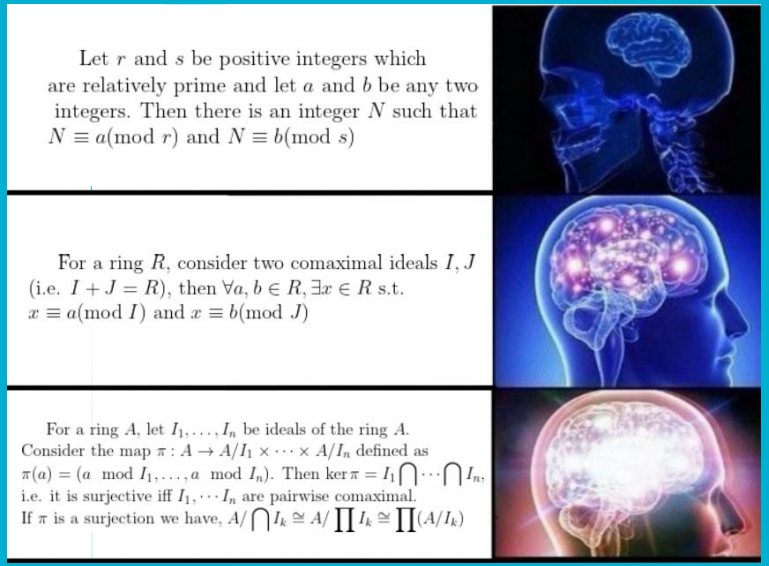
La force brute  Un bon compromis temps/mémoire (O(√n) dans les 2)

$$
\begin{aligned}
Q &= xP \\
Q &= (am + b)P \\
Q &= amP + bP \\
Q - amP &= bP
\end{aligned}
$$


baby steps
giant steps

CRT Chinese Remainder Theorem

- Soit x = a mod n1, x = b mod n2 et
- L'équation d'inconnue y: x = y mod ab possède une unique solution
- Généralisation immédiate par récurrence



Pohlig Hellman

Soit N l'ordre d'un sous groupe engendré par g. On cherche n tel que $k = g^n$

1. On décompose N en produit de facteurs premiers ($N = \prod pi^{ei}$)
2. On résout une équation approchée sur chaque pi^{ei} <= BSGS optimisé
3. On utilise le CRT pour rassembler nos morceaux d'équations


LES SOLUTIONS SUR CHAQUE PI^EI
LE PROTOCOLE
HTTPS QUI TOMBE