IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Cryptologie et protection des données

Reda Bellafqira
IMT Atlantique

2024

Définition : Groupe

On appelle *groupe* un couple G muni d'une loi interne $*$ telle que :

1. La loi $*$ est *associative* (i.e. pour tous x, y, z dans G, (x*y)*z = x*(y*z)).
2. Il existe un *élément neutre* e (i.e. pour tous $x \in G, x * e = e * x = x$).
3. Tout élément a un *symétrique* (i.e. pour tout $x \in G$, il existe $y \in G$ tel que $x * y = y * x = e$).

Définition : Groupe

- De plus, pour tous $x, y \in G$, si $x * y = y * x$, on dit que G est un groupe *commutatif* (ou *abélien*).
- Un groupe peut être fini ou infini. S'il est fini, on appelle ordre du groupe le nombre de ses éléments noté Card(G) (i.e. son cardinal).

Exemples : Groupe

- L'ensemble $(\mathbb{Z}, +)$ des entiers relatifs muni de la loi de composition $(x, y) \to x + y$ est un groupe commutatif, d'élément neutre 0. On l'appelle le groupe additif des entiers relatifs.
- En remplaçant $\mathbb{Z}$ par $\mathbb{Q}$, $\mathbb{R}$ ou $\mathbb{C}$, on obtient le groupe additif des nombres rationnels, des nombres réels et des nombres complexes, respectivement.

Définition : Sous-groupe

Soit $(G, *)$ un groupe et $H \subset G$. On dit que H est un sous-groupe de G si les conditions suivantes sont réalisées :

1. L'élément e appartient à H.
2. Pour tous $x, y \in H$, l'élément $x * y$ est dans H.
3. Pour tout $x \in H$, l'inverse x^{-1} de x est dans H

▶ Exemple : $(\mathbb{Z}, +)$ est un sous-groupe de $(\mathbb{R}, +)$

Groupe cyclique

Si $(G, *)$ est un groupe, l'ordre d'un élément $g \in G$ est le plus petit entier $n \geq 1$ tel que $g^n = 1$. Un groupe, $(G, *)$, est cyclique s'il existe un élément $g \in G$ tel que tout élément $a \in G$ soit de la forme g^n

Théorème

Si G est un groupe fini de cardinal (on dit aussi d'ordre) = $|G| = n$ alors l'ordre de tout élément $g \in G$ est un diviseur de n.

Définition : Anneau

Un anneau est un ensemble $(A, +, *)$ muni de deux lois internes. La loi $+$ détermine sur A une structure de groupe abélien. La loi $*$ possède pour tout $a, b, c \in A$ les propriété suivantes

- Elle est associative :
$$((a * b) * c) = (a * (b * c))$$
- Elle est distributive par rapport à la loi $+$:
$$a * (b + c) = a * b + a * c$$

► Exemple : $(\mathbb{Z}, +, \times)$ est un anneau.

Définition : Anneau

- Il y a un élément unité noté 1 tel que $a * 1 = 1 * a = a$
- Si la multiplication est commutative, pour tous $a, b \in A$ $a * b = b * a$, on dit que l'anneau est commutatif.

► Exemple : $(\mathbb{Z}, +, \times)$ est un anneau commutatif.

► Exemple : le groupe des matrices $M_{n,m}(\mathbb{Z})$ est un anneau *non* commutatif.

Définition

Un élément a de l'anneau A est inversible s'il existe $a' \in A$ tel que $a * a' = a' * a = 1$, a' est l'inverse de a.

▶ Exemple : Dans les entiers relatifs $\mathbb{Z}$ seuls 1 et -1 sont inversibles.

Définition : Corps

Un corps est un anneau dans lequel tout élément non nul, i.e. distinct de l'élément neutre pour l'addition a un inverse pour la multiplication, autrement dit :

- Pour tout $a \neq 0$, il existe a' tel que $a * a' = a' * a = 1$.

- Exemple : $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ sont des corps.

Théorème : Division euclidienne

Si a et b sont deux entiers ($b \neq 0$), il existe un unique couple d'entiers q et r tel que $a = bq + r$ tel que $0 \leq r \leq b - 1$.

1. On dit que b est un diviseur de a s'il existe $q \in \mathbb{N}$ tel que $a = bq$, on note $a|b$ pour dire a divise b.
2. Tout entier a supérieure à 2 possède au moins deux diviseurs triviaux 1 et lui-même car on a toujours $a = 1 \times a$, $a = a \times 1$
3. Mais il peut en avoir d'autres par exemple $28 = 4 \times 7 = 2 \times 2 \times 7 = 2 \times 14$

Définition : Nombre premier

Un nombre premier est un nombre qui n'a pas d'autre diviseurs que les diviseurs triviaux i. e. 1 et lui-même (e.g. 2,3,5,7,...,37,...)

1. Un nombre entier $n > 0$ est premier s'il n'est divisible par aucun entier inférieur ou égal à $\sqrt{n}$.
2. Soit p un nombre premier et a, b deux entiers non nuls. Si p divise le produit ab alors il divise au moins l'un des deux nombres a ou b.
3. Tout nombre entier différent de 1 possède une unique décomposition en facteurs premiers à l'ordre près des facteurs, certains facteurs peuvent être répétés.
4. L'ensemble des nombres premiers est infini.

Définition : Plus Grand Commun Diviseur (PGCD)

Le plus grand commun diviseur de deux nombres entiers a et b est le plus grand des entiers qui divisent à la fois a et b, on le note $PGCD(a, b)$ ou $a \wedge b$ ou encore (a, b).

1. Si le PGCD de a et b vaut 1 on dira que a et b sont premiers entre eux ou sont des nombres relativement premiers.
2. Attention bien distinguer entre nombres relativement premiers et nombres qui ne se divisent pas.
3. 35 et 49 ne se divisent pas car $49 = 35 \times 1 + 14$ et $35 = 0 \times 49 + 35$, mais ils ne sont pas premiers entre eux car 7 divise à la fois 35 et 49.

Définition : Plus Grand Commun Diviseur (PGCD)

Le plus grand commun diviseur de a et de b, entiers naturels, s'obtient de la manière suivante. Si

$$a = p_1^{\alpha_1} p_2^{\alpha_2} ... p_r^{\alpha_r}$$
$$b = p_1^{\beta_1} p_2^{\beta_2} ... p_r^{\beta_r}$$

avec $\alpha_i, \beta_i \geq 0$

leur décomposition en facteurs premiers, alors le PGCD de a et de b est :

$$a \wedge b = p_1^{inf\{\alpha_1, \beta_1\}} ... p_r^{inf\{\alpha_r, \beta_r\}}$$

Théorème de Bezout

- Soit a et b deux entiers naturels de PGCD : $a \wedge b = d$. Alors il existe deux entiers relatifs u, v tels que
$$au + bv = d$$
u et v sont appelés les coefficients de Bezout.
- a et b sont premiers entre eux si seulement si il existe deux entiers u et v tels que
$$au + bv = 1$$

Algorithme d'Euclide

L'algorithme d'Euclide permet de calculer le PGCD de deux entiers naturels non nuls a et b. On procède de la manière suivante :

- On effectue la division euclidienne de a par b. On note r le reste (on n'utilise pas le quotient).
- On remplace ensuite a par b et b par r. Tant que le reste est différent de 0, on réitère le procédé.

Après un certain nombre d'itérations, on obtiendra un reste égal à 0. Le PGCD de a et de b est alors le reste précédent (c'est à dire le dernier reste non nul).

Congruences

Soit a, b et m trois entiers. On dit que a est congru à b modulo m et on écrit

$$a \equiv b \mod m$$

si la différence $a - b$ est divisible par m

Congruences

Soit m un entier non nul, alors a et b sont congrus modulo m si et seulement si les restes de la division euclidienne de a par m et de b par m sont les mêmes

Congruences

L'addition et la multiplication sont compatibles à la relation de conguence sur les entiers :

$$((a \mod m) + (b \mod m)) \mod m = (a + b) \mod m$$
$$((a \mod m) \times (b \mod m)) \mod m = (a \times b) \mod m$$

Exemples : Prenons $m = 2$ alors deux entiers a et b sont congrus modulo 2 s'il sont : soit tous les deux pairs ou tous les deux impairs.

Congruences

- L'ensemble des éléments de $\mathbb{Z}$ qui sont congrus modulo m à un entier fixé s'appellera une classe de congruence modulo m
- Un élément d'une classe de congruence s'appelle un représentant de la classe de congruence.
- Parmi les représentants d'une classe de congruence, il y en a toujours un unique compris entre 0 et $m - 1$.
- On notera $\mathbb{Z}/m\mathbb{Z}$ l'ensemble des classes de congruence de $\mathbb{Z}$ modulo m.

Congruences

Tout élément de $\mathbb{Z}/m\mathbb{Z}$ possède un opposé pour l'addition. Les éléments de $\mathbb{Z}/m\mathbb{Z}$ qui ont un inverse multiplicatif sont ceux dont les représentants dans $\mathbb{Z}$ sont premiers à m.

Congruences

Si p est premier, l'anneau $\mathbb{Z}/p\mathbb{Z}$ est un corps fini à p éléments, c'est à dire que tout élément différent de la classe de zéro possède un inverse multiplicatif. On le note $\mathbb{F}_p$.

Théorème

Si p est premier, le groupe multiplicatif $(\mathbb{Z}/p\mathbb{Z})^*$ = { éléments inversibles de $(\mathbb{Z}/p\mathbb{Z}, \times)$ } est cyclique, i.e. il existe g premier avec p tel que

$$\{1, ..., p-1\} \equiv \{g^n, 0 \leq n \leq p-2\}$$

Congruences

- Un générateur g du groupe cyclique $\mathbb{Z}/p\mathbb{Z}$ s'appelle une racine primitive modulo p.
- Si $q = p^r$ avec p premier le groupe multiplicatif, $\mathbb{F}_q^*$, des éléments inversibles de $\mathbb{F}_q$ est cyclique.

Petit théorème de Fermat

Soit p un nombre premier. Tout entier a vérifie la congruence $a^p \equiv a \mod p$, et si $a \wedge p = 1$ on aussi

$$a^{p-1} \equiv 1 \mod p$$

Exemple

$$\begin{aligned}
3^6 \mod 7 &= 3^{2+4} \mod 7 & (1)\\
&= 3^2 \mod 7 \times (3^2)^2 \mod 7 & (2)\\
&= 2 \times 4 \mod 7 = 1 \mod 7 & (3)
\end{aligned}$$

Théorème des restes chinois

Soient $n_1, ..., n_r$, r entiers deux à deux premiers entre eux, soit le produit $n = n_1.n_2...n_r$ et soient $x_1, ..., x_r$ une suite d'entiers, le système d'équations :

$$\left\{\begin{array}{cccl} x & \equiv & x_1 & \mod n_1 \\ ... & ... & ... & \\ x & \equiv & x_r & \mod n_r \end{array}\right.$$

admet une unique solution x modulo n donnée par la formule :

$$x = x_1 N_1 y_1 + ... + x_r N_r y_r \mod n$$

où $N_i = \frac{n}{n_i}$ et $y_i \equiv N_i^{-1} \mod n_i$ pour $1 \leq i \leq r$.

Indicatrice d'Euler

Si m est un entier positif on note $\phi(m)$ le nombre d'entiers $b < m$ et premiers à m, ou de manière équivalente $\phi(m)$ est le nombre d'entiers $b < m$ inversibles modulo m

Proposition

Si $m = p_1^{\alpha_1}...p_r^{\alpha_r}$ est la décomposition de m en facteurs premiers distincts alors

$$\phi(m) = p_1^{\alpha_1 - 1}(p_1 - 1)...p_r^{\alpha_r - 1}(p_r - 1)$$

et si $m \wedge n = 1$ alors

$$\phi(mn) = \phi(m)\phi(n)$$

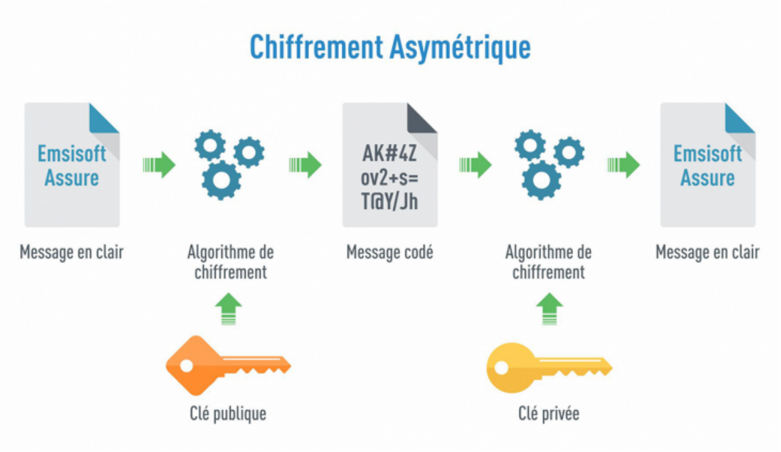
Théorème d'Euler

Soit $m > 1$ un entier et soit a un entier premier à m, alors on a : $a^{\phi(m)} \equiv 1 \mod m$

Corollaire

Soit $m \in \mathbb{N}$ et soit $a \in \mathbb{N}$ tel que $a \wedge m = 1$ alors le plus petit entier $e \geq 1$ tel que $a^e \equiv 1 \mod m$ est un diviseur de $\phi(m)$.

chiffrement asymétrique



Définition

$$\begin{array}{rccc} f_{K_{pu}} : & M & \longrightarrow & C \\ & m & \longmapsto & f(m) \end{array}$$

- M et C : espace des messages en clair et chiffré, respectivement.
- $f_{K_{pu}}$: une application bijective -> tout élément de M a un seul correspondant en C.
- K_{pu} : Clé publique pour le chiffrement

Définition

$$f'_{K_{pr}} : \begin{array}{rcl} C & \longrightarrow & M \\ x = f(m) & \longmapsto & f'(x) = f'(f(m)) = m \end{array}$$

- M et C : espace des messages en clair et chiffré, respectivement.
- $f'_{K_{pr}}$: fonction de déchiffrement paramétré par une clé privée.

Chiffrement Homomorphe

$$\begin{array}{rccc} f: & (M,+) & \longrightarrow & (C,\times) \\ & m & \longmapsto & f(m) \end{array}$$

- $\forall\ m, m' \in M$ alors $f(m+m') = f(m) \times f(m')$

Chiffrement sématiquement sûr

$$\begin{array}{rccc} f: & M \times R & \longrightarrow & (C, \times) \\ & (m, r) & \longmapsto & f(m, r) \end{array}$$

- Un message $m \in M$ peut avoir plusieurs chiffrés différents. Soit $r \neq r'$ alors $f(m, r)$ et $f(m, r')$ sont deux chiffrés de m différents

Propriété de sécurité

La sécurité des systèmes à clef publique repose sur divers problèmes calculatoires.

- RSA (Rivest, Shamir, Adleman, 1977) : Il est basé sur la difficulté de la factorisation des grands entiers.
- ELGamal : Problème du calcul du logarithme discret.
- Paillier : le problème de classe de résiduosité composite (la difficulté de distinguer les résidus d'ordre N modulo N^2 des non-résidus d'ordre N).

RSA [1977]

- Soit p et q deux grands nombres premiers distincts et $N = pq$ (nombre de RSA).
- Soit $e \in [\![65537, 2^{256}]\!]$ un entier impaire tel que $pgcd(e, \varphi(N)) = 1$.
- Soit $d \leq \varphi(N)$ un entier tel que $ed \equiv 1 \mod (\varphi(N))$.

La clé publique K_{pu} est $\{N, e\}$; la clé privée K_{pr} est $\{d\}$.
La fonction de chiffrement est définie comme :

$$\begin{array}{rccl} f_{K_{pu}} : & \mathbb{Z}_N^* & \longrightarrow & \mathbb{Z}_N^* \\ & m & \longmapsto & f(m) \equiv m^e \mod N \end{array}$$

RSA [1977]

La fonction de déchiffrement est définie comme :

$$\begin{array}{rccl} f'_{K_{pr}} : & \mathbb{Z}_N^* & \longrightarrow & \mathbb{Z}_N^* \\ & c & \longmapsto & c^d \equiv m^{ed} \equiv m \bmod N \end{array}$$

- RSA est multiplicativement homomorphe : soit $m, m' \in \mathbb{Z}_N^*$ alors
$$f(mm') = f(m)f(m')$$

- La sécurité repose sur le problème de la factorisation.

Exemple dégénéré

Alice souhaite envoyer un message à Bob qui choisit et calcule :

$$p = 11, q = 3, N = 33, \varphi(N) = 20, e = 3, d = 7$$

Il envoie à Alice sa clé publique $\{N = 33, e = 3\}$.
Alice souhaite envoyer à Bob le message 29 pour se faire elle le chiffre :

$$c = 29^3 \bmod 33 = 24389 \bmod 33 = 2 \bmod 33$$

et envoie le chiffré $c = 2$ à Bob qui grâce à sa clé privée $\{d = 7\}$ déchiffre le message :

$$m' = 2^7 \bmod 33 = 128 \bmod 33 = 29 \bmod 33$$

Bob a bien retrouvé le message d'Alice $m = m' = 29$.

► En utilisant les propriétés du modulo et le CRT le chiffrement ainsi que le déchiffrement peuvent se faire plus rapidement.

RSA-CRT

Le déchiffrement RSA-CRT consiste à deux étapes :

1. Etant donné p, q avec p> q, précalculez les valeurs suivantes :
$$dp = d \bmod p - 1$$
$$dq = d \bmod q - 1$$
$$qInv = q^{-1} \bmod p$$
2. Pour calculer le message m sachant c :
$$m_1 = c^{dp} \bmod p$$
$$m_2 = c^{dq} \bmod q$$
$$h = qInv.(m_1 - m_2) \bmod p$$
$$m = m_1 + h.q$$

Paillier [1999]

- Soit p et q Deux grands nombres premiers distincts et $N = pq$.

La clé publique K_{pu} est $\{N\}$; la clé privée K_{pr} est $\{\varphi(N)\}$.
Pour chiffrer, La fonction de chiffrement est définie comme :

$$f_{K_{pu}} : \begin{array}{ccc} \mathbb{Z}_N \times \mathbb{Z}_N^* & \longrightarrow & \mathbb{Z}_{N^2}^* \\ (m, r) & \longmapsto & f(m) = (1+N)^m \times r^N \bmod N^2 \end{array}$$

► La sécurité repose sur le problème de la factorisation.

Paillier [1999]

La fonction de déchiffrement est définie comme :

$$\begin{aligned} f'_{K_{pr}} : \quad \mathbb{Z}^*_{N^2} &\longrightarrow \mathbb{Z}_N \\ c &\longmapsto \frac{c^{\varphi(N)} \bmod N^2 - 1}{N} \times \varphi(N)^{-1} \\ &\equiv \frac{1 + Nm\varphi(N) - 1}{N} \times \varphi(N)^{-1} \\ &\equiv m \bmod N \end{aligned}$$

- Cryptosystème de Paillier est additivement homomorphe : soit $m, m' \in \mathbb{Z}_N$ alors

$$f(m+m') = f(m)f(m')$$

▶ La sécurité repose sur le problème de la factorisation.

Exemple dégénéré

Bob choisit et calcule :

$$p = 11, q = 3, N = 33, \varphi(N) = 20$$

Il peut alors envoyé à Alice sa clé publique $N = 33$.

Alice souhaite envoyer à Bob le message 26 pour se faire elle choisit $r = 32$, puis elle le chiffre :

$$c = (1+33)^{26} \times 32^{33} \bmod 33^2 = 230 \bmod 33^2$$

et envoie le chiffré $c = 230$ à Bob qui grâce à sa clé privée $\{\varphi(33) = 20\}$ déchiffre le message :

$$m' \equiv \frac{230^{20} \bmod 33^2 - 1}{33} \times 20^{-1} \equiv 25 * 5 \equiv 26 \mod 33$$

Bob a bien retrouvé le message de Alice $m = m' = 26$.

Paillier [1999]

La fonction de déchiffrement-CRT est définie comme :
L'objectif est d'évaluer $x = c^{\varphi(N)} \bmod N^2$ en utilisant le théorème des restes chinois, pour le faire on calcule :

$$\begin{aligned} x_p &= c^{\varphi(N)} \bmod p^2 \\ x_q &= c^{\varphi(N)} \bmod q^2 \\ x &= (q^{-2}(x_p - x_q) \bmod p^2)q^2 + x_q \end{aligned}$$

Pour retrouver le message m, on calcule :

$$m = \lfloor \frac{x-1}{N} \rfloor \varphi(N)^{-1} \bmod N$$

Note : L'inverse de q^2 (q^{-2}) est calculé modulo p^2, et l'inverse de $\varphi(N)$ ($\varphi(N)^{-1}$) est calculé modulo N.

ElGamal [1985]

- Soit p un grand nombre premier.
- Un générateur g de $\mathbb{Z}_p^*$.
- Un entier $b \in [\![1, p-2]\!]$ et $B \equiv g^b \bmod (p)$.

La clé publique K_{pu} est $\{p, g, B\}$; la clé privée K_{pr} est $\{b\}$.

Pour chiffrer, $m \in \mathbb{Z}_p^*$ on choisit $k \in [\![1, p-2]\!]$, puis on calcule :

$$K \equiv g^k \bmod p \quad \text{et} \quad c \equiv mB^k \bmod p$$

Pour déchiffrer le message en calculant $cK^{-b} \bmod p$:

$$cK^{-b} \equiv cg^{-kb} \equiv mg^{bk}g^{-bk} \equiv m \bmod p$$

► La sécurité repose sur le problème du logarithme discret et de D-H.

Exemple dégénéré

Bob choisit et calcule :

$$p = 167, g = 57, b = 17, B \equiv 3 \bmod (167)$$

Il peut alors envoyé à Alice sa clé publique $\{p = 167, g = 57, B = 3\}$.
Alice souhaite envoyer à Bob le message 90. Elle choisit $k = 100$ donc $K = 57^{100} \bmod 167 = 3 \bmod 167$ puis elle le chiffre :

$$c = 90 \times 3^{100} \bmod 167 = 5 \bmod 167$$

et envoie le chiffré $c = 5$ à Bob qui grâce à sa clé privée $\{b = 17\}$ déchiffre le message :

$$m' = 5 \times (3^{17})^{-1} \bmod 167 = 5 \times 28 \bmod 167 = 90 \bmod 167$$

Bob à bien retrouvé le message de Alice $m = m' = 90$.

▶ Le schéma est sémantiquement sûr.

Un test de primalité est un test qui détermine si un entier est premier ou composé. Il existe deux familles de tests de primalité :

- Tests de primalité prouvables (déterministes) qui déterminent la primalité avec certitude.
- Tests de primalité probabiliste qui déterminent si un nombre est premier ou non avec une probabilité (négligeable) d'erreur.

Théorème de Fermat

Soit p un nombre premier et a un entier tel que $a < p$ alors :

$$\text{Si } \; pgcd(a, p) = 1 \Rightarrow a^{p-1} \equiv 1[p]$$

la réciproque est fausse, par exemple : $561 = 3 \times 11 \times 17$; 561 n'est pas premier et pourtant, pour tout entier a < 561, on a $a^{561} \equiv a \mod 561$. Les nombres p qui font échouer la réciproque sont appelés nombres de Carmichaël, et forment un ensemble infini.

Test de Miller Rabin

On se donne un entier N impair et un entier t (le degré de sécurité). On calcule s et m tels que $N-1=2^s m$ avec m impair. Puis on répète t fois la séquence suivante :

1. choisir un entier a aléatoire tel que $1<a<N-1$
2. si $a^m \neq 1 \mod N$ et $a^{2^r m} \neq -1 \mod N$ pour tout r tel que $1 \leq r \leq s-1$, alors retourner N est composé. a est appelé un témoin de Miller pour le fait que N est composé.

► Pour un nombre impair composé n, $3/4$ au moins des entiers a, $1<a<n$, sont des témoins de Miller pour n. Il suffit donc de répéter le test pour suffisamment d'entiers a, pour que la probabilité qu'un entier n composé soit déclaré premier devienne très faible.

GMP

GMP (GNU Multiple Precision arithmetic library) est une bibliothèque de calcul sur les grands entiers. Cette bibliothèque fournit de nombreuses fonctions de calcul sur différents types multi-précision :

- (grands) entiers : $\mathbb{Z}$
- (grands) rationnels : $\mathbb{Q}$
- (grands) flottants : $\mathbb{R}$

Commande d'installation avec Conda

» conda install -c conda-forge gmpy2

Commande d'installation dans un terminal

» sudo apt install libmpc-dev
» sudo apt install python3-pip
» pip3 install –user gmpy2==2.1.0a2

Documentation

La documentation se trouve dans le lien suivant :
https ://gmpy2.readthedocs.io/en/latest/intro.html

Commande d'installation Conda sur Google Colab

» !pip install -q condacolab
» import condacolab
» condacolab.install()

Afficher la version de conda

» !conda –version

Regarder où ce situe le dossier conda

» !which conda

Commande d'installation avec Conda

» !conda install -c conda-forge gmpy2

Typage

Les types de bases définis dans GMP sont les suivants :

- *mpz* pour les entiers relatifs $\mathbb{Z}$
- *mpq* pour les rationnels $\mathbb{Q}$
- *mpf* pour les flottants $\mathbb{R}$

Remarque :

- Pour une utilisation cryptographique, nous ne nous intéresserons qu'au type "grand entier" $\mathbb{Z}$ de GMP (type *mpz*).

► Exemple : $x = mpz(10)$

Fonctions d'assignation d'une valeur

- $x = mpz()$, créer un nouveau objet mpz de valeur 0
- $x = mpz(y)$, créer un objet mpz de valeur y (si y est un float, il retourne sa partie entière)

Ces fonctions permettent de faire un équivalent de $x = y$.

Exemple

```
» import gmpy2
» from gmpy2 import mpz
» mpz('123') + 1
mpz(124)
» 10 - mpz(1)
mpz(9)
```

- mpz() : assignation à un mpz une chaine de caractères en précisant en quelle base est représenté l'entier dans la chaine de caractères. La base est déterminée par le préfixe de la chaine : 0x (base 16), 0b (base 2), 0 (base 8), et en base 10 par défaut.

bit_set(...)

$x.bit\_set(n)$ retourne une copie de x dont le bit n égale à 1.

$x.bit\_length()$

retourne le nombre de bits significatifs dans la représentation binaire de x. Pour la compatibilité avec Python, $mpz(0).bit\_length()$ renvoie 0.

Exponentiation modulaire : *powmod*(...)

$powmod(base, exp, mod)$ retourne $(base**exp)$ mod mod.
cette fonction calcule l'exponentiation modulaire $base^{exp}$ modulo mod. Les exposants négatifs sont autorisés si $base$ est inversible modulo mod.

Test de primalité

is_strong_prp()

La fonction *is_strong_prp*(n, a) implémente le test de Miller-Rabin pour tester la primalité de n dans la base a. Elle renvoie la valeur True ou False selon que n est probablement premier ou composé respectivement.

is_prime(x[, n=25])

retourne True si x est probablement premier. False est renvoyé si x est définitivement composé. x est vérifié par le test de Miller-Rabin pour les n petits diviseurs.

next_prime(x)

La fonction *next_prime*(...) détermine le plus petit entier premier strictement plus grand que "x".

PGCD : $gcd(x, y)$

La fonction gcd calcule le PGCD de x et y par l'algorithme d'Euclide.

PGCD étendue : $(g, u, v) = gcdext(x, y)$

La fonction gcdext calcule le PGCD de x et y par l'algorithme d'Euclide étendu. Après exécution, g contient la valeur du $PGCD(x, y)$ et u et v sont deux entiers vérifiant l'identité de Bezout ($g = xu + yv$).

Inverse modulaire : $invert(x, m)$

retourne y tel que $x * y == 1$ modulo m, ou 0 si aucun y n'existe.

mpz_random(...)

mpz_random(*random_state*, *n*) retourne un entier aléatoire uniformément distribué entre 0 et $n-1$. Le paramètre *random_state* doit d'abord être créé par *random_state*().

mpz_urandomb(...)

mpz_urandomb(*random_state*, *b*) retourne un entier aléatoire uniformément distribué entre 0 et $2{*}{*}b-1$. Le paramètre *random_state* doit d'abord être créé par *random_state*().

Le TP consiste à implémenter :

1. La clé publique et privée de RSA (p, q, e, d, N).
   - Générer deux nombres premiers (p, q) codés sur 512 bits.
2. Fonction de chiffrement.
3. Fonction de déchiffrement standard et CRT.

Le TP consiste à implémenter :

1. La clé publique et privée de El Gamal (p, g, b, k, B, K).
   - Générer un nombre premier codé sur 1024 bits.
   - Générer un générateur g du groupe multiplicatif $\mathbb{Z}_p^*$. (Pour cela il faut penser à utiliser la propriété disant que $p = 2q + 1$. Plus clairement, $g \in \mathbb{Z}_p^*$ est un générateur si seulement si $g^2 \neq 1 \mod p$ et $g^q \neq 1 \mod p$).
2. Fonction de chiffrement.
3. Fonction de déchiffrement.

Le TP consiste à implémenter :

1. La clé publique et privée de Paillier (p, q, ϕ, N).
   - Générer deux nombres premiers (p, q) codés sur 512 bits.
2. Fonction de chiffrement.
3. Fonction de déchiffrement.