# Cryptographie avancée - TP 4

Anne Garcia-Sanchez

M2i - CFA CCI Avignon, 24 octobre 2024

**Arithmétique pour les chiffrements asymétriques: inverse modulaire**

L'inverse modulaire d'un entier relatif $a$ modulo $n$ est un entier $u$ satisfaisant l'équation :

$a \times u \equiv 1 \pmod{n}$ et $u$ peut être noté $a^{-1}$

L'inverse de $a$ modulo $n$ existe si et seulement si $a$ et $n$ sont premiers entre eux c'est à dire qu'ils n'ont pas de diviseur commun autre que 1.

## 1 Inverse modulaire par tâtonnement pour des petits nombres

Par tâtonnement, chercher les inverses modulaires suivants puis vérifier avec Python.

1. l'inverse de 2 modulo 15:
2. l'inverse de 5 modulo 29:
3. l'inverse de 3 modulo 8:
4. l'inverse de 3 modulo 9:

## 2 Inverse modulaire par l'algorithme d'Euclide étendu

On rappelle l'algorithme d'Euclide étendu:

**Entrée**: entiers $a$, $b$
**Sortie**: $r$ entier et $u, v$ entiers relatifs tels que $r = PGCD(a, b)$ et $a \times u + b \times v = r$

(r0, u0, v0, r1, u1, v1) ← (a, 1, 0, b, 0, 1)
**tant que** $r1 \neq 0$ **faire**
.... q = r0 // r1 (quotient entier)
.... (r0, u0, v0, r1, u1, v1) ← (r1, u1, v1, r0-q×r1, u0-q×u1, v0-q×v1)
**fin tant que**
**Renvoyer** (u0, v0, r0)

1. Programmer une fonction Python implémentant l'algorithme d'Euclide étendu.

2. Utiliser la fonction écrite pour trouver l'inverse de 79 modulo 23.

   On rappelle que l'algorithme d'Euclide étendu permet de trouver $u$, $v$ et $r$ tels que
   $79 \times u + 23 \times v = r = 1$
   donc $79 \times u = 1 - 23 \times v$
   donc $79 \times u \equiv 1 \pmod{23}$
   donc $u$ est l'inverse de 79 modulo 23.

3. Vérifier que $79 \times 79^{-1} \equiv 1 \pmod{23}$.

4. Calculer l'inverse de 79 modulo 23 en utilisant la fonction `pow`.

# 3 Inversion modulaire pour inverser la fonction RSA

On veut déchiffrer un message secret:

```
ciphertext = b'\xb1\xa2\x0f\x18\xb0\xd7\x81-H\x19\x1bW\xbcf$\xa8\x98\x8b\xdf\xbe\xf1\x0f
\xcf\x97\xe1>\x99?\x19G\x8aie\x980^\x99F\x1aD\xed\x12{\x19\xe7\t\xba\x86'
```

Il a été chiffré avec un AES en mode GCM avec le nonce:

```
b'\x1b\xda3\xac\x87\xcdM\xd7\x18\x12\x8djbT\xee\x02'
```

On ne connait pas la clé mais on a intercepté la clé chiffrée par RSA:

c = 64058176184997834950693853025106406054

La clé publique est donnée par:

N = 236162332383177856298590687609142183389
e = 65537

On remarque que N est bien sûr trop petit pour assurer la sécurité du chiffrement!
On va pouvoir déchiffrer $c$.

1. Factoriser N pour calculer $\varphi(N) = (p-1)(q-1)$.
   Pour cela, on peut utiliser un outil en ligne tel que `Dcode`.

2. Vérifier que $e$ et $\varphi(N)$ sont bien premiers entre eux et retrouver la clé secrète $d$.
   C'est ici qu'on a besoin de l'inverse modulaire!

3. Avec la clé secrète, déchiffrer la clé du chiffrement AES.

4. Déchiffrer le message.

# 4 Bonus: RSA sur hackropole

1. Résoudre le challenge SMIC(2).

2. Résoudre le challenge `Rien à signaler`