# TD 1 Cryptographie

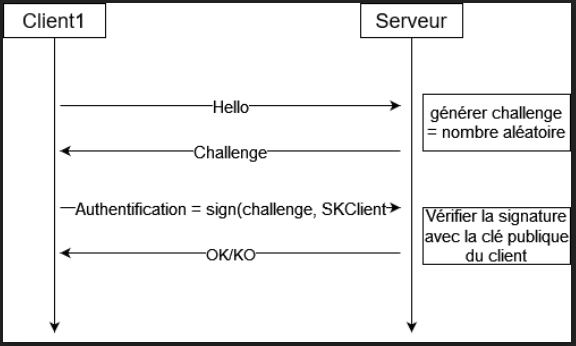
# Exercice 1 :

1. Chiffrement asymétrique
2. Que le serveur ai déjà la clef publique du client (ce qui nécessite en amont un canal authentifié)
3. Mot de passe / MAC (symétrique)

On pourrait utiliser un MAC Proctlimer

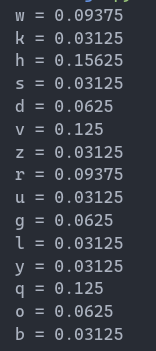
On ne peut utiliser la même clé sur plusieurs serveurs car le serveur doit aussi stocker la clé.

Si le serveur est corrompu la clé secrète sera corrompue



# Exercice 2 :

1. “bj qtaj hwduyt”



En % dans le mot

Code :

vasy

decode = "wkh sdvvzrug lv vhyhq grqw whoo dqbrqh"

decode = decode.replace(" ","")

j = {}

for i in decode:

if i not in j.keys():

j[i] = 1

else:

j[i] += 1

for inte in j.keys():

j[inte] = j[inte] / len(decode)

print(f'{inte} = {j[inte]}')

Les lettres les plus présentes sont h = 15% et v =12.5%’ ou q =12.5%. Ecart H - E → 3, écart V - T →2, écart Q-T → 3, la clé est donc -3, on peut ainsi décrypter le message : “**the password is seven don’t tell anyone**”

Exercice 3 :

1. “a simple example” - key = “TRUE” ⇒ T JCQICY IQRGTEV
2. INSPIRINGYOURDIGITALFUTURE avec TNCY

Exercice 4 :

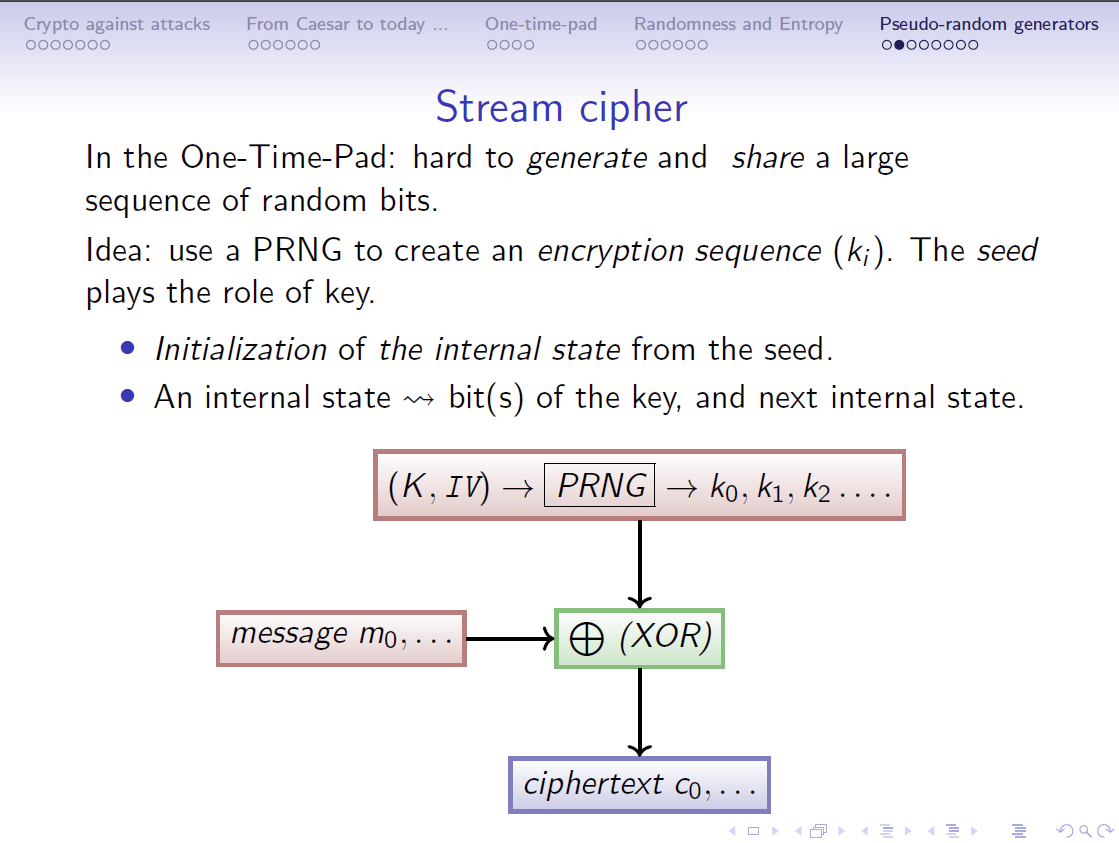
**PRNG : Pseudo-Random Generator**

1. A partir d’une clé K et d’un vecteur d’initialisation IV on génère une séquence pseudo aléatoire S(K, IV).

Pour chiffrer un message M on calcule C = M ⊕ S(K,IV)

Pour déchiffrer : C ⊕ S(K, IV) = M ⊕ S(K, IV) ⊕ S(K, IV) = M

⊕ = XOR



1. Soit C’ = C⊕M’, alors on a

C’=(M ⊕ S(K,IV))⊕ M’

=(M ⊕ M’)⊕ S(K,IV)

Donc c’est un chiffré valide pour M⊕M’.