

ISIFC 3, Crypto Chiffrement Symétrique

Jean-François COUCHOT

Université Marie & Louis Pasteur, UFR-ST



Plan



Introduction générale au chiffrement

Chiffres symétriques

Attaque par force brute



Plan



Introduction générale au chiffrement

Terminologie

Grands principes en cryptologie

Chiffres symétriques

Attaque par force brute



Plan



Introduction générale au chiffrement

Terminologie

Grands principes en cryptologie

Chiffres symétriques

Attaque par force brute



La cryptographie

Définition et historique

Définition (Cryptographie)

La **cryptographie** («kryptos» caché, «graphein» écriture) est l'ensemble des techniques (algorithmes et protocoles) permettant de protéger un message

Historique : un art ancien et une science nouvelle

- ▶ 5 siècle avant J.C., Spartiates : utilisation de scytales
- ▶ Chiffre de César : décalage alphabétique
- ▶ 1580 : chiffre par substitution de symboles – Marie Stuart
- ▶ 1586 : traité des chiffres – Chiffre de Vigenère
- ▶ 1930 : enigma
- ▶ Fin 20ème : apparition du chiffrement à clé publique – RSA en 1977

La cryptographie

Formalisation



Notations

Le message en clair M est transformé en message chiffré C (*ciphertext*) via la fonction de chiffrement E (*encrypt*) et la clef K (*key*)

$$C = E(M, K)$$

Le message chiffré C est transformé en message en clair M via la fonction de déchiffrement D (*decrypt*) et la clef K^{-1} (qui peut être égale à K)

$$M = D(C, K^{-1})$$



La cryptanalyse

Définition

Définition (Cryptanalyse)

La *cryptanalyse* (analyse du secret) est la science qui consiste à tenter de trouver un message ayant été protégé par une technique de cryptographie

Des techniques diverses et variées

- ▶ Attaque par force brute
- ▶ Analyse fréquentielle, linéaire, différentielle, ...
- ▶ ...

Classes d'attaques

- ▶ *ciphertext-only* : Oscar connaît uniquement un chiffré C
- ▶ *known-plaintext* : Oscar connaît un message M et le chiffré C associé
- ▶ *chosen-plaintext* : Oscar choisit un message M ...
- ▶ *chosen-ciphertext* : Oscar choisit un chiffré C et obtient M associé

Chiffre, chiffrement, déchiffrement : vocabulaire

Définition (Chiffre)

Un **chiffre** est un code secret permettant de transcrire un texte clair (libellé) en un texte chiffré (cryptogramme), généralement à l'aide d'une **clef** de chiffrement

Définition (Chiffrement)

Le **chiffrement** est la réalisation de cette transcription. On **chiffre**

Définition (Déchiffrement)

Le **déchiffrement** est l'opération inverse du chiffrement. On **déchiffre**

Définition (Décryptage)

Le **décryptage** consiste à trouver un texte clair à partir d'un texte chiffré, **sans** la clef de chiffrement. On **décrypte**. *Cryptage* et *crypter* n'existent pas !

Définition (Cryptologie (science du secret))

La *cryptologie* est l'ensemble constitué de la cryptographie et de la cryptanalyse

Plan



Introduction générale au chiffrement

Terminologie

Grands principes en cryptologie

Chiffres symétriques

Attaque par force brute



Principe de Kerckhoffs¹



Principe de Kerckhoffs

1. Un cryptogramme ne doit pas pouvoir être déchiffré sans la clé
2. Les clés doivent être simples et modifiables
3. Les cryptogrammes doivent être transportables, c'est-à-dire “télégraphiables”
4. L'appareil de codage et les documents doivent être transportables
5. Le système doit être simple d'utilisation
6. Le système de chiffrement doit être au préalable examiné par des experts

1. La cryptographie militaire, dans *Journal des sciences militaires*, Auguste Kerckhoffs, 1883



Maxime de Shannon²

Maxime de Shannon

«L'adversaire connaît le système.»

Dit autrement...

- ▶ La sécurité repose sur le secret de la clef et non sur le secret de l'algorithme
- ▶ Le déchiffrement sans la clef doit être impossible
- ▶ Trouver la clef à partir du texte clair et du texte chiffré est impossible

Confusion et diffusion

- ▶ *Confusion* : relation la plus complexe possible entre la clef de chiffrement et le cryptogramme
- ▶ *Diffusion* : dissipation de la redondance statistique d'un texte clair dans le cryptogramme Un bit changé en entrée doit changer chaque bit en sortie avec une probabilité $\frac{1}{2}$ → effet d'avalanche

2. *Communication Theory of Secrecy Systems*, Claude Shannon, 1949

Plan



Introduction générale au chiffrement

Chiffres symétriques

Chiffre de Vernam : élégant, utile ?

Chiffre Advanced Encryption Standard : référence par block

Attaque par force brute



Plan



Introduction générale au chiffrement

Chiffres symétriques

Chiffre de Vernam : élégant, utile ?

Chiffre Advanced Encryption Standard : référence par block

Attaque par force brute



Chiffre de Vernam, principe, propriété



Message M et clef K parfaitement aléatoire et de même longueur

- ▶ Le message chiffré C est : $C = M \oplus K$
- ▶ Déchiffrement : $M = C \oplus K$
- ▶ La clef K : jetée après le chiffrement (appelé aussi masque jetable)

Chiffrement inconditionnellement sûr

- ▶ Comme la clef est parfaitement aléatoire, toute clef est équiprobable, et donc elle n'apporte aucune information au message
- ▶ La relation entre le texte clair, le texte chiffré et la clef ne permet pas d'obtenir d'information
- ▶ On en déduit, que la connaissance du texte chiffré n'apporte aucune information sur le texte clair

Conséquences

- ▶ Attaque par force brute : impossible de trouver le texte original !
- ▶ Mais clef aussi longue que le message à transmettre : intérêt ?



Plan



Introduction générale au chiffrement

Chiffres symétriques

Chiffre de Vernam : élégant, utile ?

Chiffre Advanced Encryption Standard : référence par block

Attaque par force brute



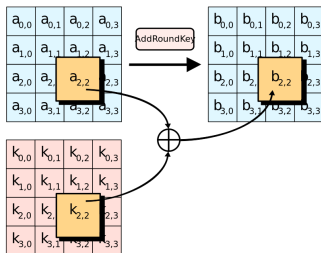
AES³ : chiffrement par bloc de 128b.

Chiffrement pr. clef K de 128b. et message M de taille multiple de 128b.

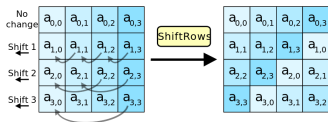
1. Message M et clef K : comme des tableaux 4×4 B.
2. KeyExpansion : génération des 11 clefs $K^0 \dots, K^{10}$ de 128b
3. Préliminaire : AddRoundKey sur l'état initial $XOR(a^0, K^0)$
4. Pour t de 0 à 8 :
 - 4.1 SubBytes : $\forall i, j$, application d'une fonction $a_{i,j}^t = S(a_{i,j}^t)$ non linéaire
 - 4.2 ShiftRows : décalage des cellules, ligne par ligne $a_i^t = a_i^t \ll i$
 - 4.3 MixColumns : produit matriciel, colonne par colonne $a_j^t = C.a_j^t$
 - 4.4 AddRoundKey $a^{t+1} = XOR(a^t, K^{t+1})$
5. Tour final :
 - 5.1 SubBytes
 - 5.2 ShiftRows
 - 5.3 AddRoundKey $a^{10} = XOR(a^9, K^{10})$

3. Daemen, Joan, & Rijmen, Vincent (2002). The design of Rijndael : AES-the advanced encryption standard. In Information Security and Cryptography. springer.

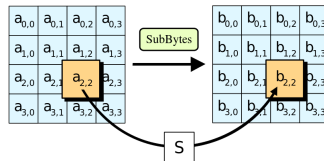
AES : en schéma ⁴



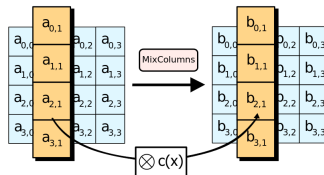
AddRoundKey



ShiftRows



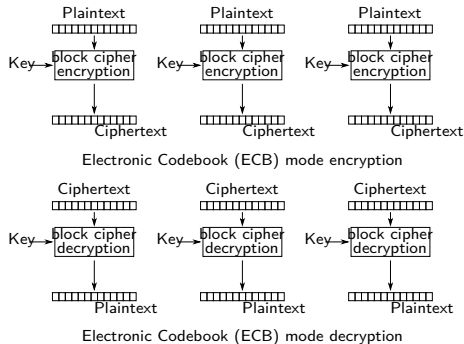
SubBytes



MixColumns

4. https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

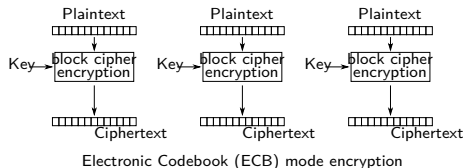
Mode Electronic CodeBook (ECB)⁵



- ▶ Chaque bloc de 128b. chiffré séparément : $C_i = E(M_i, K)$
- ▶ Chaque bloc de 128b. déchiffré séparément : $M'_i = D(C_i, K)$

5. https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

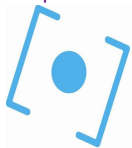
Mode ECB : analyse du cas d'école



Inconvénients $\rightsquigarrow$ ne PAS utiliser

- Taille du message nécessairement multiple de 128b ? Solution ?
- 2 blocs identique $M_1 = M_2 \rightsquigarrow$ chiffrés identiques $C_1 = C_2$

Exemple : chiffré d'une image régulière

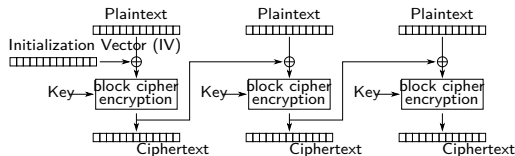


ISIFC

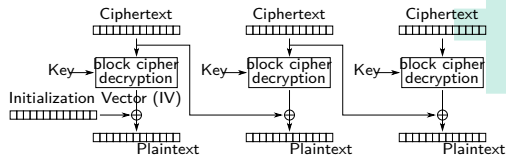
génie biomédical



Mode Cipher Block Chaining⁷ (CBC)



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

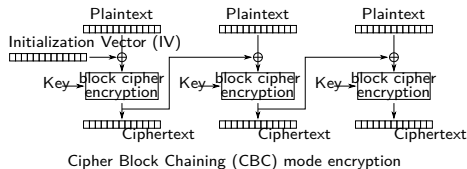
6

- ▶ IV : vecteur d'initialisation aléatoire, partagé en clair
- ▶ Mélange de chaque bloc M_i avec le chiffré C_{i-1} précédent : $C_0 = IV$ puis $C_i = E(M_i \oplus C_{i-1}, K)$
- ▶ Déchiffrement : $C_0 = IV$ puis $M'_i = D(C_i, K) \oplus C_{i-1}$

6. [https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Cipher_block_chaining_\(CBC\)](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Cipher_block_chaining_(CBC))

7. William F. Ehsam, Carl H. W. Meyer, John L. Smith, Walter L. Tuchman, "Message verification and transmission error detection by block chaining", US Patent 4074066, 1976.

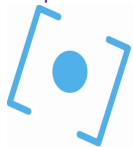
Mode CBC : analyse de cette amélioration



Inconvénients, avantages

- ▶ - Pas parallélisable au chiffrement : 1 seul bit de changé dans $M_1 \rightsquigarrow$ tout recalculer
- ▶ + Parallélisable au déchiffrement : on dispose de tous les C_i
- ▶ + Meilleure diffusion qu'ECB : IV erroné au déchiffrement $\rightsquigarrow$ seul M'_1 est faux

Exemple : chiffre d'une image régulière



ISIFC

g é n i e B i o m é d i c a l



Mode CBC : code avec pycryptodome



```
from base64 import b64encode
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
from Crypto.Random import get_random_bytes
```

```
data = "isifc".encode('utf-8')
key = get_random_bytes(16)
with open(path+'aeskey', 'wb') as f:
    f.write(key)
```

```
cipher = AES.new(key, AES.MODE_CBC)
ct_bytes = cipher.encrypt(pad(data, AES.block_size))
```

```
iv = b64encode(cipher.iv).decode('utf-8')
ct = b64encode(ct_bytes).decode('utf-8')
```

```
json_string = json.dumps({'iv':iv, 'ciphertext':ct})
with open(path+'json_data.json', 'w') as outfile:
    outfile.write(json_string)
print(json_string)
```

```
{ "iv": "cOzSOIKdI51Yy6GrL7BGUQ==",
  "ciphertext": "EQ8KaZnFYPKRywZ1YpTTZA=="}
```

```
import json
from base64 import b64decode
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad
```

```
with open(path+'aeskey', 'rb') as f:
    key = f.read()
```

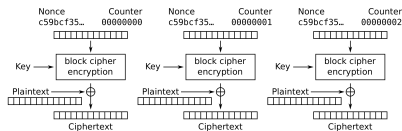
```
with open(path+'json_data.json') as json_file:
    b64 = json.load(json_file)
```

```
iv = b64decode(b64['iv'])
ct = b64decode(b64['ciphertext'])
cipher = AES.new(key, AES.MODE_CBC, iv)
```

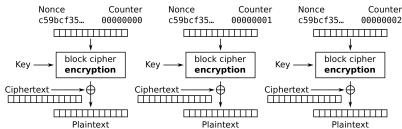
```
pt = unpad(cipher.decrypt(ct), AES.block_size)
print("The message was: ", pt.decode('utf-8'))
```

The message was: isifc

Mode CounTeR⁸ (CTR) : Diffie, Hellman 1979



Counter (CTR) mode encryption

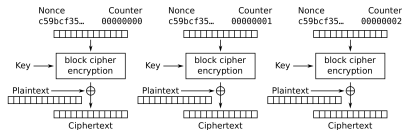


Counter (CTR) mode decryption

- ▶ Nonce : vecteur unique, peut être prévisible
- ▶ Mélange XOR du chiffrement de (Nonce||Counter) selon K avec M_i :
$$C_i = E((\text{Nonce}||\text{Counter}_i), K) \oplus M_i$$
- ▶ Déchiffrement : $M'_i = E((\text{Nonce}||\text{Counter}_i), K) \oplus C_i$

8. [https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Counter_\(CTR\)](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Counter_(CTR))

Mode CTR : analyse de cette amélioration



Inconvénients, avantages

- ▶ - IV erroné $\rightsquigarrow$ tout M' est faux
- ▶ + Parallélisable ;

Exemple : chiffrer d'une image régulière



Modes combinant intégrité et confidentialité



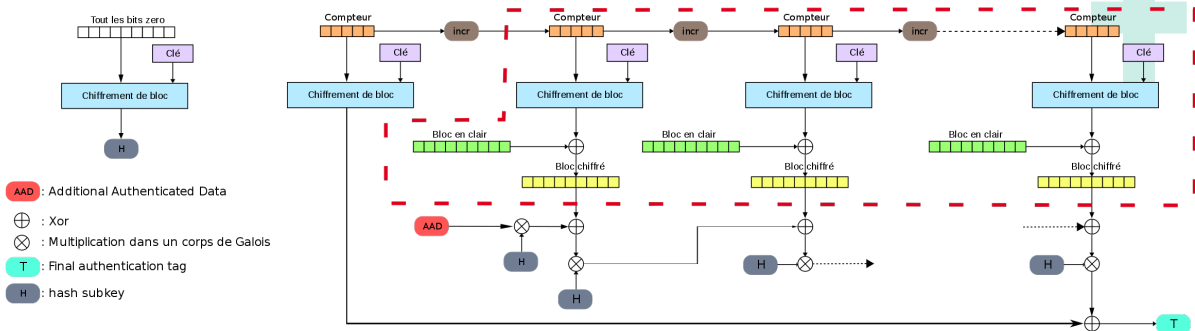
Motivation

- ▶ Modes précédents : confidentialité mais pas intégrité (impossible pour le · la destinataire d'établir si le message chiffré a été modifié)
- ▶ Apparition d'une autre famille de modes opératoires (2003) AEAD (Authenticated Encryption with Associated Data) combinant chiffrement et intégrité en une primitive unique
- ▶ Galois/Counter Mode (GCM), Counter with CBC-MAC (CCM)
- ▶ Nécessité de transférer en + des données pour l'intégrité



Galois/Counter Mode (GCM)

Chiffrement en image⁹



Analyse

- Intégrité : génération d'un TAG d'authentification à partir de d'AAD
- Inconvénients, avantages algorithmiques : comme CTR

9. https://fr.wikipedia.org/wiki/Galois/Counter_Mode

Mode GCM : code avec pycryptodome

```
import json
from base64 import b64encode
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
from Crypto.Random import get_random_bytes
```

```
data = "isifc".encode('utf-8')
header = "entête".encode('utf-8')
key = get_random_bytes(16)
with open(path+'aeskey', 'wb') as f:
    f.write(key)
cipher = AES.new(key, AES.MODE_GCM)
cipher.update(header)
```

```
ciphertext, tag = cipher.encrypt_and_digest(data)
```

```
json_k = [ 'nonce', 'header', 'ciphertext', 'tag' ]
json_v = [ b64encode(x).decode('utf-8')
           for x in (cipher.nonce, header, ciphertext, tag)]
```

```
json_string = json.dumps(dict(zip(json_k, json_v)))
with open(path+'json_data.json', 'w') as outfile:
    outfile.write(json_string)
print(json_string)
```

```
{"nonce": "TRzMWR11WUKqAaIJDmDfSw==",
 "header": "ZW50w6pOZSBxdWVsY29ucXV1",
 "ciphertext": "BvP9JVE=",
 "tag": "fMd65haAYwdEptYGm7t8Tg=="}
```

```
import json
from base64 import b64decode
from Crypto.Cipher import AES
```

```
with open(path+'aeskey', 'rb') as f:
    key = f.read()
```

```
with open(path+'json_data.json') as json_file:
    b64 = json.load(json_file)
```

```
json_k = [ 'nonce', 'header', 'ciphert', 'tag' ]
json_dict = {k:b64decode(b64[k]) for k in json_k}
```

```
try:
    cipher = AES.new(key, AES.MODE_GCM,
                     nonce=json_dict['nonce'])
    cipher.update(json_dict['header'])
    pt = cipher.decrypt_and_verify(json_dict['ciphert'],
                                   json_dict['tag'])
    print("The message was: ", pt.decode('utf-8'))
except (ValueError, KeyError):
    print("Incorrect decryption")
```

The message was: isifc

Ordres de grandeur d'une attaque par force brute sur AES-128

- ▶ **Hypothèses matérielles et logicielles** : utilisation d'un processeur actuel délivrant 240 000 Mips, et un algorithme nécessitant 1200 instructions élémentaires pour tester chaque clé
- ▶ **Durée d'évaluation d'une seule clé** : 5×10^{-9} seconde
- ▶ **Clés à tester et durée totale** : une attaque requiert de tester en moyenne la moitié de l'espace des clés (2^{127}) clés, i.e. $2,7 \times 10^{22}$ années
- ▶ **Parallélisation de l'attaque** : $2,7 \times 10^9$ processeurs en parallèle aboutit à 10^{13} années

