Le seul scheme parfait : one-time pad (= chiffrement de vernam) mais on ne doit utiliser la clé qu’une seule fois.

chiffrement par block 🡪 symetrique

ECB : blocs de taille fixes, on les code chacun tout seul avec un algo de chiffrement. Sauf que on retrouve facilement le message de base

CBC : Blocs de taille fixes, on les code chacun avec un algo de chiffrement et le chiffré du bloc précédent

OFB : un peu différent des deux premiers

CTR : prendre un nombre au hasar, le completer avec des 0, le coder avec un algo de chiffrement, aire un xor , et le prochain on ajoute ..0001

2 types de fonction de chiffremnt/dechiffrement

DES : clé de 64 bits avec 8 bits qui servent a rien donc 56 bits

AES : 128 ou 192 ou 256 bits . 10 tours pour faire le message crypté

Fonction de hachage : prend un texte et renvoie une sortie de taille fixe peut importe la taille de l’entrée. Une bonne fonction de hachage ne renvoie jamais une même sortie pour deux entrées différentes.

Bonne fonction de hachage :

* résistance à une pré image : Si on veut une sortie y en particulier on ne doit pas pouvoir prédire le x
* résistante aux collisions
* second pré image resistant ( un peu comme collision)

C’est facile de hacher un truc mais très compliquer de faire l’inverse, c’est one way.

méthode pour hacher :

* Merkle Damgard
* MD5 ( cassée depuis 2004)
* SHA

Oracle : une fonction ( regarde )

Birthday paradox

MAC : code pour authentifier le message

Sécurité démantique : Si je vois un chiffré je ne peux rien déduire a part sa longueur.

* les chiffrés doivent etre indistinguables : si on a deux codes on ne peux pas dire celui là c’est le A et celui là le b
* résistants aux attaques

Asymétrique : Une clé publique et une clé privée

clé publique + message 🡪 chiffré

clé privée + message crypté 🡪 déchiffré

clé privée+ message 🡪 signature

signature + message déchiffré+ clé publique 🡪 vrai ou faux ( pour savoir si c’est bien cette personne qui m’envoie un message)

diffie-Hellman : prendre un groupe fini, Anissa envoie un ya=g^a , bilel anvoie gb=g^b et la clé de anissa et bilel est donc g^ab

equation fondamentale : Z=Y^x mod N

RSA : permet de generer des clés publique/ privée ( asymétrique)

Protocole de signature :

* generateur de clé
* algo de signature
* algo de verification

goal de securité :

* l’attaquant ne peut pas trouver la clé privée avec la clé privée
* l’attaquant ne peut pas trouver un moyen de trouver une signature sans la clé privée
* l’attaquant ne peut pas créer un message et une signature

signature avec RSA : message^clé privée

verification de la signature : verifier que message = signature^clée privée de l’envoyeur

el gamal : permet de signer et de chiffrer , déchiffrer ( voir le cours 4 vers p40 pour les calculs)

DSA. : alternative a el gamal