LA MOYENNE EN CRYPTO POUR LES NULS

Un livre de William ESCRIVA

basé sur les examens de 2020 et 2019

mon paypal : paypal@WillEsc

Les points importants:

LA CALCULATRICE!!!!!!

oui, j’ai oublié d'utiliser POWERMOD plein de fois… mais c’est plus compréhensible sans

Dans les deux examens les exercices sur les points suivants sont présent:

- Chiffrement par fonction affine
- Chiffrement RSA
- Chiffrement ElGamal

Les programmes/fonctions vitales dans la calculatrice:

- MODULO
- PGCD(a,b)
- INVMODULO(a,n) qui retourne x tel que (x = a^-1 mod n)
- **POWERMOD(a,p,n)** qui retourne x tel que (x = **a^p mod n**)
- Une liste des nombres premiers jusqu’à 149 avec les nombres premier sûr en évidence.

```
def inversemod(a,n):
    d,u,v = xgcd(a,n)
    if d==1: return u%n
    else: return None
```

```
def powermod(a,m,n):
    res = 1; x = a
    while m:
        if m & 1: res = (res*x) % n
        x = (x*x) % n
        m >>=1           # m //= 2
    return res
```

Liste nb premier et nb premier en gras (ceux de la forme 2p+1 avec p premier):
2, 3, **5**, **7**, **11**, 13, 17, 19, **23**, 29, 31, 37, 41, 43, **47**, 53, **59**, 61, 67, 71, 73, 79, **83**, 89, 97, 101, 103, **107**, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, **167**, 173, **179**, 181, 191, 193, 197, 199

mon paypal : paypal@WillEsc

Chiffrement Affine:

CHIFFREMENT AFFINE:

Il s'agit simplement de chiffrer une lettre x avec la fonction ax+b (mod n) = y ou y sera la lettre chiffré et n la taille de notre alphabet.

Pour l'alphabet normal on chiffre A = 0,B =1 … Z = 25, a et b vont prendre une valeur associée à une lettre.

Attention a doit être premier avec le nombre de lettres total!!! sinon des doublons vont apparaître.

pour l'alphabet normal a peut prendre 12 valeurs (1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25) et b 26 valeurs, ce qui nous fait 12 x 26 = 312 clés admissible.

DECHIFFREMENT AFFINE:

En possédant a et b on va devoir trouver a' tel que $x = a'(y-b) \pmod{n}$ pour un alphabet de taille n

Pour cela on va devoir utiliser INVMODULO.

En calculant $a' = INVMOD(a,n)$

ensuite on applique à la formule

$a'(y-n) \pmod{n} = a'y - a'b \pmod{n} = x$

mon paypal : paypal@WillEsc

CORRECTION EXERCICE 1 (2020-2021):

a) les valeurs possible pour b sont comprises dans [0;36]
   les valeurs possible pour a sont les nombres compris entre [1;36] premier avec 37
   $PGCD(1,37) = 1$
   $PGCD(2,37) = 1$
   $PGCD(3,37) = 1$ etc…
   REMARQUE : 37 est un nombre premier ! donc a va prendre les valeurs dans [1;36]

   Combien de possibilités? 36*37 = 1369 clés admissible

b) Cette fois ci 36 n'est pas premier, de plus tous les chiffres pairs vont être supprimés pour les valeurs possible de a.
   $PGCD(1,36) = 1$
   $PGCD(3,36) = 3$
   $PGCD(5,36) = 1$
   $PGCD(7,36) = 1$
   $PGCD(9,36) = 3$
   $PGCD(11,36) = 1$
   $PGCD(13,36) = 1$
   $PGCD(15,36) = 3$
   $PGCD(17,36) = 1$
   $PGCD(19,36) = 1$
   $PGCD(21,36) = 3$
   $PGCD(23,36) = 1$
   $PGCD(25,36) = 1$
   $PGCD(27,36) = 9$
   $PGCD(29,36) = 1$
   $PGCD(31,36) = 1$
   $PGCD(33,36) = 3$
   $PGCD(35,36) = 1$
   12 valeurs possible pour a et 36 valeurs possible pour b, soit 12*36=432 clés admissibles.

c) soit a = 5 et b = 3
   on calcule $a' = INVMOD(5,37) = 15$ la formule est donc
   $a'(y-b) \pmod{n} = 15 * (y-3) \pmod{37} = 15y - 45 \pmod{37}$

d) La classe serait:

```
class Affine37:

        size=37

        def __init__(self, a ,b):
                self.a = a
                self.b = b

        def encrypt(m):
                return (a*m+b)%size

        def uncrypt(c):
                return (invMod(a,size)*(c-b))%size
```

e) Ici on cherche la fonction pour crypter E en B puis on vérifie avec L en 0

on as $L = 11$ et $ENCRYPT(11) = 27 = 0$ car le premier char est 0

on as $E = 4$ et $ENCRYPT(4) = 1 = B$

donc:

$$27 = a * 11 + b \pmod{37}$$
$$1 = a * 4 + b \pmod{37}$$

on cherche en brute et on trouve $a = 9$ et $b = 2$

$$9 * 4 + 2 \pmod{37} = 36 + 2 \pmod{37} = 38 \pmod{37} = 1$$
$$9 * 11 + 2 \pmod{37} = 99 + 2 \pmod{37} = 101 \pmod{37} = 27$$

f) 0BOURB0BQZ05ENW =
27 1 14 20 17 1 27 1 16 25 27 32 4 13 22

$$a' = INVMOD(9,37) = 33$$

on utilise la formule de décryptage:
$$a'(y - b) \pmod{37} = 33(y - 2) \pmod{37}$$

$33 * (27 - 2) \pmod{37} = 33 * 25 \pmod{37} = 11 = L$

$33 * (1 - 2) \pmod{37} = -33 \pmod{37} = 4 = E$

$33 * (14 - 2) \pmod{37} = 26 =$

$33 * (20 - 2) \pmod{37} = 2 = C$

$33 * (17 - 2) \pmod{37} = 14 = O$

… D
… E
… “ “
… E
… S
… T
…” “
… 1
… 2
… 3
…4
les message crypté est: LE CODE EST 1234

mon paypal : paypal@WillEsc

<u>CHIFFREMENT RSA:</u>

Chiffrement RSA:

Les notions importantes à retenir ici sont qu'on dispose d'une clé privé constitué de **p, q, d** et une clé publique **n, e**

p et q sont deux nombre premier (normalement suffisament grand)

$$n = p * q$$

$$\varphi(n) = (p-1)(q-1)$$

e sera donné dans l'énoncé

$$d = INVMOD(e, \varphi(n))$$

Pour chiffrer un message m en un message c <u>(côté publique)</u>

$$c = m^e \pmod{n}$$

Pour déchiffrer un message c en un message m <u>(côté privé)</u>

$$m = c^d \pmod{n}$$

SIGNATURE RSA:

Pour signer un message m avec une signature s <u>(côté privé)</u>

$$s = m^d \pmod{n}$$

Pour vérifier une signature s d'un message m on vérifie <u>(côté publique)</u>

$$m == s^e \pmod{n}$$

mon paypal : paypal@WillEsc

CORRECTION EXERCICE 3 (2020-2021):

a) on as n = 187 et e = 3
   le message crypté c est:
   $c = 15^3 \pmod{187} = 9$

b) on as maintenant $n = 187$ et $\varphi(n) = 160$ il faut trouver p et q (qui sont premier)

   on sait que $n = p * q$ et $\varphi(n) = (p-1)(q-1)$

   Comme n est petit on peut tester en brut en faisant 187/p = q

   on va trouver p = 11 et q = 17
   et en effet
   $11 * 17 = 187 = n$
   $10 * 16 = 160 = \varphi(n)$

CORRECTION EXERCICE 4 (2020-2021):

on cherche à savoir si une signature d'un document qui as été crypté par chiffrement RSA est authentique

on as n = 1833 et e = 3
pour vérifier une signature RSA on applique la formule:
$m == s^e \pmod{n}$
Si on trouve m la signature est authentique sinon elle est fausse

pour m = 12 et s = 363
$363^3 \pmod{1833} = 12$ soit $POWERMOD(363, 3, 1833) = 12$
la signature est authentique

pour m = 13 et s = 227
$227^3 \pmod{1833} = 710$ soit $POWERMOD(227, 3, 1833) = 710$
la signature est fausse

mon paypal : paypal@WillEsc

CHIFFREMENT ELGAMAL:

Le chiffrement d'ElGamal s'appuie sur la complexité du problème du logarithme discret.

Il y a plusieurs notions à comprendre pour pouvoir l'utiliser.

- Un nombre premier sûr p s'écrit sous la forme 2q+1 avec q premier

- L'ordre d'un sous groupe est son nombre d'éléments.
  en fait ici on va avoir un p et demander l'ordre du sous groupe avec le générateur g pour cela il faudra simplement appliqué la formule
  $$g^i \pmod p$$
  et trouvé combien d'éléments distincts on va avoir avant de recommencer le cycle.

- La clé privé qui sera un x

- La clé publique de ce chiffrement sera une triplette composé de:
  - p le nombre premier sûr choisi
  - g le générateur
  - y qui se calcule grâce à $y = g^x$ **OU** $y = g^x \pmod p$

- Pour chiffrer un message on va donner un c1 et un c2 qu'on aura modifier grâce à un k choisie aléatoirement.
  - $c1 = g^k$ **OU** $c1 = g^k \pmod p$ = $POWERMOD(g, k, p)$
  - $c2 = m * y^k$ **OU** $c2 = m * y^k \pmod p$

- pour déchiffrer un message on applique la formule

  $$m = \frac{c2}{c1^x}$$

  OU

  $$m = (c2 * (INVMOD(c1, p)^x \pmod p)) \pmod p$$

pourquoi ça marche???

$$\frac{c2}{c1^x} = \frac{m * y^k}{(g^k)^x} = \frac{m * (g^x)^k}{(g^k)^x} = \frac{m * (g^{xk})}{(g^{kx})} = m$$

mon paypal : paypal@WillEsc

CORRECTION EXERCICE 5 (2020-2021):

a) $53 * 2 + 1 = 107$ avec 53 premier donc oui 107 est premier sûr

b) Ici on cherche la taille du cycle en utilisant G = 9 et la formule $g^i \pmod p$ ce qui donne:

$9^0 \pmod{107} = 1$

$9^1 \pmod{107} = 9$

$9^2 \pmod{107} = 81$

$9^3 \pmod{107} = 87$

$9^4 \pmod{107} = 34$

…

$9^{51} \pmod{107} = 37$

$9^{52} \pmod{107} = 12$

$9^{53} \pmod{107} = 1$

$9^{54} \pmod{107} = 9$

A partir de 53 on recommence le cycle, l'ordre du sous groupe engendré par g est 53.

c) Une clé publique ElGamal ressemble à une triplette (g,p,y)
on calcule

$y = g^x \pmod p = 9^8 \pmod{107} = 43046721 \pmod{107} = 86$

Donc la clé publique est : (9,107,86)

d) Pour chiffrer un message on utilise la clé publique et on cherche c1 et c2 tel que

- $c1 = g^k$ **OU** $c1 = g^k \pmod p$
- $c2 = m * y^k$ **OU** $c2 = m * y^k \pmod p$

Comme on as générer y en utilisant le modulo **ON DOIT** générer c1 et c2 avec le modulo.

On as $m = 10$ et $k = 5$

$c1 = 9^5 \pmod{107} = 59049 \pmod{107} = 92$

$c2 = 10 * 86^5 \pmod{107} = 10 * 4704270176 \pmod{107} = 34$

Le message cryptée est donc (92,34)

e) Pour déchiffrer le message (92,34) on applique
$m = (c2 * (INVMOD(c1,p)^x \pmod p)) \pmod p$
soit
$(34 * (INVMOD(92,107)^8 \pmod{107})) \pmod{107}$
$(34 * (57^8 \pmod{107})) \pmod{107}$
$(34 * 101) \pmod{107}$
$3434 \pmod{107}$
$m = 10$

<u>CORRECTION EXERCICE 2 (2020-2021): (PAS 100% JUSTE)</u>

a) Si on fixe k alors dans la clé publique (c1,c2), c1 sera alors toujours identique et c2 codera toujours de la même façon une lettre. En ayant un texte connu on va pouvoir le crypté et faire un dictionnaire.
   Si on regarde avec l'exercice (d) d'avant si on laisse k = 5 alors c1 sera toujours 92 et à chaque fois qu'on aura m = 10 alors c2 sera 34, donc pour une clé publique (g,p,y) en utilisant k=5 à chaque fois si on croise un message crypté (92,34) on devine que m = 10.

SOURCE

Chiffrement Affine:

https://fr.wikipedia.org/wiki/Chiffre_affine

Chiffrement RSA:

https://fr.wikipedia.org/wiki/Chiffrement_RSA
https://www.dcode.fr/chiffre-rsa

Signature RSA:

https://www.di.ens.fr/~nitulesc/files/crypto6.pdf

Chiffrement ElGamal:

https://www.youtube.com/watch?v=Bs3ITCajSZ8

Et parfois:

https://igm.univ-mlv.fr/~jyt/Crypto/index.html