

# TABLE DES MATIERES

## REMERCIEMENTS

## LISTE DES ABBREVIATIONS..... IV

## NOTATIONS.....X

## INTRODUCTION GENERALE..... 1

## CHAPITRE 1 : LES RESEAUX WIFI ..... 3

|                                                                    |    |
|--------------------------------------------------------------------|----|
| 1.1. INTRODUCTION.....                                             | 3  |
| 1.1.1. Généralités sur les réseaux sans fil.....                   | 3  |
| 1.1.1.1. Les réseaux WPAN .....                                    | 3  |
| 1.1.1.2. Les réseaux WLAN .....                                    | 4  |
| 1.1.1.3. Les réseaux WMAN .....                                    | 4  |
| 1.1.1.4. Les réseaux WWAN .....                                    | 4  |
| 1.1.2. Avantages et inconvénients des réseaux sans fil.....        | 4  |
| 1.2. ARCHITECTURE DES RESEAUX WIFI .....                           | 5  |
| 1.2.1. Architecture cellulaire et type de topologie.....           | 5  |
| 1.2.1.1. Mode infrastructure .....                                 | 6  |
| 1.2.1.2. Mode Ad-hoc .....                                         | 7  |
| 1.2.2. Le partage du canal radio.....                              | 8  |
| 1.3. LES MODELES DE COUCHE .....                                   | 9  |
| 1.3.1. La couche physique (PHY) .....                              | 10 |
| 1.3.1.1. Le Frequency Hopping Spread Spectrum (FHSS).....          | 10 |
| 1.3.1.2. Le Direct Sequence Spread Spectrum (DSSS) .....           | 11 |
| 1.3.1.3. L'Orthogonal Frequency Division Multiplexing (OFDM) ..... | 12 |
| 1.3.2. Les évolutions du 802.11 .....                              | 14 |
| 1.3.2.1. Le 802.11a .....                                          | 14 |
| 1.3.2.2. Le 802.11b .....                                          | 15 |
| 1.3.2.3. Le 802.11g .....                                          | 15 |
| 1.3.2.4. Le 802.11n.....                                           | 16 |
| 1.3.3. La norme 802.11n.....                                       | 16 |
| 1.3.3.1. La diversité spatiale.....                                | 16 |
| 1.3.3.2. Principe de la technique MIMO .....                       | 16 |
| 1.3.3.3. Avantages du MIMO dans les réseaux haut débit.....        | 17 |
| 1.3.4. La couche Liaison de données .....                          | 19 |
| 1.3.4.1. Le protocole CSMA/CA.....                                 | 20 |
| 1.3.4.2. Les méthodes d'accès au support.....                      | 21 |
| 1.3.4.3. Propriétés supplémentaires des couches MAC et LLC .....   | 22 |
| 1.3.4.4. Les évolutions de la couche MAC.....                      | 22 |
| 1.4. LE FORMAT DES TRAMES .....                                    | 23 |
| 1.4.1. Les trames de la couche PHY .....                           | 23 |
| 1.4.1.1. Le préambule.....                                         | 24 |
| 1.4.1.2. L'en – tête PLCP .....                                    | 24 |
| 1.4.2. Les trames MAC.....                                         | 24 |
| 1.5. CONCLUSION .....                                              | 26 |

## CHAPITRE 2 : LA SECURITE DES RESEAUX WIFI ..... 27

|                                         |    |
|-----------------------------------------|----|
| 2.1. INTRODUCTION.....                  | 27 |
| 2.1.1. Les causes de l'insécurité ..... | 27 |
| 2.1.2. Qui sont les attaquants ? .....  | 27 |

|                                                                                        |           |
|----------------------------------------------------------------------------------------|-----------|
| 2.1.3. Motivations et but des agresseurs .....                                         | 28        |
| 2.1.4. Comment se protéger ? .....                                                     | 28        |
| 2.2. LES PRINCIPALES ATTAQUES CONTRE LE RESEAU SANS FIL .....                          | 28        |
| 2.2.1. Le war-driving .....                                                            | 28        |
| 2.2.2. L'intrusion .....                                                               | 29        |
| 2.2.3. Le déni de service .....                                                        | 30        |
| 2.2.3.1. Le flooding .....                                                             | 30        |
| 2.2.3.2. Le smurf .....                                                                | 30        |
| 2.2.3.3. Le débordement de tampon .....                                                | 30        |
| 2.2.4. Le brouillage radio .....                                                       | 30        |
| 2.2.5. L'interception de données .....                                                 | 30        |
| 2.3. LES DIFFERENTES SOLUTIONS DE SECURITE .....                                       | 31        |
| 2.3.1. Le chiffrement des trames .....                                                 | 31        |
| 2.3.1.1. Le Wired Equivalent Privacy (WEP) .....                                       | 31        |
| 2.3.1.2. Le WPA (Wi-Fi Protected Access) et le WPA2 .....                              | 35        |
| 2.3.1.3. Le IEEE 802.11i .....                                                         | 42        |
| 2.3.2. Mécanisme d'authentification .....                                              | 42        |
| 2.3.2.1. Le IEEE 802.1x .....                                                          | 43        |
| 2.3.3. Autres solutions .....                                                          | 45        |
| 2.3.3.1. Mise en place d'un Virtual Private Network .....                              | 45        |
| 2.3.3.2. Installer des firewalls .....                                                 | 45        |
| 2.3.3.3. Le filtrage d'adresse MAC .....                                               | 46        |
| 2.4. CONCLUSION .....                                                                  | 46        |
| <b>CHAPITRE 3 : THEORIE DE LA CRYPTOGRAPHIE .....</b>                                  | <b>47</b> |
| 3.1. DEFINITIONS .....                                                                 | 47        |
| 3.2. CRYPTOGRAPHIE CONVENTIONNELLE .....                                               | 48        |
| 3.2.1. Chiffrement et déchiffrement .....                                              | 48        |
| 3.2.2. Décryptement .....                                                              | 49        |
| 3.2.3. Chiffrement par bloc .....                                                      | 49        |
| 3.2.3.1. Principe .....                                                                | 49        |
| 3.2.3.2. Mode de chiffrement .....                                                     | 50        |
| 3.2.4. Chiffrement par flot .....                                                      | 54        |
| 3.2.5. Intégrité et authenticité .....                                                 | 55        |
| 3.3. LE CHIFFREMENT PAR SUBSTITUTION .....                                             | 55        |
| 3.3.1. Chiffrement simple .....                                                        | 55        |
| 3.3.2. Chiffrement par transposition .....                                             | 56        |
| 3.4. LE CHIFFREMENT A CLE SECRETE .....                                                | 56        |
| 3.5. LE CHIFFREMENT A CLE PUBLIQUE .....                                               | 57        |
| 3.5.1. Notion de clé de session .....                                                  | 58        |
| 3.5.1.1. Intérêt d'une clé de session .....                                            | 58        |
| 3.5.1.2. Algorithme de Diffie-Hellman .....                                            | 59        |
| 3.6. L'ALGORITHME RC4 .....                                                            | 59        |
| 3.6.1. Initialisation de l'algorithme RC4 .....                                        | 59        |
| 3.6.2. Génération des clés RC4 .....                                                   | 60        |
| 3.6.3. Clés faibles générées par RC4 .....                                             | 60        |
| 3.7. L'ALGORITHME AES (ADVANCED ENCRYPTION STANDARD) .....                             | 60        |
| 3.7.1. Fonctionnement .....                                                            | 61        |
| 3.7.2. Algorithme pour le calcul de contrôle d'intégrité Michael utilisé par AES ..... | 62        |
| 3.8. CONCLUSION .....                                                                  | 64        |
| <b>CHAPITRE 4 : SIMULATION .....</b>                                                   | <b>65</b> |

|                                                                                                       |           |
|-------------------------------------------------------------------------------------------------------|-----------|
| <b>4.1. FONCTIONNEMENT DU LOGICIEL .....</b>                                                          | <b>66</b> |
| 4.1.1. Ouverture d'un fichier .....                                                                   | 67        |
| 4.1.2. Affichage des propriétés du fichier .....                                                      | 68        |
| 4.1.3. Chiffrement en utilisant l'algorithme de cryptage RC4.....                                     | 69        |
| 4.1.4. Affichage de la structure du paquet.....                                                       | 70        |
| 4.1.4.1. Paquet (MPDU) WEP.....                                                                       | 71        |
| 4.1.4.2. Paquet (MPDU) WPA.....                                                                       | 71        |
| 4.1.5. Déchiffrement avec l'algorithme RC4 .....                                                      | 71        |
| 4.1.6. Chiffrement/déchiffrement d'un fichier (texte) en utilisant l'algorithme de cryptage AES ..... | 72        |
| 4.1.7. Quitter le logiciel .....                                                                      | 74        |
| <b>4.2. CONCLUSION .....</b>                                                                          | <b>74</b> |
| <b>CONCLUSION GENERALE .....</b>                                                                      | <b>75</b> |
| <b>ANNEXE 1 : PRESENTATION DU LOGICIEL ECLIPSE .....</b>                                              | <b>76</b> |
| <b>ANNEXE 2 : CODE SOURCE.....</b>                                                                    | <b>78</b> |
| <b>BIBLIOGRAPHIE .....</b>                                                                            | <b>82</b> |
| <b>PAGE « RENSEIGNEMENTS » .....</b>                                                                  | <b>84</b> |
| <b>RESUME.....</b>                                                                                    | <b>85</b> |
| <b>ABSTRACT .....</b>                                                                                 | <b>85</b> |

## **LISTE DES ABBREVIATIONS**

|         |                                                        |
|---------|--------------------------------------------------------|
| 3G      | Troisième génération de la communication mobile        |
| 802.11  | Standard WLAN                                          |
| 802.16  | Standard WMAN                                          |
| ACK     | Acknowledge, acknowledgement                           |
| ACL     | Access Control List                                    |
| AES     | Advanced Encryption Standard                           |
| AMPS    | Advanced Mobile Phone System                           |
| AP      | Access Point                                           |
| AWGN    | Additive White Gaussian Noise                          |
| BPSK    | Binary Phase Shift Keying                              |
| BSA     | Basic Service Area                                     |
| BSS     | Basic Service Set                                      |
| BSSID   | Basic Service Set Identity                             |
| CAM     | Continuous Aware Mode                                  |
| CBC     | Cipher Block Chaining                                  |
| CBC-MAC | Cipher Block Chaining-Message Authentication Code      |
| CCM     | Counter-Mode + CCM-MAC                                 |
| CCMP    | CBC-MAC Protocol                                       |
| CDMA    | Code Division Multiple Access                          |
| CFB     | Cipher FeedBack                                        |
| CM      | Counter Mode                                           |
| CRC     | Cyclical Redundancy Code                               |
| CSMA/CA | Carrier Sense Multiple Access with Collision Avoidance |

|         |                                                        |
|---------|--------------------------------------------------------|
| CSMA/CD | Carrier Sense Multiple Access with Collision Detection |
| CTAK    | CipherText Auto Key                                    |
| CTS     | Clear To Send                                          |
| DCF     | Distributed Coordination Function                      |
| DES     | Data Encryption Standard                               |
| DFS     | Dynamic Frequency Selection                            |
| DHCP    | Dynamic Host Control Protocol                          |
| DIFS    | Distributed Inter Frame Space                          |
| DoS     | Denial of Service                                      |
| DS      | Distribution System                                    |
| DSSS    | Direct Sequence Spread Spectrum                        |
| EAP     | Extensible Authentication Protocol                     |
| ECB     | Electronic Code Book                                   |
| ESA     | Extended Service Area                                  |
| ESS     | Extended Service Set                                   |
| ESSID   | Extended Service Set IDentity                          |
| ETSI    | European Telecommunications Standardization Institute  |
| FC      | Frame Control                                          |
| FCC     | Federal Communications Commission                      |
| FCS     | Frame Check Sequence                                   |
| FDDI    | Fiber Distributed Data Interface                       |
| FDMA    | Frequency Division Multiple Access                     |
| FHSS    | Frequency Hopping Spread Spectrum                      |
| FSC     | Frame Sequence Check                                   |
| FTP     | File Transfert Protocol                                |

|           |                                                   |
|-----------|---------------------------------------------------|
| GHz       | Gigahertz                                         |
| GPRS      | General Packet Radio Service                      |
| GPS       | Global Positionning System                        |
| GSM       | Global System for Mobile communication            |
| GTK       | Group Temporary Key                               |
| HiperLAN2 | High Performance Radio LAN 2.0                    |
| HR-DSSS   | High Rate - DSSS                                  |
| Hz        | Hertz                                             |
| IAPP      | Inter Access Point Protocol                       |
| IBSS      | Independant Basic Service Set                     |
| ICV       | Integrity Check Value                             |
| IDE       | Integrated Development Environment                |
| IEEE      | Institute of Electrical and Electronics Engineers |
| IP        | Internet Protocol                                 |
| IR        | Infrared                                          |
| ISI       | Intersymbol Interference                          |
| ISM       | Industrial, Scientific, and Medical (band)        |
| ISO       | International Organization for Standardization    |
| ITU       | International Telecommunication Union             |
| ITU-T     | ITU Telecommunications Standardization Sector     |
| IV        | Initialization Vector                             |
| J.-C.     | Jésus Christ                                      |
| KAK       | Key Auto Key                                      |
| kHz       | kilohertz                                         |
| LAN       | Local Area Network                                |

|      |                                                    |
|------|----------------------------------------------------|
| LLC  | Logical Link Control                               |
| LSB  | Least Significant Byte                             |
| MAC  | Media Access Control / Message Authentication Code |
| MAN  | Metropolitan Area Network                          |
| MAP  | Maximum A posteriori Probability                   |
| MAU  | Multistation Access Unit                           |
| Mbps | Megabits par seconde                               |
| MHz  | Megahertz                                          |
| MIC  | Message Integrity Code                             |
| MiM  | Man in the Middle                                  |
| MIMO | Multiple Input Multiple Output                     |
| MKK  | Kensa-kentei Kyokai                                |
| MPDU | Media Access Control Protocol Data Unit            |
| MSB  | Most Significant Byte                              |
| MSDU | Media Access Control Service Data Unit             |
| MSps | Million(s) de symboles par seconde                 |
| NAS  | Network Authentication Service                     |
| NIC  | Network Interface Controller                       |
| NMT  | Nordic Mobile Telephone                            |
| NSA  | National Security Agency                           |
| OFB  | Output FeedBack                                    |
| OFDM | Orthogonal Frequency Division Multiplexing         |
| OSI  | Open Systems Interconnection                       |
| OTP  | One Time Password                                  |
| PAN  | Personal Area Network                              |

|         |                                                         |
|---------|---------------------------------------------------------|
| PCF     | Point Coordination Function                             |
| PCMCIA  | Personal Computer Memory Card International Association |
| PDA     | Personal Digital Assistant                              |
| PDU     | Protocol Data Unit                                      |
| PEAP    | Protected EAP                                           |
| PHY     | se réfère à la couche physique du modèle OSI            |
| PLCP    | Physical Layer Convergence Procedure                    |
| PN      | Packet Number                                           |
| PSK     | Pre-Shared Key                                          |
| PSPM    | Power Save Polling Mode                                 |
| PTK     | Pre-Temporary Key                                       |
| QAM     | Quadrature Amplitude Modulation                         |
| QoS     | Quality of Service                                      |
| QPSK    | Quadrature Phase Shift Keying                           |
| RADIUS  | Remote Authentication Dial In User Service              |
| RC4     | Rivest Cipher 4                                         |
| RTS/CTS | Ready To Send / Clear To Send                           |
| SDM     | Spatial Diversity Multiplexing                          |
| SHA     | Secure Hash Algorithm                                   |
| SNMP    | Simple Network Management Protocol                      |
| SSL     | Secure Socket Layer                                     |
| STCB    | Space Time Code Block                                   |
| SWT     | Standard Widget Toolkit                                 |
| TCP     | Transfert Control Protocol                              |
| TDMA    | Time Division Multiple Access                           |

|       |                                                 |
|-------|-------------------------------------------------|
| TKIP  | Temporary Key Integrity Protocol                |
| TPC   | Transmit Power Control                          |
| TSC   | TKIP Sequence Counter                           |
| TTLS  | Tunneled Transport Layer Security               |
| UMTS  | Universal Mobile Telecommunication System       |
| UML   | Uniform Markup Language                         |
| URL   | Uniform Resource Locator                        |
| USB   | Universal Serial Bus                            |
| VPN   | Virtual Private Network                         |
| WAN   | Wide Area Network                               |
| WECA  | Wireless Ethernet Compatibility Alliance        |
| WEP   | Wired Equivalent Privacy                        |
| WiFi  | Wireless Fidelity                               |
| WiMAX | Worldwide Interoperability for Microwave Access |
| WLL   | Wireless Local Loop                             |
| WPA   | WiFi Protected Access                           |
| WSW   | Websphere Studio Workbench                      |

## NOTATIONS

|                 |                                                                |
|-----------------|----------------------------------------------------------------|
| $C, C', c, c_i$ | message crypté                                                 |
| $D_k$           | algorithme de déchiffrement                                    |
| $E, E_k$        | algorithme de chiffrement                                      |
| $H$             | fonction de hachage cryptographique                            |
| $K, k$          | clé de chiffrement                                             |
| $l$             | taille d'un bloc CFB                                           |
| $M, M', m, m_i$ | message original                                               |
| $\text{mod}$    | opérateur modulo                                               |
| $N$             | nombre de porteuses (OFDM)                                     |
| $R$             | séquence de bits pseudo-aléatoire générée par l'algorithme RC4 |
| $s(t)$          | signal OFDM                                                    |
| $T_s$           | durée des symboles (OFDM)                                      |
| $T_{si}$        | période d'un train de symboles (OFDM)                          |

## **INTRODUCTION GENERALE [1] [2]**

L'onde de choc engendré par l'apparition des technologies sans fil continue de se propager. Depuis quelques années, les solutions des réseaux locaux de type Wireless Fidelity (WiFi) ont été amplement adoptées tant par les utilisateurs à domicile que par les entreprises. En effet, le WiFi est une technologie de liaison informatique sans fil mise en place pour fonctionner en réseau interne et, depuis, devenue un moyen d'accès à l'internet haut débit. Il est basé sur la norme IEEE 802.11.

Dans la pratique, le WiFi permet de relier des ordinateurs portables, des machines de bureau, des assistants personnels (PDA) ou même des périphériques à une liaison haut débit sur un rayon de plusieurs dizaines de mètres en intérieur. Dans un environnement ouvert, la portée peut atteindre plusieurs centaines de mètres voire plusieurs dizaines de kilomètres avec une variante WiMAX ou avec des antennes directionnelles. La norme 802.11g est la plus répandue dans le commerce actuellement ; elle offre un débit de 54 Mbits/s sur la bande de fréquences des 2.4 GHz.

La technologie WiFi est une solution plus qu'idéale pour les prestataires de services comme les centres de vacances, les campings, l'hôtellerie, les gares et les aéroports ; mais aussi pour les structures qui destinent leurs locaux à des congrès, des séminaires ou toutes autres réunions professionnelles. Bien que ces réseaux offrent des avantages comme une circulation libre dans les locaux avec les ordinateurs portables sans traîner des câbles et une grande simplicité de mise en œuvre, les diverses transmissions se font à l'air libre par l'intermédiaire des ondes électromagnétiques. Il est donc très difficile d'arriver à confiner les émissions des ondes radio dans un périmètre restreint, ce qui fait qu'il est très facile pour une personne non autorisée d'écouter le réseau. On peut dire donc que du point de vue de la sécurité, les solutions sans fil sont assez vulnérables par rapport aux réseaux filaires. De ce fait, des personnes mal intentionnées peuvent en profiter pour nuire à ces réseaux sans fil.

Actuellement, on recense un certain nombre d'attaques dont les réseaux WiFi font l'objet. Citons par exemple l'interception de données, l'intrusion réseau, le brouillage radio ou encore les dénis de service. La méthode la plus utilisée pour ce faire est le war-driving qui consiste à circuler dans la ville avec un ordinateur portable équipé d'une carte réseau sans fil à la recherche de réseaux sans fil.

Ces attaques peuvent être dangereuses, surtout dans le cas où il s'agit d'un réseau d'entreprise. De ce fait, il est donc primordial de sécuriser son réseau sans fil. La sécurité étant un point fondamental dans le déploiement d'un réseau sans fil, nous avons donc choisi de nous y intéresser particulièrement, d'où le titre de ce mémoire « LE CHIFFREMENT DES DONNEES DANS LES RESEAUX WIFI ».

Nous allons donc commencer par quelques généralités sur les réseaux. Nous parlerons ensuite des réseaux locaux sans fil, en particulier du WiFi. Puis, nous parlerons des attaques dont font l'objet les réseaux WiFi ainsi que des solutions de sécurité adoptées jusqu'alors. Nous terminerons enfin par simuler le déroulement du chiffrement d'une trame destinée à être transmise sur un réseau.

## CHAPITRE 1 : LES RESEAUX WIFI

### 1.1. Introduction [2] [4] [9] [10]

Le terme « réseau » définit un ensemble d'entités (objets, personnes, ...) interconnectées les unes avec les autres. Un réseau permet ainsi de faire circuler des éléments matériels entre chacune de ces entités selon des règles bien définies.

Un réseau ou *network* est un ensemble d'ordinateurs et de périphériques connectés les uns aux autres.

#### 1.1.1. Généralités sur les réseaux sans fil

Un réseau sans fil (en anglais, *wireless network*) est, par définition, un réseau dans lequel deux terminaux (ordinateur, PDA, ...) peuvent communiquer sans liaison filaire. Grâce à un réseau sans fil, un utilisateur a la possibilité de rester connecté tout en se déplaçant dans un périmètre géographique plus ou moins étendu.

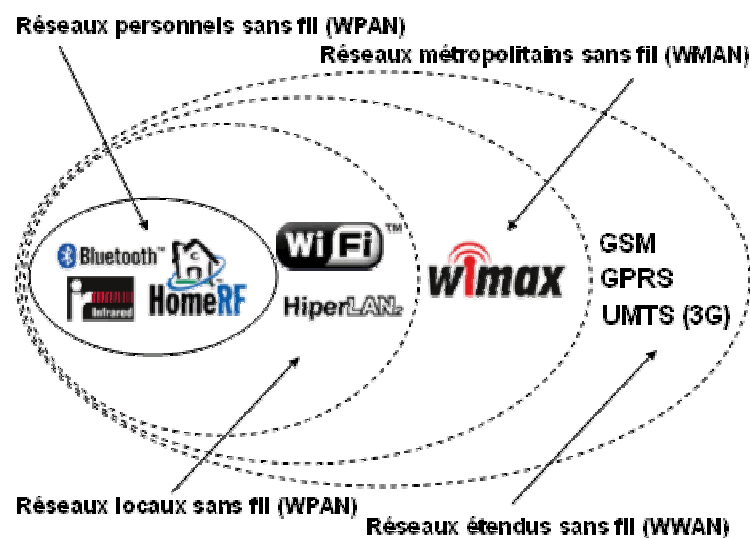


Figure 1.01 : classification des réseaux sans fil

Comme pour les réseaux filaires, on distingue aussi différents types de réseaux sans fil selon leur taille, leur vitesse de transfert de données ainsi que leur étendue.

##### 1.1.1.1. Les réseaux WPAN

Le réseau personnel sans fil concerne les réseaux sans fil d'une faible portée (quelques dizaines de mètres). Ce type de réseau sert généralement à relier des périphériques (imprimante, téléphone

portable, ...) ou un assistant personnel (PDA : *Personal Digital Assistant*) à un ordinateur sans liaison filaire, ou bien à permettre la liaison sans fil entre deux machines très peu distantes.

Quelques exemples de WPAN: Bluetooth, homeRF, ZigBee, infrarouge

#### 1.1.1.2. Les réseaux WLAN

Le *réseau local sans fil* (noté **WLAN** pour *Wireless Local Area Network*) est un réseau permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de mètres. Il permet de relier entre eux les terminaux présents dans la zone de couverture. Il existe plusieurs technologies concurrentes : le WiFi, l'hiperLAN2.

#### 1.1.1.3. Les réseaux WMAN

Le *réseau métropolitain sans fil* (WMAN : *Wireless Metropolitan Area Network*) est connu sous le nom de **Boucle Locale Radio** (BLR). Les WMAN sont basés sur la norme *IEEE 802.16*. La boucle locale radio offre un débit utile de 1 à 10 Mbit/s pour une portée de 4 à 10 kilomètres, ce qui destine principalement cette technologie aux opérateurs de télécommunication.

La norme de réseau métropolitain sans fil la plus connue est le WiMAX, permettant d'obtenir des débits de l'ordre de 70 Mbit/s sur un rayon de plusieurs kilomètres.

#### 1.1.1.4. Les réseaux WWAN

Le *réseau étendu sans fil* (WWAN : *Wireless Wide Area Network*) est également connu sous le nom de *réseau cellulaire mobile*. Il s'agit des réseaux sans fil les plus répandus puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fil. Les principales technologies sont les suivantes : le GSM, le GPRS, l'UMTS.

### **1.1.2. Avantages et inconvénients des réseaux sans fil**

Un des avantages les plus marquants des réseaux sans fil est sans aucun doute leur mobilité. De plus, il est très facile de les installer contrairement aux réseaux filaires et ils sont très économiques (du point de vue consommation énergétique).

Le principal inconvénient des réseaux sans fil relève de la sécurité. En effet, étant donné que les données sont transmises via des ondes hertziennes dont on ne maîtrise pas la portée exacte, des personnes mal intentionnées pourraient capter les communications, même de l'extérieur de

l'enceinte de l'organisme. De plus, les informations accessibles via la technologie sans fil peuvent aussi être sujettes à des attaques sur leur intégrité. Il est ainsi possible que des personnes mal intentionnées se procurent un accès illicite aux machines reliées via le réseau WiFi et modifient les données stockées sur ces postes. Puis, il est techniquement facile de lancer des attaques de déni de service sur des réseaux à technologie sans fil non protégés. Il suffit de bombarder les stations de base, pour immobiliser le réseau. A part la question de sécurité, on remarque aussi une chute rapide des débits au fur et à mesure que la distance augmente. La portée des réseaux locaux sans fil est réduite et il est préférable de les utiliser dans des zones découvertes (sans obstacles).

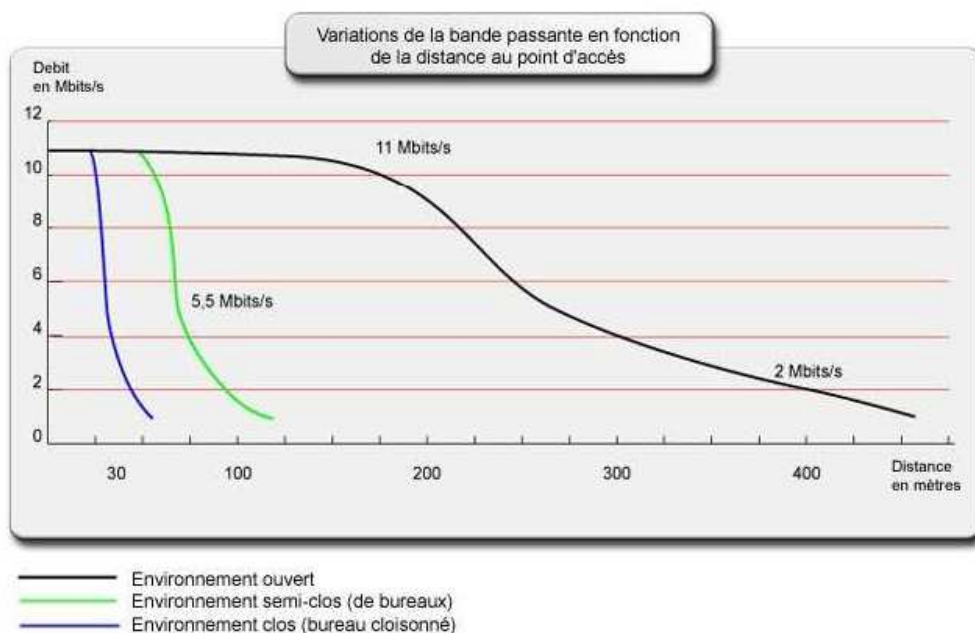


Figure 1.02 : Variation de la bande passante en fonction de la distance au point d'accès

Certains appareils comme les téléphones sans fil ou les fours à microondes utilisent la même bande de fréquences à savoir 2.4Ghz. Ils peuvent donc créer des interférences et perturber le réseau 802.11.

## 1.2. Architecture des réseaux WiFi [2] [3] [5] [11] [12]

### 1.2.1. Architecture cellulaire et type de topologie

La norme IEEE 802.11 est basée sur une architecture cellulaire que l'on peut comparer à celle utilisée dans la téléphonie mobile. Il existe deux topologies possibles dans les réseaux sans fil IEEE 802.11 :

- Le mode infrastructure,
- Le mode ad-hoc.

#### 1.2.1.1. Mode infrastructure

Dans les réseaux de type Infrastructure, chaque périphérique est relié au réseau via un point d'accès (AP) WiFi. On dit que le périphérique est le *client* et l'AP le *maître*. Un réseau de ce type s'appelle un *Basic Service Set* (BSS) et couvre un espace qu'on appelle une *cellule* ou *Basic Service Area* (BSA).

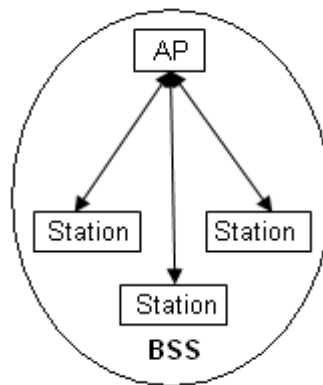


Figure 1.03 : Basic Service Set

Chaque BSS est identifié par un nombre composé de 48 bits : c'est le BSSID. En mode Infrastructure, ce BSSID correspond tout simplement à l'adresse MAC du point d'accès. L'AP sert de relais entre les périphériques, mais il peut aussi servir de relais vers un réseau filaire, par exemple un réseau d'entreprise.

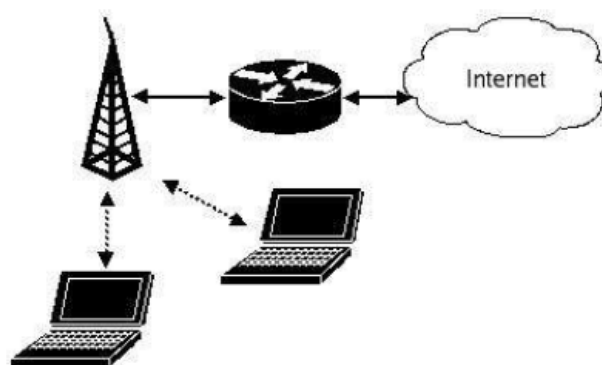


Figure 1.04 : Réseau WiFi minimal

Plusieurs points d'accès peuvent être déployés pour atteindre une plus large couverture WiFi. Ces BSS multiples peuvent être reliés par un système de distribution (*Distribution System*, DS) de façon à former un unique réseau sans fil. Le DS peut être un réseau filaire Ethernet, un câble de point à point, ou encore une liaison sans fil. Il est alors possible à un utilisateur de se déplacer

dans l'ensemble de la zone de couverture sans souffrir de ralentissement ou d'interruption de sa connexion : en cas de besoin, la liaison bascule automatiquement vers le point d'accès offrant la meilleure connexion (c'est le *hand-over*). On parle dans ce cas d'*Extended Service Set* (ESS) qui couvre naturellement un espace appelé l'*Extended Service Area* (ESA), composé de plusieurs cellules. Chaque ESS est identifié par un nom de 32 caractères maximum qui s'appelle l'ESSID ou simplement SSID.

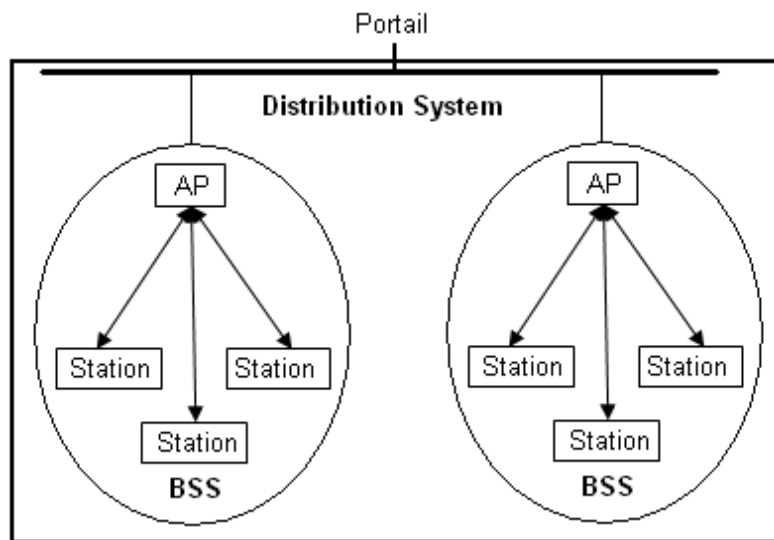


Figure 1.05: Extended Service Set

#### 1.2.1.2. Mode Ad-hoc

Dans les réseaux de type Ad Hoc, chaque périphérique communique directement avec les périphériques situés à sa portée, sans passer par un intermédiaire. Ce mode est pratique pour l'échange de données entre quelques stations en l'absence d'une quelconque infrastructure réseau (aucun point d'accès). Le réseau ainsi constitué s'appelle un *Independant Basic Service Set* (IBSS).

Malheureusement, le mode Ad Hoc a deux inconvénients majeurs. Premièrement, il peut parfois être assez pénible à configurer. En effet, imaginons quelques personnes qui souhaitent simplement s'échanger des fichiers avec le mode Ad Hoc.

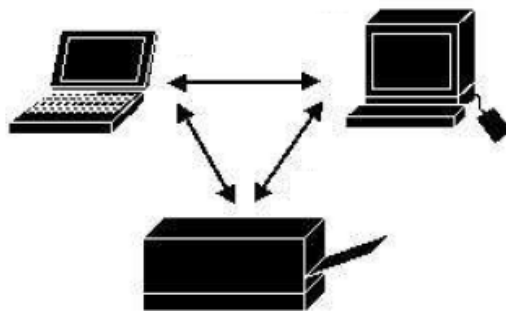


Figure 1.06: Un réseau ad-hoc avec deux ordinateurs et une imprimante.

Dans ce cas, l'un d'entre eux pourra configurer son adaptateur WiFi en mode Ad Hoc, en précisant le canal WiFi et le SSID à utiliser. Les autres pourront alors s'associer à ce réseau Ad Hoc en le détectant avec leur adaptateur WiFi, ou bien en configurant eux-mêmes leur propre adaptateur WiFi. Ensuite, à moins que l'un d'entre eux n'ait configuré un serveur DHCP sur son ordinateur, ils devront se mettre d'accord sur une configuration IP et configurer leur système d'exploitation en conséquence. Ensuite, il leur faudra encore activer le partage de fichier ou démarrer un serveur FTP. Deuxièmement, ce mode Ad Hoc ne dit pas comment deux stations peuvent communiquer entre elles par l'intermédiaire d'un troisième. Pour gérer cela, il faut installer des logiciels spécifiques sur chaque machine, c'est-à-dire gérer le routage des paquets à un niveau supérieur dans les couches OSI. Ces logiciels sont souvent capables d'appliquer des règles de routage complexes pour acheminer les paquets d'une station à une autre en passant par le chemin le plus efficace, c'est-à-dire le plus court, ou le moins encombré, malgré les déplacements éventuels des stations les unes par rapport aux autres. On obtient alors ce qu'on appelle un réseau « maillé » (*mesh network*). L'inconvénient est qu'il faut déployer le même logiciel sur chaque périphérique car il n'existe pas encore de standard pour les réseaux maillés sur 802.11.

### **1.2.2. Le partage du canal radio**

Toutes les variantes du WiFi découpent la bande de fréquences sur laquelle elles reposent (2.4 GHz ou 5 GHz) en canaux. Ils sont différents selon les variantes utilisées.

Le 802.11 FHSS utilise la bande de 2.4 GHz et la découpe en canaux de 1 MHz numérotés à partir de 2400 MHz. Les canaux utilisables changent en fonction de la législation du pays où l'on se trouve, mais en deux mots on a droit aux canaux 2 à 83 en Europe et aux canaux 2 à 80 aux Etats-Unis. Du coup, la plupart des matériels se limitent aux canaux 2 à 80.

Pour toutes les autres variantes du WiFi sur la bande de 2.4 GHz, c'est-à-dire le 802.11 DSSS, le 802.11b et le 802.11g, quatorze canaux de 22 MHz de largeur sont définis, également numérotés à partir de 2400 MHz. Leurs centres ne sont espacés que de 5 MHz de sorte qu'ils se superposent en partie. Ceci permet de choisir avec une certaine souplesse la bande de fréquence que l'on préfère utiliser, mais si l'on a deux réseaux au même endroit et qu'ils utilisent des canaux voisins, on aura beaucoup d'interférences. Pour éviter les interférences, on recommande un espace de cinq canaux au moins, donc on ne peut utiliser que trois canaux simultanément au même endroit. Les canaux 1 à 13 sont utilisables en Europe, mais en Amérique on ne peut utiliser que les canaux 1 à 11. Quant au canal 14, seul le Japon y a droit. En conséquence, on utilise habituellement les canaux 1, 6 et 11 qui sont suffisamment espacés pour éviter toute interférence et sont autorisés presque partout sur la planète. Au mieux, avec le 802.11g, on peut donc avoir trois communications simultanées à **54 Mbps** chacune soit un total de 162 Mbps.

| Pays                                    | Etats – Unis | Europe | Japon | France  |
|-----------------------------------------|--------------|--------|-------|---------|
| <b>Nombre de sous - canaux utilisés</b> | 1 à 11       | 1 à 13 | 14    | 10 à 13 |

Tableau 1.01 : Affectation des canaux pour le 802.11 DSSS, le 802.11b et le 802.11g

En ce qui concerne le 802.11a qui utilise la bande de fréquence de 5 GHz, les centres de canaux successifs sont également espacés de 5 MHz, mais la numérotation commence à 5000 MHz. Par exemple, le canal 34 a pour centre 5170 MHz car  $34 \times 5 + 5000 = 5170$ . De plus, chaque canal a 20 MHz de largeur, donc le canal 34 s'étend de 5160 à 5180 MHz. Naturellement, si l'on souhaite éviter tout chevauchement, il faut utiliser au moins un écart de quatre canaux. Le tableau suivant montre les canaux utilisables actuellement avec le 802.11a en Europe.

### 1.3. Les modèles de couche [6] [9] [13] [14] [15] [16] [27] [28]

Comme tous les standards 802, la norme 802.11 s'attache à définir les couches basses du modèle OSI pour une liaison sans fil utilisant des ondes électromagnétiques, c'est-à-dire :

- La couche physique (notée parfois PHY), proposant trois types de codage de l'information,
- La couche liaison de données, constituée de deux sous-couches : le contrôle de la liaison logique (*Logical Link Control*, ou LLC) et le contrôle d'accès au support (*Media Access Control*, MAC).

La couche physique définit la modulation des ondes radioélectriques et les caractéristiques de la signalisation pour la transmission de données, tandis que la couche liaison de données définit l'interface entre le bus de la machine et la couche physique, notamment une méthode d'accès proche de celle utilisée dans le standard Ethernet et les règles de communication entre les différentes stations.

En situant le réseau sans fil dans le modèle OSI, nous voyons qu'il ne concerne que la couche 2. Le réseau sans fil est donc indépendant du protocole utilisé sur le réseau.

|                                                   |                                    |      |    |               |                |     |
|---------------------------------------------------|------------------------------------|------|----|---------------|----------------|-----|
| <b>OSI Layer 2</b><br><i>Data Link Layer</i>      | 802.11 Logical Link Control (LLC)  |      |    |               |                |     |
|                                                   | 802.11 Medium Access Control (MAC) |      |    |               |                |     |
| <b>OSI Layer 1</b><br><i>Physical Layer (PHY)</i> | FHSS                               | DSSS | IR | Wi-Fi 802.11b | Wi-Fi5 802.11a | ... |

Figure 1.07 : Couches 1 et 2 du modèle OSI pour le WiFi

### 1.3.1. La couche physique (PHY)

Le WiFi utilise trois types de modulation :

- Le Frequency Hopping Spread Spectrum (FHSS)
- Le Direct Sequence Spread Spectrum (DSSS)
- L'Orthogonal Frequency Division Multiplexing (OFDM)

Le FHSS n'est utilisé que dans la première version du standard 802.11. Le 802.11a repose exclusivement sur l'OFDM, le 802.11b exclusivement sur le DSSS ou l'OFDM, en fonction du débit souhaité. Ces trois modulations sont incompatibles, ce qui fait que seuls le 802.11 DSSS, le 802.11b et le 802.11g sont compatibles entre eux.

#### 1.3.1.1. Le Frequency Hopping Spread Spectrum (FHSS)

La technique de *Frequency Hopping Spread Spectrum* consiste à découper la large bande de fréquence en un minimum de 75 canaux (*hops* ou sauts d'une largeur de 1MHz), puis de transmettre en utilisant une combinaison de canaux connue de toutes les stations de la cellule. Dans la norme 802.11, la bande de fréquence 2.4 – 2.4835 GHz permet de créer 79 canaux de 1 MHz. La transmission est ainsi réalisée en émettant successivement sur un canal puis sur un autre pendant une courte période de temps (d'environ 400 ms), ce qui permet à un instant donné de transmettre un signal plus facilement reconnaissable sur une fréquence donnée.

L'émetteur et le récepteur s'accordent sur un schéma de saut, et les données sont envoyées sur une séquence de sous-canaux. Chaque conversation sur le réseau 802.11 s'effectue suivant un schéma de saut différent, et ces schémas sont définis de manière à minimiser le risque que deux expéditeurs utilisent simultanément le même sous-canal.

La séquence de fréquences utilisée est publique. FHSS est utilisé dans le standard 802.11 de telle manière à réduire les interférences entre les transmissions des diverses stations d'une cellule.

### 1.3.1.2. Le Direct Sequence Spread Spectrum (DSSS)

La technique de *Direct Sequence Spread Spectrum* divise la bande des 2,4 GHz en 14 canaux de 22 MHz. Les canaux adjacents se recouvrent partiellement, seuls trois canaux sur les 14 étant presque entièrement isolés. Le DSSS augmente la fréquence du signal numérique en le combinant avec un autre signal d'une fréquence plus élevée. Les données sont transmises intégralement sur l'un de ces canaux de 22 MHz, sans saut. La technique du « *chipping* » aide à compenser le bruit généré par un canal donné, c'est-à-dire moduler chaque bit avec la *séquence barker*.

Dans ce but, le standard 802.11 DSSS original spécifie un *chipping* sur 11 bits (baptisé *séquence barker*) pour le codage des données. La longueur du « *chipping code* » détermine combien de données seront transmises au-dessus d'une unité de temps (c'est-à-dire la bande passante). Ainsi chaque bit valant 1 est remplacé par la séquence de bits **10110111000** et chaque bit valant 0 par son complément **01001000111**.

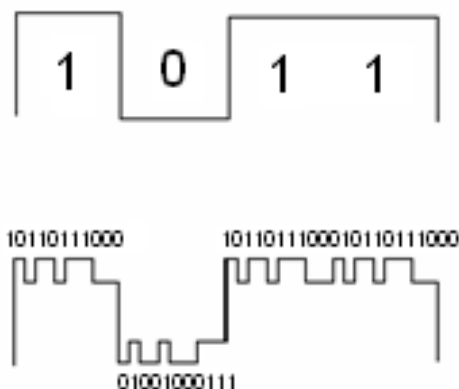


Figure 1.08 : Codage des données avec le 802.11 DSSS

La redondance inhérente à chaque chip associée à l'étalement du signal sur le canal de 22 MHz assure le contrôle et la correction d'erreur. Même si une partie du signal est endommagée, il peut dans la plupart des cas être récupéré, ce qui minimise les demandes de retransmission.

Chaque séquence de 11 chips représente un seul bit de données (**1** ou **0**) et est converti en forme d'onde (ou symbole) émissible. Ces symboles sont transmis à la vitesse de 1 MSps (1 million de symboles par seconde) par la technique BPSK (*Binary Phase Shift Keying*).

Pour supporter les environnements plus bruyants et étendre la portée des équipements, les WLAN 802.11b utilisent la variation dynamique du débit (*dynamic rate shifting*), qui permet d'ajuster les taux de transmission automatiquement pour compenser les variations du canal radio. Dans une situation idéale, les utilisateurs se connectent à un taux de **11 Mbps** plein. Cependant, lorsque les équipements sont déplacés au-delà de leur portée optimale pour un débit de 11 Mbps, ou en cas d'interférences conséquentes, les équipements 802.11b transmettent à des vitesses inférieures, redescendant en 5.5, 2 et 1 Mbps. De la même façon, si le périphérique revient dans un rayon compatible avec des transmissions plus rapides, la vitesse de la connexion s'accélère automatiquement. La variation dynamique du débit est un mécanisme de couche physique transparent à la fois pour l'utilisateur et pour les couches supérieures de la pile de protocoles.

#### 1.3.1.3. L'Orthogonal Frequency Division Multiplexing (OFDM)

OFDM est une méthode de codage appliquée aux normes 802.11a et g qui permet d'obtenir une meilleure bande passante. De ce fait, OFDM divise la bande de fréquence en bandes secondaires qui transmettent simultanément des fractions de données. Plus le nombre de canaux est élevé, plus les données transmises en parallèle sont nombreuses, plus la bande passante est élevée.

OFDM est un cas particulier de modulation multi porteuse ; il consiste à répartir les symboles sur un grand nombre de porteuses à bas débit. Pour un train de symboles initial de période  $T_{si}$ , les symboles seront répartis en  $N$  trains plus lents et auront alors une durée  $T_s = N \cdot T_{si}$ . Cette diminution du rythme symbole entraîne une diminution des interférences entre symboles d'un rapport  $N$ .

##### 1.3.1.3.1. Principe de la modulation

Pour répartir les données à transmettre sur les  $N$  porteuses, on groupe les symboles  $c_k$  par paquets de  $N$ . Les  $c_k$  sont des nombres complexes définis à partir des éléments binaires par une constellation souvent de modulation MAQ à 4, 16, 64,  $2^q$  états.

La séquence de  $N$  symboles  $c_0, c_1, \dots, c_{N-1}$  constitue un symbole OFDM. Le  $k$ -ième train de symboles parmi les  $N$  trains module un signal de fréquence  $f_k$ . Le signal modulé du train s'écrit sous forme complexe :

$$c_k e^{2j\pi f_k t} \quad (1.01)$$

Le signal total  $s(t)$  correspondant à l'ensemble des  $N$  symboles réassemblés en un symbole OFDM :

$$s(t) = \sum_{k=0}^{N-1} c_k e^{2j\pi f_k t} \quad (1.02)$$

Les fréquences sont orthogonales si l'espace entre deux fréquences adjacentes  $f_k$  et  $f_{k+1}$  est  $1/T_s$ . En effet, chaque porteuse modulant un symbole pendant une fenêtre rectangulaire temporelle de durée  $T_s$ , son spectre est un sinus cardinal, fonction qui s'annule tous les multiples  $1/T_s$ .

$$f_k = f_0 + \frac{k}{T_s} \quad (1.03)$$

Ainsi, lorsque l'échantillonnage est effectué précisément à la fréquence  $f_k$  d'une sous-porteuse il n'y a aucune interférence avec les autres sous-porteuses. C'est ce qui permet de recouvrir les spectres des différentes porteuses et d'obtenir ainsi une occupation optimale du spectre.

Le nombre de sous-porteuses  $N$  est choisi de manière à remplir les deux conditions primordiales :

$$T_s \gg T_m \quad (1.04)$$

afin de pouvoir considérer le canal plat, et

$$T_s \ll \frac{1}{B_d} \quad (1.05)$$

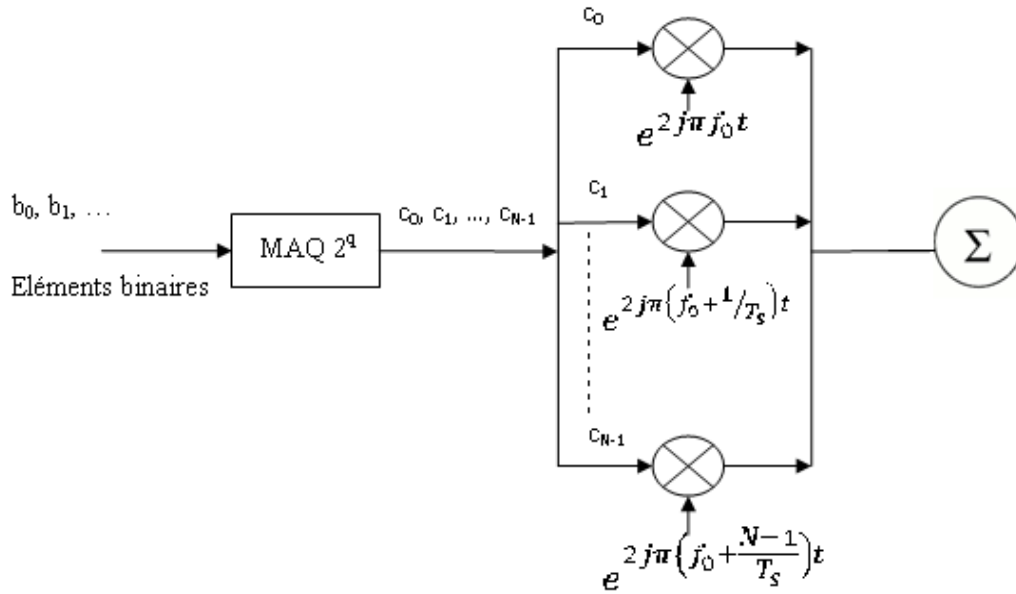


Figure 1.09 : Schéma de principe du modulateur OFDM

### 1.3.1.3.2. Principe de la démodulation

Le signal parvenu au récepteur s'écrit sur une durée symbole  $T_s$  :

$$y(t) = \sum_{k=0}^{N-1} c_k H_k(t) e^{2j\pi(f_0 + \frac{k}{T_s})t} \quad (1.06)$$

$H_k(t)$  est la fonction de transfert du canal autour de la fréquence  $f_k$  et à l'instant  $t$ . Cette fonction varie lentement et on peut la supposer constante sur la période  $T_s$  (définie dans l'équation (1.05)). La démodulation classique consisterait à démoduler le signal suivant les  $N$  sous-porteuses suivant le schéma classique de la figure 1.10.

Les systèmes WiFi à 5 GHz supportent différents débits en utilisant différents types de modulation. Actuellement, l'utilisation des modulations BPSK, QPSK, 16 QAM et 64 QAM supporte des débits de **6 Mbps** jusqu'à plus de **54 Mbps**.

### 1.3.2. Les évolutions du 802.11

Les principales améliorations concernant les couches physiques sont :

#### 1.3.2.1. Le 802.11a

Fréquence radio : 5 GHz

Modulation radio : OFDM

Débit maximum théorique : 54 Mbps

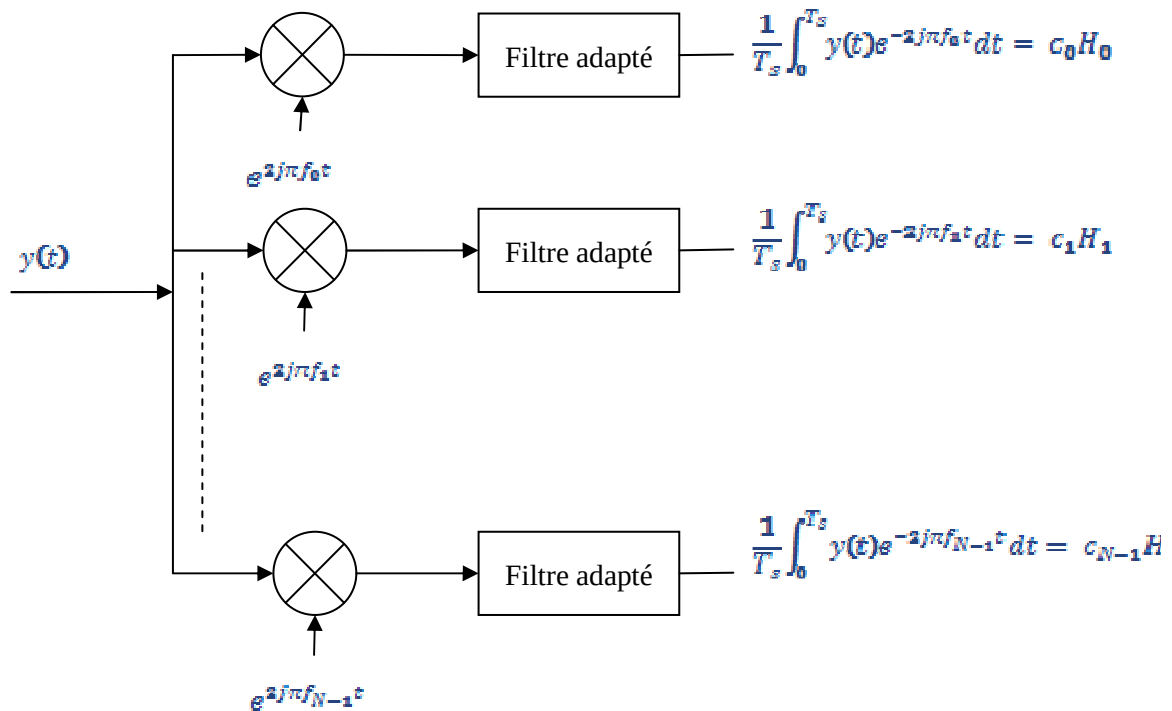


Figure 1.10 : Schéma de principe du démodulateur OFDM

#### 1.3.2.2. Le 802.11b

Fréquence radio : 2.4 GHz

Modulation : DSSS ou HR-DSSS

Débit maximum théorique : 11 Mbps

#### 1.3.2.3. Le 802.11g

Fréquence radio : 2.4 GHz

Modulation : DSSS, HR-DSSS ou OFDM

Débit maximum théorique : 54 Mbps

#### 1.3.2.4. Le 802.11n

Compatible avec le 802.11a et le 802.11b/g, il permettra grâce à de nombreuses améliorations techniques d'atteindre des débits très élevés ( $> 100$  Mbps).

Le débit théorique atteint les 540 Mbps grâce aux technologies MIMO et OFDM.

#### 1.3.3. La norme 802.11n

Lorsque les bornes WiFi 802.11a et g ont utilisé deux antennes, le but était de trouver et d'utiliser l'antenne qui avait la meilleure puissance pour dialoguer avec un autre équipement. Le point d'accès utilise alors une seule des deux antennes par communication.

Une nouvelle génération de WiFi a été conçue en tenant compte de deux contraintes : garder le même spectre de fréquences et ne pas augmenter la puissance. Le MIMO (*Multiple Input Multiple Output*) est alors apparu. Ce dernier, combiné avec la technologie WiFi 802.11 a et g, a donné naissance au 802.11n. A la différence des bornes ancienne génération, le MIMO utilise moins deux antennes pour envoyer des informations différentes sur chacune d'elles.

##### 1.3.3.1. La diversité spatiale

Les WLAN, comme le WiFi, sont bien souvent utilisés dans des milieux couverts avec tous types d'obstacles engendrant de nombreuses réflexions. Ces différentes réflexions entraînent la création de plusieurs canaux de transmission, comme le montre la figure 1.11.

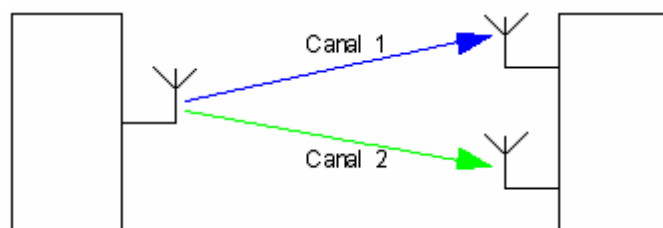


Figure 1.11 : La diversité spatiale

Ces différents canaux forment ce que l'on appelle la **diversité spatiale**.

##### 1.3.3.2. Principe de la technique MIMO

L'une des limitations du MIMO est donc que cette technique ne peut être utilisée qu'en intérieur, là où les réflexions sont possibles. Le secret de MIMO réside dans un algorithme capable

d'exploiter les différences de temps de parcours des signaux émis par un même émetteur en fonction de leurs rebonds.

En théorie, MIMO pourrait fonctionner avec un seul récepteur, mais le taux d'erreur serait alors trop important. Si l'on installe plus de récepteurs que d'émetteurs, le système y gagne en efficacité, mais y perd en simplicité et en coût. Concrètement, les émetteurs n'ont pas besoin d'être très distants les uns des autres, et les récepteurs non plus.

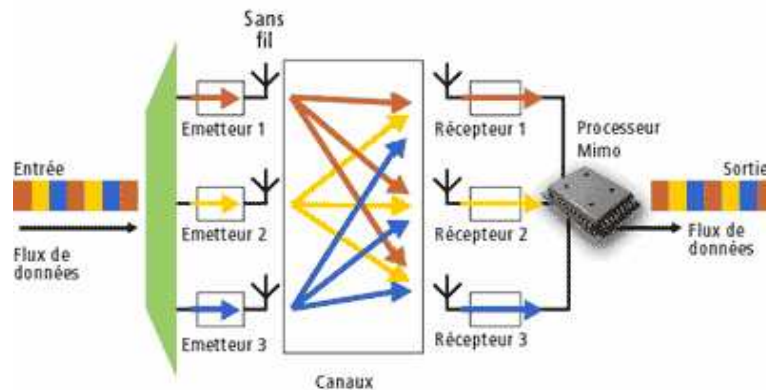


Figure 1.12 : Principe de la technique MIMO

### 1.3.3.3. Avantages du MIMO dans les réseaux haut débit

#### 1.3.3.3.1. Spatial Diversity Multiplexing (SDM)

Dans la figure 1.15, on constate qu'avec deux antennes de chaque côté, quatre canaux distincts sont créés. Lorsqu'un paquet S1 est envoyé à partir de l'antenne 1E à destination de 1R et que le paquet S2 est envoyé de 2E pour 2R, à la réception, chaque antenne reçoit à la fois ce qui a été envoyé par 1E et 2E multiplié par un coefficient complexe en fonction de leur canal.

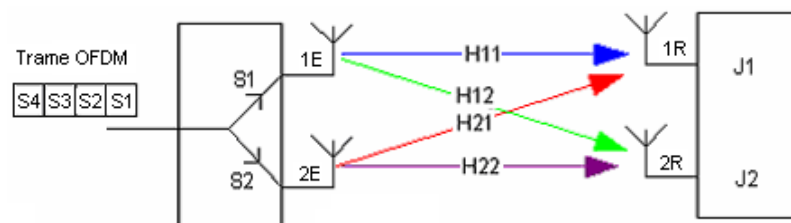


Figure 1.13 : SDM

Pour recomposer la trame, il va falloir résoudre un système de deux équations à deux inconnues afin d'isoler S1 et S2.

$$J1 = H11 S1 + H21 S2 \quad (1.08)$$

$$J2 = H12 S1 + H22 S2 \quad (1.09)$$

La façon dont on retrouve le signal émis est relativement simple grâce à l'OFDM. Le problème est de déterminer les coefficients complexes de chaque canal pour décoder la trame. Ce problème est vite écarté grâce à la structure de la trame OFDM WiFi. Une trame OFDM 802.11 est toujours constituée de deux paquets d'en-tête connus de l'émetteur et du récepteur ; de cette façon, les différents coefficients seront déterminés et le signal sera ensuite décodé puis la trame reconstituée.

| Préambule      | En-tête        |          |                  |        |                 |                   | MPDU | Tail | Pad |
|----------------|----------------|----------|------------------|--------|-----------------|-------------------|------|------|-----|
| 12<br>symboles | Rate<br>4 bits | Reserved | Length<br>4 bits | Parity | Trail<br>6 bits | Service<br>4 bits |      |      |     |

Figure 1.14 : Structure d'une trame OFDM

Légende :

- **Rate** : débit utilisé sur l'interface OFDM
- **Reserved** : réservé pour un usage futur ; ne contient que des 0
- **Length** : nombre d'octets dans la trame ; détermine la fin de trame
- **Parity** : calcul de parité ; détection d'erreur
- **Tail** : « queue », réservé pour un usage futur ; ne contient que des 0
- **Service** : réservé pour un usage futur ; ne contient que des 0.

Grâce au SDM, le MIMO permet au 802.11 d'atteindre des débits théoriques avoisinant les **150 Mbps** avec 2 antennes de chaque côté et une bande de 20 MHz de spectre ; avec 40 MHz, on atteint **270 Mbps**. Par ailleurs, la largeur de canal est flexible : elle peut être de 10, 20 ou 40 MHz. Malgré tout, ce type de fonctionnement ne rend pas le système WiFi plus robuste.

#### 1.3.3.3.2. Space Time Code Block (STCB)

Tout comme le MIMO-SDM, le MIMO-STCB envoie des signaux différents sur chaque antenne mais le principe du STBC est d'introduire une redondance d'information entre les deux antennes et ainsi, d'améliorer la robustesse pour une même puissance d'émission mais sans gagner du débit. Au niveau de l'émission, dans la trame OFDM qui a été générée précédemment, l'émetteur va prendre les paquets de cette trame deux par deux. En réalité, le système d'émission va envoyer

d'un côté S1 et S2 sans les modifier, et de l'autre côté il sera envoyé d'abord le conjugué de S2 et après, le conjugué de S1 négatif.

$$Y(n) = H_{11} S_1 + H_{21} S_2^* \quad (1.10)$$

$$Y(n+1) = H_{11} S_2 - H_{21} S_1^* \quad (1.11)$$

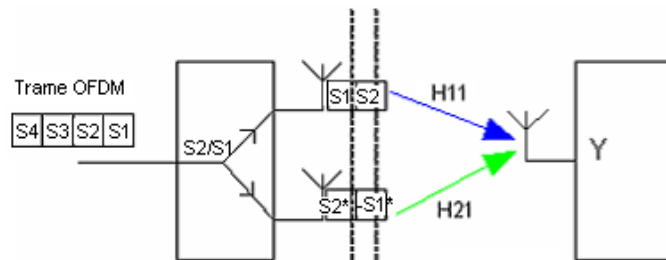


Figure 1.15 : STCB

Le système de réception va, dans un premier temps, déterminer les coefficients complexes des canaux, comme dans les SDM. Ensuite, ce système va recevoir un premier couple S1 et le conjugué de S2, puis dans un second temps, S2 et  $-S_1^*$ . Avec ces deux couples, la trame émise va pouvoir être reconstituée de façon très sûre grâce à un système de vérification entre les paquets S1 et S2 envoyés et leurs conjugués.

Grâce au STBC, le MIMO rend le WiFi plus robuste tout en gardant des débits équivalents à l'ancienne génération.

En comparant le MIMO avec l'ancienne génération de WiFi au niveau de la communication, en laissant de côté la sécurité, on peut constater deux améliorations majeures :

- A même puissance et à distance égale, le MIMO permet de doubler le débit du 802.11a ou g mais le réseau n'en est pas robuste ;
- A même puissance et à distance égale, le MIMO permet d'augmenter la portée des équipements tout en garantissant une qualité de réception.

#### 1.3.4. La couche Liaison de données

La couche Liaison de Données de la norme 802.11 est composée de deux sous-couches :

- La couche de contrôle de la liaison logique LLC,
- La couche de contrôle d'accès au support MAC.

Le standard 802.11 utilise la LLC 802.2 et l'adressage sur 48 bits, simplifiant le pontage entre les réseaux sans fil et les réseaux câblés. Le 802.11 MAC est très proche de 802.3 dans sa conception ; en effet, il est conçu pour supporter de multiples utilisateurs sur un support partagé en faisant détecter le support par l'expéditeur avant d'y accéder.

#### 1.3.4.1. Le protocole CSMA/CA

Le protocole CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) tente d'éviter au maximum les collisions (deux trames émises quasi simultanément qui se percutent) en imposant un accusé de réception systématique des paquets (**ACK**). Ainsi un paquet ACK est émis par la station de réception pour chaque paquet de données arrivé correctement.

##### 1.3.4.1.1. Fonctionnement

La station voulant émettre écoute le réseau. Si le réseau est encombré, la transmission est retardée. Si le réseau est libre pendant un temps donné (appelé **DIFS** pour *Distributed Inter Frame Space*), alors la station commence l'émission. La station transmet un message appelé **RTS** (*Ready To Send*) contenant des informations sur le volume des données qu'elle souhaite émettre et sa vitesse de transmission. Le récepteur (un AP dans le mode Infrastructure) répond un **CTS** (*Clear To Send*) et la station commence l'émission des données.

A la réception de toutes les données émises par la station, le récepteur envoie un accusé de réception (ACK). Toutes les stations avoisinantes patientent alors pendant un temps qu'elle estime être nécessaire à la transmission du volume d'information à émettre à la vitesse annoncée. Une fois que la trame ACK est reçue par l'émetteur, la station réceptrice met un terme au processus. Si la trame ACK n'est pas détectée par la station émettrice (parce que le paquet original ou le paquet ACK n'a pas été reçu intact), une collision est supposée et le paquet de données est retransmis après attente d'un autre temps aléatoire.

CSMA/CA permet donc de partager l'accès au médium dans un réseau sans fil. Le défaut principal de ce mécanisme d'accusé de réception est l'ajout à 802.11 d'une charge inconnue en 802.3. Nous pouvons alors affirmer qu'un réseau local 802.11 aura toujours des performances inférieures à un LAN Ethernet de débit théorique équivalent.

#### 1.3.4.1.2. RTS/CTS

Un autre problème de la couche MAC, spécifique au sans fil, réside dans le phénomène du “nœud caché” où deux stations situées de chaque côté d’un point d’accès peuvent entendre toutes les deux une activité du point d’accès, mais pas de l’autre station.

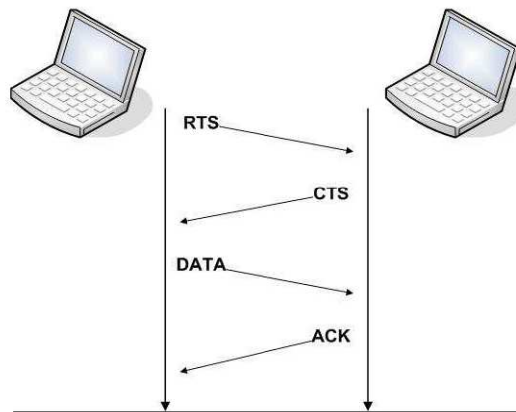


Figure 1.16 : Fonctionnement du protocole CSMA/CA

Ce problème est généralement lié aux distances ou à la présence d’un obstacle. Pour résoudre ce problème, le standard 802.11 définit sur la couche MAC un protocole optionnel de type RTS/CTS. Lorsque cette fonction est utilisée, une station émettrice transmet un RTS et attend que le point d’accès réponde par un CTS. Toutes les stations du réseau peuvent entendre le point d’accès et le CTS leur permet de retarder toute transmission prévue, la station émettrice pouvant alors transmettre et recevoir un accusé de réception sans aucun risque de collision. L’ajout par le protocole RTS/CTS de la réservation temporaire du support à la charge du réseau alourdit la transmission et risque de diminuer la bande passante.

#### 1.3.4.2. Les méthodes d’accès au support

Le *Point Coordination Function (PCF)* est une fonction qui garantit une transmission à un rythme régulier, permettant de synchroniser les flux (images, sons ou autres) ou de travailler en temps réel. C’est une alternative à CSMA/CA qui utilise la fonction *Distributed Coordination Function (DCF)*.

Les stations effectuent régulièrement trois actions, à savoir le sondage périodique par l’AP (*polling*), l’émission d’une station si elle est autorisée et la réception d’une station si elle est sélectionnée.

#### 1.3.4.3. Propriétés supplémentaires des couches MAC et LLC

La couche MAC de 802.11 offre deux autres caractéristiques de robustesse à savoir les sommes de contrôle CRC et la fragmentation des paquets. Une somme de contrôle est calculée pour chaque paquet et rattachée à celui-ci afin d'assurer que les données n'ont pas été corrompues durant leur transfert. Cette technique diffère d'Ethernet où les protocoles de niveau supérieur tels que TCP gèrent le contrôle d'erreur.

La fragmentation des paquets permet de casser les gros paquets en unités de plus petite taille, ce qui s'avère particulièrement utile dans les environnements très congestionnés ou lorsque les interférences posent problème, puisque les gros paquets courent plus de risque d'être corrompus. Cette technique limite le risque de devoir retransmettre un paquet et améliore ainsi globalement les performances du réseau sans fil. La couche MAC est responsable de la reconstitution des fragments reçus.

La couche LLC de 802.11 gère aussi la gestion d'énergie. Deux modes de gestion d'alimentation sont prévus :

- Le CAM (*Continuous Aware Mode*) : l'appareil, toujours allumé, consomme de l'énergie en permanence.
- Le PSPM (*Power Save Polling Mode*) : l'appareil est mis en veille. L'AP met alors en file d'attente les données qui lui sont destinées.

#### 1.3.4.4. Les évolutions de la couche MAC

- **802.11c** : Il apporte quelques précisions sur le fonctionnement d'un Access Point connecté à un réseau filaire.
- **802.11d** : Il établit la liste des règles à suivre selon le pays pour pouvoir émettre sur telle ou telle fréquence : éviter tel ou tel canal, limiter la puissance, ...
- **802.11e** : ratifié fin 2005, il définit des mécanismes permettant de contrôler le flux de données et le partage de média entre plusieurs stations : ceci permet notamment de mettre en œuvre une véritable gestion de la QoS pour permettre l'échange fluide de données multimédias.
- **802.11f** : Il définit le protocole inter points d'accès (*Inter Access Point Protocol*, IAPP) : celui-ci précise comment des points d'accès d'un même réseau sans fil doivent communiquer entre eux. L'IAPP permet de constituer un réseau sans fil à partir d'AP de

différents constructeurs. Les AP ont souvent besoin de communiquer entre eux, notamment pour permettre le hand-over.

- **802.11h** : Il apporte les modifications à la couche physique 802.11a ainsi qu'à la couche MAC afin de mieux adapter le WiFi au marché européen. En effet, le mode de fonctionnement du WiFi entre en conflit avec certaines communications satellites pour la bande de fréquence de contrôle de la puissance de transmission (*Transmit Power Control*, TPC) dynamique, qui consiste à ne jamais émettre plus fort que nécessaire, tout en restant assez fort pour que le récepteur puisse capter le signal. En outre, si un canal est occupé par une communication radio non WiFi, le 802.11h s'appelle la sélection dynamique de fréquence (*Dynamic Frequency Selection*, DFS). Le 802.11h a été validé en septembre 2003.
- **802.11i** : La première solution de sécurité offerte par la première version de la couche MAC s'appelle le Wired Equivalent Privacy (WEP). Le 802.11i a été ratifié le 24 juin 2004 et il apporte une solution nettement plus sûre et flexible que le WEP.
- **802.11j** : Il est au Japon ce que le 802.11h est à l'Europe : il définit une série de mécanismes pour adapter le 802.11a à la législation du Japon.
- **802.11k** : Il définit un certain nombre de paramètres radio et de statistiques qui peuvent être échangés entre équipements WiFi ou présentés à un utilisateur.
- **802.11s** : Groupe de travail pour les réseaux maillés.

#### **1.4. Le format des trames [3]**

##### **1.4.1. Les trames de la couche PHY**

Lorsqu'un paquet de données doit être envoyé sur les ondes, l'adaptateur WiFi commence par le traiter au niveau de la couche MAC. En bref, le paquet est éventuellement fragmenté et les fragments sont encapsulés dans des paquets appelés des *MAC Protocol Data Unit* (MPDU). La couche physique a donc pour responsabilité de transmettre sur les ondes les MPDU fournis par la couche MAC et inversement, de fournir à la couche MAC les paquets reçus sur les ondes.

Au niveau de la couche physique, le MPDU est inclus dans une trame 802.11 dont la structure est celle représentée par la figure 1.17.

|                  |                       |             |
|------------------|-----------------------|-------------|
| <b>Préambule</b> | <b>En – tête PLCP</b> | <b>MPDU</b> |
|------------------|-----------------------|-------------|

Figure 1.17 : Structure d'une trame 802.11

#### 1.4.1.1. Le préambule

Le préambule permet au récepteur de se synchroniser correctement avec l'émetteur, de s'adapter aux légers décalages de fréquence qui peuvent survenir et éventuellement de choisir l'antenne à la réception, si le récepteur en a plusieurs.

#### 1.4.1.2. L'en – tête PLCP

L'en-tête *Physical Layer Convergence Procedure* (PLCP) contient des informations importantes pour que le récepteur puisse se préparer à la réception du MPDU. En particulier, la longueur de la trame et la vitesse de transfert à utiliser sont indiquées. L'en-tête PLCP est toujours transmis à 1 Mbps, quelle que soit la couche physique utilisée et le débit peut augmenter pour la transmission du MPDU.

### 1.4.2. Les trames MAC

Le standard 802.11 définit le format des trames échangées. Chaque trame est constituée d'un en-tête (appelé *MAC header*, d'une longueur de 30 octets), d'un corps et d'un *FCS* (*Frame Sequence Check*) permettant la correction d'erreur.

| <b>FC</b><br>(2)                              | <b>D/ID</b><br>(2) | <b>Adresse 1</b><br>(4 octets) | <b>Adresse 2</b><br>(4 octets) | <b>Adresse 3</b><br>(4 octets) | <b>SC</b><br>(2) | <b>Adresse 4</b><br>(4 octets) |
|-----------------------------------------------|--------------------|--------------------------------|--------------------------------|--------------------------------|------------------|--------------------------------|
| <b>Corps de la trame</b><br>(0 à 2312 octets) |                    |                                |                                |                                |                  |                                |
| <b>FCS</b><br>(2)                             |                    |                                |                                |                                |                  |                                |

Figure 1.18 : Structure d'une trame MAC

Description des champs :

- **FC** (*Frame Control*, en français *contrôle de trame*) : ce champ de deux octets est constitué des informations suivantes :
  - o **Versión du protocole** : ce champ de 2 bits permettra de prendre en compte les évolutions de version du standard 802.11. La valeur est égale à zéro pour la première version

- o **Type ou Sous-type** : ces champs, respectivement de 2 et 4 bits, définissent le type et le sous-type des trames explicités dans le tableau ci-dessous. Le type *gestion* correspond aux demandes d'association ainsi qu'aux messages d'annonce du point d'accès. Le type *contrôle* est utilisé pour l'accès au média afin de demander des autorisations pour émettre. Enfin le type *données* concerne les envois de données (la plus grande partie du trafic).
- o **To DS** : ce bit vaut 1 lorsque la trame est destinée au système de distribution (DS), il vaut zéro dans les autres cas. Toute trame envoyée par une station à destination d'un point d'accès possède ainsi un champ *To DS* positionné à 1.
- o **From DS** : ce bit vaut 1 lorsque la trame provient du système de distribution (DS), il vaut zéro dans les autres cas. Ainsi, lorsque les deux champs *To* et *From* sont positionnés à zéro il s'agit d'une communication directe entre deux stations (mode *ad hoc*).
- o **More fragments** (*fragments supplémentaires*) : permet d'indiquer (lorsqu'il vaut 1) qu'il reste des fragments à transmettre
- o **Retry** : ce bit spécifie que le fragment en cours est une retransmission d'un fragment précédemment envoyé (et sûrement perdu)
- o **Power Management** (*gestion d'énergie*) : indique, lorsqu'il est à 1, que la station ayant envoyé ce fragment entre en mode de gestion d'énergie
- o **More Data** : ce bit, utilisé pour le mode de gestion d'énergie, est utilisé par le point d'accès pour spécifier à une station que des trames supplémentaires sont stockées en attente.
- o **WEP** : ce bit indique que l'algorithme de chiffrement WEP a été utilisé pour chiffrer le corps de la trame.
- o **Order** (*ordre*) : indique que la trame a été envoyée en utilisant la classe de service strictement ordonnée (*Strictly-Ordered service class*)

| Version du protocole<br>(2 bits) |                    | Type<br>(2 bits)     |                  | Sous-type<br>(4 bits) |                      |                |                  |
|----------------------------------|--------------------|----------------------|------------------|-----------------------|----------------------|----------------|------------------|
| To DS<br>(1 bit)                 | From DS<br>(1 bit) | More Frag<br>(1 bit) | Retry<br>(1 bit) | Power Mgt<br>(1 bit)  | More Data<br>(1 bit) | WEP<br>(1 bit) | Order<br>(1 bit) |

Figure 1.19 : Champ FC

- **Durée / ID** : Ce champ indique la durée d'utilisation du canal de transmission.
- **Champs d'adresses** : une trame peut contenir jusqu'à 3 adresses en plus de l'adresse de 48 bits

- **Contrôle de séquence** : ce champ permet de distinguer les divers fragments d'une même trame. Il est composé de deux sous-champs permettant de réordonner les fragments :
  - o Le *numéro de fragment*
  - o Le *numéro de séquence*
- **CRC** : une somme de contrôle servant à vérifier l'intégrité de la trame.

### 1.5. Conclusion

Les bandes allouées au WiFi sont 2.4 GHz et 5 GHz. Le standard IEEE 802.11 supporte à la fois les techniques FHSS et DSSS pour la bande de 2.4 GHz. Les techniques à étalement de spectre sont robustes contre les interférences et ne nécessitent pas d'égalisation adaptative. Pour des débits de transmission supérieurs à 10 Mbps, l'OFDM montre de meilleures performances.

## CHAPITRE 2 : LA SECURITE DES RESEAUX WIFI

### 2.1. Introduction [13]

La sécurité d'un réseau est un niveau de garantie que l'ensemble des machines du réseau fonctionnent de façon optimale et que les utilisateurs des dites machines possèdent uniquement les droits qui leur ont été octroyés.

Il peut s'agir d'empêcher des personnes non autorisées d'agir sur le système de façon malveillante, d'empêcher les utilisateurs d'effectuer des opérations involontaires capables de nuire au système, de sécuriser les données en prévoyant les pannes ou de garantir la non-interruption d'un service

#### 2.1.1. Les causes de l'insécurité

On distingue généralement deux types d'insécurité :

- **l'état actif d'insécurité**, c'est-à-dire la non-connaissance par l'utilisateur des fonctionnalités du système, dont certaines pouvant lui être nuisibles (par exemple la non-désactivation de services réseaux non nécessaires à l'utilisateur)
- **l'état passif d'insécurité**, c'est-à-dire lorsque l'administrateur (ou l'utilisateur) d'un système ne connaît pas les dispositifs de sécurité dont il dispose

#### 2.1.2. Qui sont les attaquants ?

Il existe trois grands types de pirates :

- Les **White Hats** qui sont des pirates bienveillants. Souvent consultants en sécurité, cyber policiers ou administrateurs réseaux, ils concentrent leur énergie à la recherche de failles dans le but de les signaler pour qu'elles soient comblées,
- Les **Black Hats** qui ont pour but premier la nuisance. Ce sont eux les cybercriminels : créateurs de virus, cyber espions, cyber terroristes et cyber escrocs.
- Les **Gray Hats** sont globalement intermédiaires entre les deux types de pirates cités précédemment : ils pénètrent dans les systèmes mais sans les détériorer. Ils ont plutôt pour

but de récupérer des informations sans se faire repérer ou de tester leurs connaissances sur de nouveaux systèmes.

### **2.1.3. Motivations et but des agresseurs**

Les motivations des agresseurs que l'on appelle communément "pirates" peuvent être multiples : l'attrance de l'interdit, le désir d'argent (violer un système bancaire par exemple), le besoin de renommée (impressionner des amis) ou l'envie de nuire (détruire des données, empêcher un système de fonctionner)

Les pirates pourront être des employés malveillants, des journalistes, des concurrents (espionnage industriel), la police ou des organisations nationales de renseignement, des terroristes, ou simplement des criminels isolés.

Le but des agresseurs est souvent de prendre le contrôle d'une machine afin de pouvoir réaliser les actions qu'ils désirent. Pour cela il existe différents types de moyens :

- l'obtention d'informations utiles pour effectuer des attaques
- utiliser les failles d'un système
- l'utilisation de la force pour casser un système

### **2.1.4. Comment se protéger ?**

- se tenir au courant
- connaître le système d'exploitation
- réduire l'accès au réseau (firewall)
- réduire le nombre de points d'entrée (ports)
- définir une politique de sécurité interne (mots de passe)

## **2.2. Les principales attaques contre le réseau sans fil [2] [3] [13]**

### **2.2.1. Le war-driving**

Etant donné qu'il est très facile d'"écouter" des réseaux sans fils, une pratique venue tout droit des Etats-Unis consiste à circuler dans la ville avec un ordinateur portable (voire un assistant personnel) équipé d'une carte réseau sans fil à la recherche de réseaux sans fils, il s'agit du **war-**

**driving** (parfois noté *wardriving* ou *war-Xing* pour "war crossing"). Des logiciels spécialisés dans ce type d'activité permettent même d'établir une cartographie très précise en exploitant un matériel de géolocalisation (GPS, *Global Positionning System*).

Les cartes établies permettent ainsi de mettre en évidence les réseaux sans fil déployés non sécurisés, offrant même parfois un accès à internet ! De nombreux sites capitalisant ces informations ont vu le jour sur internet, si bien que des étudiants londoniens ont eu l'idée d'inventer un "langage des signes" dont le but est de rendre visible les réseaux sans fils en dessinant à même le trottoir des symboles à la craie indiquant la présence d'un réseau sans fil, il s'agit du « **war-chalking** » (francisé en *craieFiti* ou *craie-fiti*). Deux demi-cercles opposés désignent ainsi un réseau ouvert offrant un accès à Internet, un rond signale la présence d'un réseau sans fil ouvert sans accès à un réseau filaire et enfin un W encerclé met en évidence la présence d'un réseau sans fil correctement sécurisé (cf. figure 2.01).

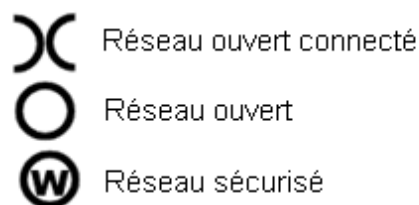


Figure 2.01 : Le Warchalking

### 2.2.2. L'intrusion

Lorsqu'un point d'accès est installé sur le réseau local, il permet aux stations d'accéder au réseau filaire et éventuellement à internet si le réseau local y est relié. Un réseau sans fil non sécurisé représente de cette façon un point d'entrée royal pour le pirate au réseau interne d'une entreprise ou une organisation.

Outre le vol ou la destruction d'informations présentes sur le réseau et l'accès à internet gratuit pour le pirate, le réseau sans fil peut également représenter une aubaine pour ce dernier dans le but de mener des attaques sur Internet. En effet étant donné qu'il n'y a aucun moyen d'identifier le pirate sur le réseau, l'entreprise ayant installé le réseau sans fil risque d'être tenue responsable de l'attaque.

### **2.2.3. Le déni de service**

Le *Denial of Service* (DoS) vise à stopper un service réseau en cours pour perturber le bon fonctionnement d'un système. Elle n'a pas de finalité si elle est utilisée seule, mais elle peut aider d'autres. Il existe plusieurs types de DoS :

#### **2.2.3.1. Le flooding**

Il consiste à envoyer de gros paquets de données à une machine plus vite qu'elle ne peut en recevoir. Au bout d'un moment, la machine visée finit par se déconnecter.

#### **2.2.3.2. Le smurf**

Le smurf est une ancienne attaque qui s'appuie sur le ping et les serveurs de diffusion. On commence par falsifier son adresse IP pour se faire passer pour la machine cible. Ensuite, on envoie un message ping sur un serveur de diffusion. Il le fera suivre à toutes les machines qui sont connectées qui renverront chacune un « pong » au serveur qui fera suivre à la machine cible. Celle-ci sera alors inondée sous les paquets et finira par se déconnecter.

#### **2.2.3.3. Le débordement de tampon**

Il se base sur une faille du protocole IP : on envoie à la machine cible des blocs de données d'une taille supérieure à la capacité d'un paquet. Ces données seront alors fractionnées pour l'envoi et réassemblées par la machine cible. A ce moment, il y aura débordement des variables internes dû à la non-conformité de la taille des données reçues.

### **2.2.4. Le brouillage radio**

Il consiste à émettre des signaux radio de telle manière à produire des interférences.

Les ondes radio sont très sensibles aux interférences, c'est la raison pour laquelle un signal peut facilement être brouillé par une émission radio ayant une fréquence proche de celle utilisée dans le réseau sans fil. Un simple four à micro-ondes peut ainsi rendre totalement inopérable un réseau sans fil lorsqu'il fonctionne dans le rayon d'action d'un point d'accès.

### **2.2.5. L'interception de données**

Elle consiste à écouter les transmissions des différents utilisateurs du réseau sans fil.

Par défaut un réseau sans fil est non sécurisé, c'est-à-dire qu'il est ouvert à tous et que toute personne se trouvant dans le rayon de portée d'un point d'accès peut potentiellement écouter toutes les communications circulant sur le réseau. Pour un particulier la menace est faible car les données sont rarement confidentielles, si ce n'est les données à caractère personnel. Pour une entreprise en revanche l'enjeu stratégique peut être très important.

### 2.3. Les différentes solutions de sécurité [2] [3] [4] [13] [17]

Conformément à la normalisation de l'IEEE, la sécurité des réseaux 802.11 s'appuie, en première analyse, sur deux composants principaux : le cryptage et l'authentification. De nombreux articles montrent que l'ensemble des professionnels de la sécurité s'accordent sur un point : la mise en œuvre de ces composants n'offre pas de garanties suffisantes de sécurité.

#### 2.3.1. Le chiffrement des trames

Un bon chiffrement garantit la confidentialité des données. Le cryptage consiste à prendre un message, dit « en clair », et à le soumettre à un algorithme mathématique pour produire un texte « crypté ». Le déchiffrement est la transformation inverse. Les algorithmes de cryptage se servent le plus souvent d'une valeur, appelée clé, qui sert à chiffrer et à déchiffrer les données. Les deux principales méthodes de cryptage utilisées actuellement sont le **cryptage symétrique** (également appelé cryptage à clé partagée) et le **cryptage asymétrique** (également connu sous le nom de cryptage à clé publique ou à clé privée). Le cryptage symétrique est environ 1000 fois plus rapide que le cryptage asymétrique et est donc utilisé pour le cryptage en masse des données. De manière générale, lorsque l'algorithme de cryptage est bien conçu, la protection est d'autant plus grande que la clé est longue car le pirate devra utiliser davantage de « force brute » pour essayer toutes les clés possibles afin de décrypter le message.

##### 2.3.1.1. Le Wired Equivalent Privacy (WEP)

L'IEEE a spécifié que le protocole WEP (*Wired Equivalent Privacy*) serait utilisé pour crypter les trames de données 802.11.

Le *WEP* est un protocole chargé du chiffrement des trames 802.11 utilisant l'algorithme symétrique *RC4* avec des clés d'une longueur de 64 bits ou 128 bits. Le principe du WEP consiste à définir dans un premier temps une clé secrète de 40 ou 104 bits. Cette clé secrète doit être déclarée au niveau du point d'accès et des clients. La clé sert à créer un nombre pseudo-aléatoire

d'une longueur égale à la longueur de la trame. Chaque transmission de donnée est ainsi chiffrée en utilisant le nombre pseudo-aléatoire comme masque grâce à un *OU Exclusif* entre le nombre pseudo-aléatoire et la trame.

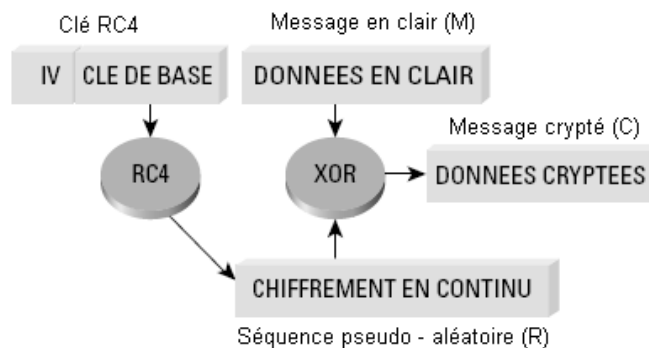


Figure 2.02 : Processus de cryptage WEP.

La clé de session partagée par toutes les stations est statique, c'est-à-dire que pour déployer un grand nombre de stations Wifi il est nécessaire de les configurer en utilisant la même clé de session. Ainsi la connaissance de la clé est suffisante pour déchiffrer les communications.

De plus, 24 bits de la clé servent uniquement pour l'initialisation, ce qui signifie que seuls 40 bits de la clé de 64 bits servent réellement à chiffrer et 104 bits pour la clé de 128 bits.

Dans le cas de la clé de 40 bits, une attaque par force brute (c'est-à-dire en essayant toutes les possibilités de clés) peut très vite amener le pirate à trouver la clé de session. De plus une faille décelée par Fluhrer, Mantin et Shamir concernant la génération de la chaîne pseudo-aléatoire rend possible la découverte de la clé de session en stockant 100 Mo à 1 Go de trafic créés intentionnellement.

Le WEP n'est donc pas suffisant pour garantir une réelle confidentialité des données. Pour autant, il est vivement conseillé de mettre au moins en œuvre une protection WEP 128 bits afin d'assurer un niveau de confidentialité minimum et d'éviter de cette façon 90% des risques d'intrusion.

#### 2.3.1.1.1. Le vecteur d'initialisation

Pour éviter la répétition de la clé RC4, le WEP utilise une solution très simple : pour chaque paquet, l'émetteur génère un numéro unique, qu'il ne réutilisera jamais (en principe). Un tel numéro est appelé « *nonce* ». Une façon simple de générer des nonces est d'utiliser une simple

séquence (1, 2, 3, ...), mais chaque constructeur est libre de générer le nonce à sa guise. Ce nonce est combiné à une clé fixe pour créer la clé RC4 utilisée pour le cryptage du paquet. Avec le WEP, ce nonce est composé de 24 bits et s'appelle **vecteur d'initialisation** (*Initialisation Vector*, IV). Il est simplement rajouté avant la clé WEP pour former la clé RC4. La clé RC4 a donc le format suivant :

| IV                 | Clé WEP                         |
|--------------------|---------------------------------|
| 3 octets (24 bits) | 5 ou 13 octets (40 ou 104 bits) |

Figure 2.03 : La clé RC4

Bien entendu, pour pouvoir décrypter le message, le récepteur doit connaître la clé RC4 au complet. Il connaît déjà la clé WEP puisqu'elle est configurée dans chaque poste et chaque AP du réseau, mais comment connaître l'IV ? La réponse est simple : l'IV est envoyé en clair au début de chaque paquet (après l'en-tête MAC).

Voici le format d'un paquet crypté avec le WEP :

| IV       | ID      | Données cryptées | ICV crypté |
|----------|---------|------------------|------------|
| 3 octets | 1 octet | 0 à 2304 octets  | 4 octets   |

Figure 2.04 : Format d'un paquet crypté avec le WEP

Le champ ID indique laquelle des quatre clés WEP (de 0 à 3) a été utilisée pour le cryptage. L'ICV fait partie du message crypté. Il suffit donc au récepteur de lire l'IV, de le rajouter avec la bonne clé WEP pour former la clé RC4, et à partir de là, de décrypter le message.

#### 2.3.1.1.2. Le contrôle d'intégrité

Le dernier volet de la sécurité WEP est le contrôle de l'intégrité des paquets échangés. Le but est de s'assurer qu'ils ne soient pas modifiés par un pirate pendant leur transport. Rappelons qu'un code de redondance cyclique (CRC) de 32 bits est rajouté à la fin de chaque paquet Wifi. Ce code est calculé en fonction du contenu du paquet et en constitue une sorte de résumé : si un pirate modifie ne serait-ce qu'un seul bit du paquet, le CRC ne sera plus valable. Malheureusement, le CRC a été conçu pour lutter contre les erreurs de transmission, mais il ne peut rien contre un pirate : en effet, si un pirate intercepte un paquet et le modifie, il lui suffit de recalculer le CRC avant de laisser ce paquet poursuivre sa route. Dans ce cas, le destinataire ne se rendra pas compte que le paquet a été modifié car le CRC sera correct.

Pour résoudre ce problème, le WEP a défini un mécanisme assez simple : un code de vérification de l'intégrité du message (*Integrity Check Value*, ICV) est calculé de façon similaire au CRC habituel, sur 32 bits également. Toutefois, l'ICV est calculé non pas à partir du paquet crypté comme le CRC habituel, mais à partir du message original. L'ICV est inséré à la fin du message, et le tout crypté par l'algorithme décrit précédemment.

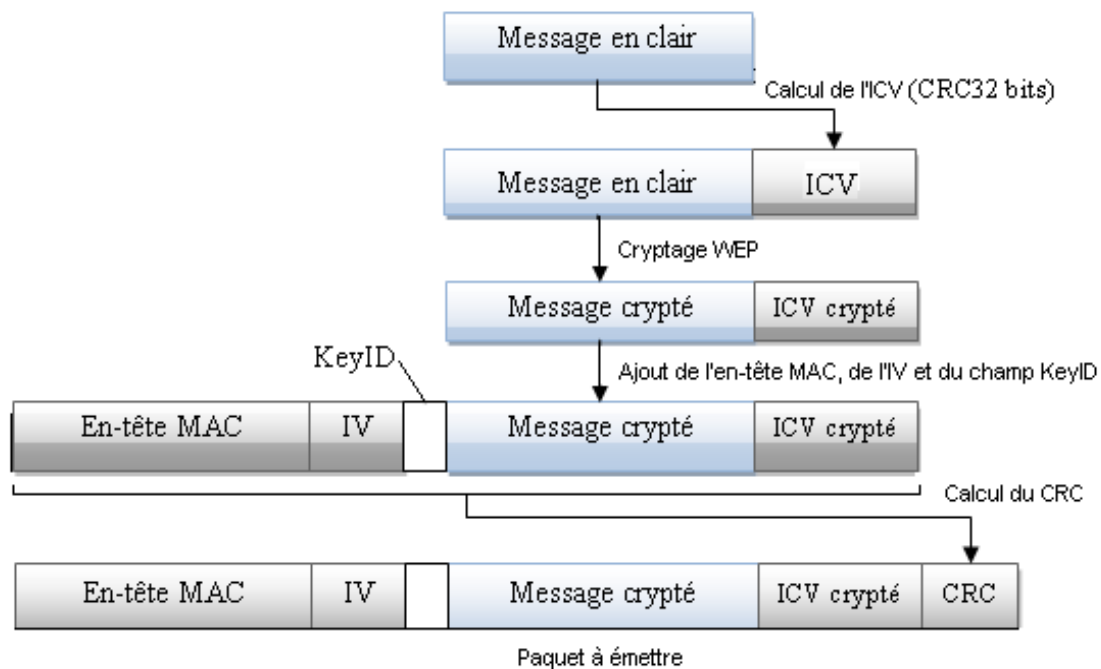


Figure 2.05 : Le contrôle d'intégrité avec l'ICV

C'est-à-dire, si le message original est  $M$  et que la séquence pseudo-aléatoire générée par l'algorithme RC4 est  $R$ , alors le message crypté  $C$  est défini par la formule suivante :

$$C = [M \parallel \text{CRC}(M)] \oplus R \quad (2.01)$$

$\parallel$  signifie « suivi de »

De cette façon, si un pirate veut modifier le paquet crypté sans connaître la clé WEP, il aura en principe du mal à le faire. En effet, la modification donne un nouveau message crypté  $C'$ , correspondant à un nouveau message en clair  $M'$ . Puisque le pirate ne possède pas la clé WEP, il ne peut pas savoir quel est ce message  $M'$  et ne peut donc pas calculer son ICV. Même s'il parvenait à le calculer, il ne pourrait pas le crypter pour l'intégrer à la fin du message  $C'$ , car il ne possède pas la clé WEP.

### 2.3.1.2. Le WPA (Wi-Fi Protected Access) et le WPA2

Le WPA est une solution de sécurisation de réseau Wi-Fi proposé par la *Wifi Alliance* afin de combler les lacunes du WEP.

Le WPA est une version « allégée » du protocole 802.11i, reposant sur des protocoles d'authentification et un algorithme de cryptage robuste : **TKIP** (*Temporary Key Integrity Protocol*). Le protocole TKIP permet la génération aléatoire de clés et offre la possibilité de modifier la clé de chiffrement plusieurs fois par secondes, pour plus de sécurité.

Le fonctionnement de WPA repose sur la mise en œuvre d'un serveur d'authentification (la plupart du temps un serveur RADIUS), permettant d'identifier les utilisateurs sur le réseau et de définir leurs droits d'accès. Néanmoins, il est possible pour les petits réseaux de mettre en œuvre une version restreinte du WPA, appelée *WPA-PSK*, en déployant une même clé de chiffrement dans l'ensemble des équipements, ce qui évite la mise en place d'un serveur RADIUS.

Le WPA (dans sa première mouture) ne supporte que les réseaux en mode infrastructure, ce qui signifie qu'il ne permet pas de sécuriser des réseaux sans fil d'égal à égal (mode ad-hoc).

Le WPA et le WPA2 sont identiques du point de vue de leur architecture globale et donc de leur mise en œuvre. Le WPA repose sur un algorithme de cryptage défini par le protocole *Temporal Key Integrity Protocol* (TKIP), lui-même basé sur l'algorithme RC4 ; alors que le WPA2 repose, au choix, sur le TKIP ou sur un autre algorithme de cryptage appelé *Advanced Encryption Standard* (AES). Le WPA2 offre donc le choix du cryptage, contrairement au WPA qui impose le TKIP.

Une autre différence importante est que le WPA n'est compatible qu'avec les réseaux de type Infrastructure et non les réseaux Ad-Hoc. Quant au WPA2, il peut sécuriser les deux types de réseau.

Il existe deux architectures pour le WPA ou le WPA2 :

- Avec des clés partagées : on parle de *WPA Personal*
- Avec une architecture 802.1x : on parle de *WPA Enterprise*

|                        | <b>WPA</b>                                                    | <b>WPA2</b>                                                     |
|------------------------|---------------------------------------------------------------|-----------------------------------------------------------------|
| <b>Enterprise Mode</b> | Authentification : IEEE 802.1x / EAP<br>Encryption : TKIP/MIC | Authentification : IEEE 802.1x / EAP<br>Encryption : AES – CCMP |
| <b>Personal Mode</b>   | Authentification : PSK<br>Encryption : TKIP/MIC               | Authentification : PSK<br>Encryption : AES – CCMP               |

Tableau 2.01 : Enterprise Mode et Personal Mode dans WPA

PSK : Pre-Shared Key ou clé partagée.

Il existe toutefois une contrainte imposée par le WPA et le WPA2 : si l'on veut bénéficier de la distribution automatique des clés de cryptage, la méthode EAP utilisée doit être capable de générer une clé secrète entre le client et le serveur d'authentification. Si c'est le cas, on dit que la méthode est génératrice de clés (*Key Generating*).

Pour mettre en place une architecture 802.1x, il est très fortement recommandé d'utiliser une méthode génératrice de clés. Par exemple :

- EAP/TLS
- TTLS
- PEAP
- EAP/FAST

Les méthodes TTLS, PEAP et EAP/FAST peuvent être utilisées avec n'importe quelle méthode d'authentification interne.

#### 2.3.1.2.1. La solution TKIP

La solution de sécurité *Temporal Key Integrity Protocol* (TKIP) a été introduite avec le WPA en 2002, en réponse aux défaillances de sécurité du WEP. Le cryptage TKIP ne possède aucune des failles majeures du WEP, pourtant il repose sur le même algorithme RC4.

Les principales modifications apportées par TKIP sont :

- Le contrôle d'intégrité repose sur le protocole Michael, qui est plus puissant que le contrôle d'intégrité (ICV) du WEP ;
- Le vecteur d'initialisation (*Initialisation Vector*, IV) est beaucoup plus long, 48 bits, contre 24 bits pour le WEP : ceci permet d'éviter complètement la réutilisation des clés RC4 ;
- La clé de cryptage change à chaque paquet ;
- L'IV est également utilisé pour contrer les attaques de relecture ;
- Les clés sont distribuées selon un mécanisme plus souple et plus sûr que celui du WEP.

L'IV utilisé dans le protocole TKIP a une longueur de 48 bits et il est simplement incrémenté à chaque paquet envoyé. Alors que toutes les valeurs possibles d'un IV de 24 bits peuvent être épuisées en quelques heures dans un réseau chargé, il faudrait plusieurs milliers d'années pour épuiser celles d'un IV de 48 bits. Ceci permet de garantir qu'une même clé RC4 ne sera jamais utilisée deux fois de suite par une station.

Les 16 derniers bits de l'IV sont envoyés en clair dans chaque paquet, entre l'en-tête MAC et le champ ID. Les 32 bits restants sont insérés juste avant les données cryptées et constituent ce que l'on appelle **IV étendu** (*Extended IV*).

| 16 bits de l'IV + ... | ID      | 32 bits IV | Données cryptées | ICV crypté |
|-----------------------|---------|------------|------------------|------------|
| 3 octets              | 1 octet | 4 octets   | 0 à 2300 octets  | 4 octets   |

Figure 2.06 : Structure du paquet crypté avec TKIP

Le sixième bit du champ ID sert maintenant à indiquer si le champ Extended IV est présent ou non. Avec TKIP, il est toujours égal à 1 car l'Extended IV est toujours présent.

#### a) Calcul de la clé RC4

Le calcul de la clé RC4 se fait de la façon suivante : les 24 premiers bits sont composés des 16 derniers bits de l'IV plus 8 bits choisis de telle sorte que les clés RC4 faibles soient évitées ; les 104 derniers bits sont le résultat de fonctions de *hash* appliquées sur l'IV au complet (48 bits), l'adresse MAC de l'émetteur et la clé temporaire (PTK ou GTK, selon le trafic).

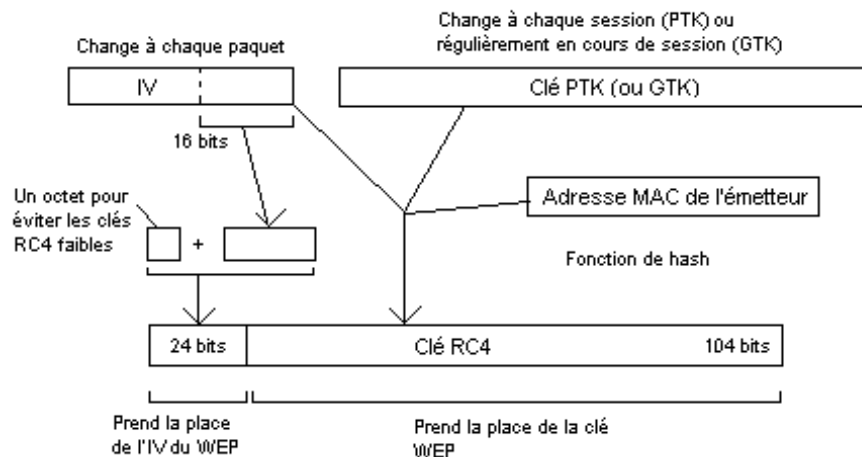


Figure 2.07 : Génération d'une nouvelle clé RC4 à chaque paquet

#### b) Le contrôle d'intégrité Michael

Le contrôle d'intégrité de TKIP repose sur le protocole Michael, développé en 2002 par Neils Ferguson, justement pour TKIP. Ce protocole a été conçu pour répondre aux attentes exprimées par les concepteurs de la solution WPA : offrir une sécurité sur le matériel existant. Michael a donc été conçu pour être très simple à mettre en œuvre, sûr et rapide.

Le protocole Michael fonctionne en calculant un code d'intégrité de message (*Message Integrity Code*, MIC) de 20 bits à partir de la clé d'intégrité (de 128 bits), de l'adresse MAC de destination, de l'adresse MAC de l'émetteur et enfin du message non crypté.

Ce MIC est rajouté à la fin du message non crypté et est crypté avec lui (figure 2.08). Le protocole Michael travaille au niveau du MSDU (*Mac Service Data Unit*). En d'autres termes, il est calculé sur le message à envoyer, avant une éventuelle fragmentation et est rajouté à celui-ci. A la réception, c'est donc après la reconstitution d'un MSDU à partir d'un ou plusieurs fragments (MPDU) que le MIC peut être validé.

#### 2.3.1.2.2. La solution AES

##### a) Le WPA/AES

La sécurité et la performance de l'AES sont les deux raisons pour lesquelles le groupe de travail du 802.11i s'est tourné vers cet algorithme lorsqu'il cherchait une solution de sécurité plus sûre

que le WEP. Ce groupe a alors développé une solution de sécurité complète reposant sur cet algorithme AES. Cette solution inclut :

- Une authentification forte reposant sur le protocole 802.1x ;
- Un mécanisme de distribution automatique des clés ;
- Un contrôle d'intégrité puissant ;
- Un mécanisme empêchant toute attaque de relecture.

Des stations WPA/TKIP et WPA/AES peuvent coexister sur un même réseau, pourvu que les AP gèrent ce nouveau mode mixte. Comme pour le mode mixte WEP+WPA, le trafic de groupe dans un réseau mixte TKIP+AES est crypté avec l'algorithme le moins fort des deux, c'est-à-dire en l'occurrence le TKIP.

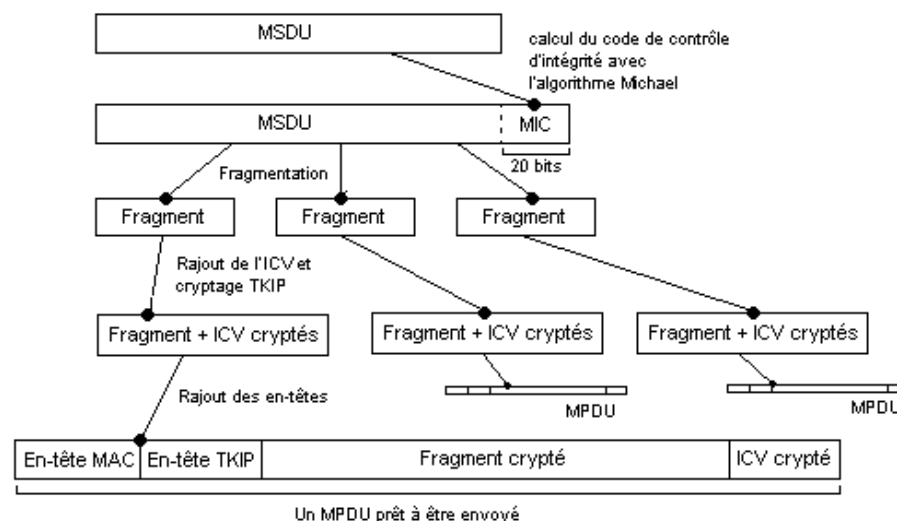


Figure 2.08 : Le contrôle d'intégrité Michael

Le WPA/AES repose quant à lui sur un mode inventé par le groupe de travail 802.11i : ce mode porte le nom de *Counter-Mode+CCM-MAC* (CCM). Comme son nom l'indique, le CCM repose sur le Counter-Mode pour le cryptage. En outre, l'algorithme de contrôle d'intégrité CBC-MAC est utilisé. La même clé est utilisée pour le cryptage et pour le calcul du code CBC. Le CCM rajoute en outre un élément essentiel : il utilise un nonce de 48 bits pour crypter et calculer le CBC de chaque message, afin que deux messages identiques envoyés avec la même clé ne donnent

jamais le même résultat. Ce nonce est en fait séquentiel et il est également utilisé pour éviter les attaques de relectures. On l'appelle le numéro de paquet (*Packet Number*, PN).

Enfin, le mode CCM rajoute également une fonctionnalité intéressante : le code CBC peut être calculé sur le message crypté *plus* des éléments non cryptés. Cela peut permettre au récepteur de s'assurer qu'un champ non crypté, comme l'adresse MAC source d'un paquet par exemple, n'a pas été modifiée par un pirate.

b) Le CCMP (Counter-Mode/Cipher Block Chaining Message Authentication Protocol)

Le rôle du CCMP est de préciser exactement comment le CCM doit être utilisé dans le contexte du WiFi.

Rappelons brièvement comment un paquet à envoyer est traité par la couche MAC :

- Le paquet fourni à la couche MAC par les couches réseau supérieures s'appelle MSDU (*Mac Service Data Unit*) ;
- La couche MAC commence éventuellement par fragmenter le MSDU en plusieurs fragments appelés MPDU (*Mac Protocol Data Unit*) ;
- Chaque MPDU est composé d'un en-tête MAC et des données.

Un en-tête est rajouté entre l'en-tête MAC et les données : il a une structure semblable à celle de l'en-tête TKIP et contient le numéro de paquet (PN) de 48 bits utilisé par CCM, ainsi que l'index de la clé temporaire (PTK ou GTK) utilisée pour le cryptage.

| PN0     | PN1     | Rsv     | ID      | PN2     | PN3     | PN4     | PN5     |
|---------|---------|---------|---------|---------|---------|---------|---------|
| 1 octet | 1 octet | 1 octet | 1 octet | 1 octet | 1 octet | 1 octet | 1 octet |

Figure 2.09 : Structure de l'en-tête CCMP

On voit que les 48 bits du PN, de PN0 à PN5 ne sont pas contigus.

Le but est de conserver une structure similaire à celle de l'en-tête TKIP et WEP. On retrouve notamment le champ ID du WEP et de TKIP. Le troisième octet est réservé pour un usage futur. Cet en-tête CCMP est donc rajouté entre l'en-tête MAC et les données cryptées par le Counter-

Mode/AES. Seule la moitié du code CBC est conservée pour former le code d'intégrité. Ce code est appelé *Message Integrity Code* (MIC) et est rajouté à la fin des données à crypter.

| En-tête MAC | En-tête CCMP | Données cryptées | MIC crypté | CRC      |
|-------------|--------------|------------------|------------|----------|
| 30 octets   | 8 octets     | 0 à 2296 octets  | 8 octets   | 4 octets |

Figure 2.10 : Structure d'un paquet crypté avec le WPA/AES

Avec le CCMP, le code d'intégrité MIC est calculé sur l'ensemble du message plus l'en-tête CCMP et MAC, hormis les champs modifiables. Contrairement au WEP et à TKIP, les en-têtes sont donc protégées.

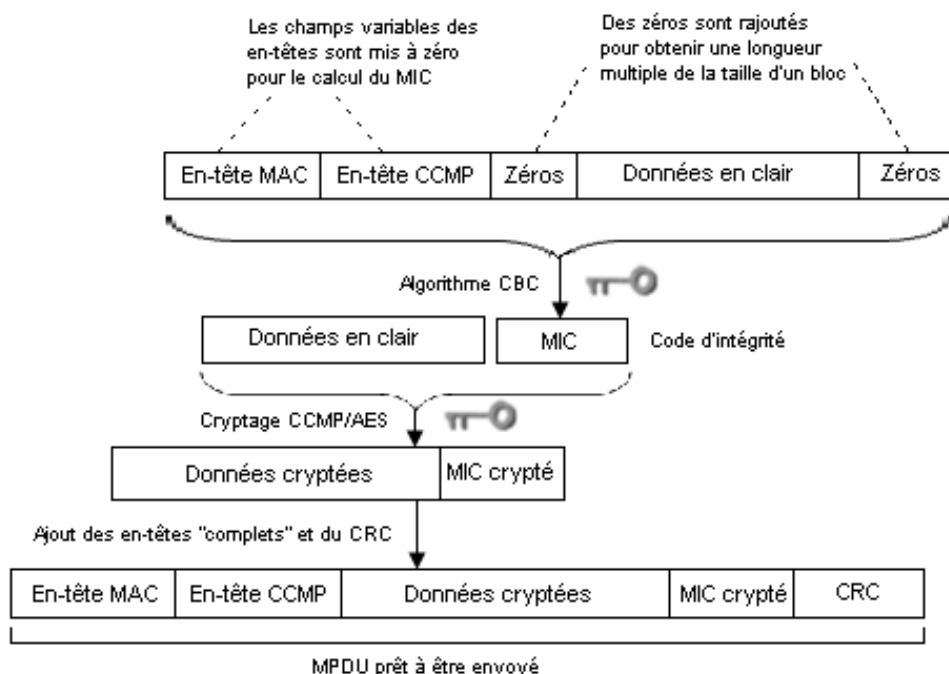


Figure 2.11 : Le calcul de l'intégrité de CCMP.

Le cryptage du CCMP est réalisé avec le CCM/AES. L'algorithme de cryptage est le Counter-Mode /AES, mais le compteur sur lequel repose cet algorithme contient le PN et l'adresse MAC de l'émetteur, plus une partie qui augmente pour chaque bloc crypté, selon le principe du Counter-Mode. Comme pour le calcul du MIC, ceci permet de garantir que deux stations possédant la même clé, le même PN et envoyant le même message n'obtiendront jamais deux fois le même résultat.

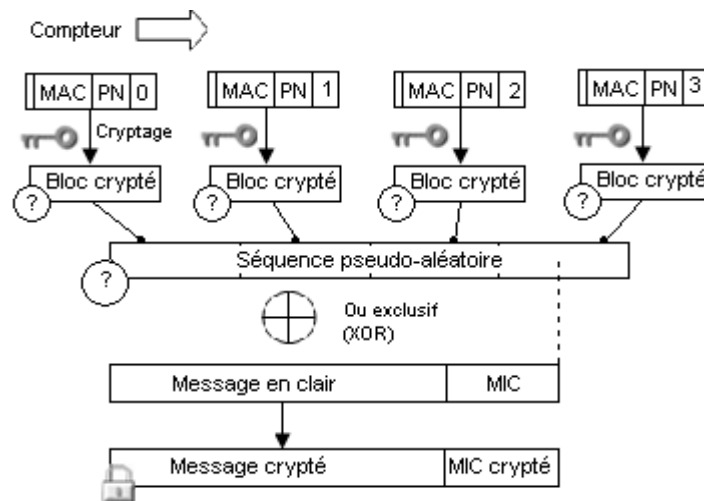


Figure 2.12 : Le cryptage de CCMP

### 2.3.1.3. Le IEEE 802.11i

Le 802.11i a été ratifié le 24 juin 2004, afin de fournir une solution de sécurisation poussée des réseaux Wifi. Il s'appuie sur l'algorithme de chiffrement TKIP, comme le WEP, mais supporte également l'AES (*Advanced Encryption Standard*), beaucoup plus sûr.

La Wifi Alliance a ainsi créé une nouvelle certification, baptisée **WPA2**, pour les matériels supportant le standard 802.11i (ordinateur portable, PDA, carte réseau, etc.).

Contrairement au WPA, le WPA2 permet de sécuriser aussi bien les réseaux sans fil en mode infrastructure que les réseaux en mode ad-hoc.

### 2.3.2. Mécanisme d'authentification

L'IEEE a spécifié deux algorithmes d'authentification pour les réseaux 802.11. Le premier type est une authentification ouverte qui correspond à un algorithme nul : toute station qui demande une authentification obtient un accès. La seconde forme d'authentification est dite « à clé partagée » et exige que la station qui demande et celle qui accorde l'accès soient toutes deux configurées avec des clés WEP identiques. La première station envoie une demande d'authentification à la deuxième qui lui renvoie une trame test en clair. Le demandeur crypte la trame test avec sa clé WEP et la renvoie à la station d'authentification. Celle-ci tente de décrypter la trame : si le texte décrypté correspond au texte en clair initialement envoyé, elle en conclut que le demandeur possède une clé WEP valide et lui accorde l'accès.

### 2.3.2.1. Le IEEE 802.1x

Le standard 802.1x est une solution de sécurisation, mise au point par l'IEEE en juin 2001, permettant d'authentifier (identifier) un utilisateur souhaitant accéder à un réseau (filaire ou non) grâce à un serveur d'authentification.

Le 802.1x repose sur le protocole **EAP** (*Extensible Authentication Protocol*), défini par l'IETF, dont le rôle est de transporter les informations d'identification des utilisateurs.

#### 2.3.2.1.1. Le protocole EAP (*Extensible Authentication Protocol*)

Le fonctionnement du protocole EAP est basé sur l'utilisation d'un contrôleur d'accès (en anglais *authenticator*), chargé d'établir ou non l'accès au réseau pour un utilisateur (en anglais *supplicant*). Le contrôleur d'accès est un simple garde-barrière servant d'intermédiaire entre l'utilisateur et un serveur d'authentification (en anglais *authentication server*), il ne nécessite que très peu de ressources pour fonctionner. Dans le cas d'un réseau sans fil, c'est le point d'accès qui joue le rôle de contrôleur d'accès.

Le serveur d'authentification (appelé parfois *NAS*, pour *Network Authentication Service*, voire *Network Access Service* (Serveur d'accès réseau)) permet de valider l'identité de l'utilisateur, transmis par le contrôleur réseau, et de lui renvoyer les droits associés en fonction des informations d'identification fournies. De plus, un tel serveur permet de stocker et de comptabiliser des informations concernant les utilisateurs afin, par exemple, de pouvoir les facturer à la durée ou au volume (dans le cas d'un fournisseur d'accès par exemple).

La plupart du temps le serveur d'authentification est un serveur RADIUS (*Remote Authentication Dial In User Service*), un serveur d'authentification standard défini par les RFC 2865 et 2866, mais tout autre service d'authentification peut être utilisé.

Ainsi, le schéma global suivant récapitule le fonctionnement global d'un réseau sécurisé avec le standard 802.1x :

1. Le contrôleur d'accès, ayant préalablement reçu une demande de connexion de la part de l'utilisateur, envoie une requête d'identification ;

2. L'utilisateur envoie une réponse au contrôleur d'accès, qui la fait suivre au serveur d'authentification ;
3. Le serveur d'authentification envoie un « challenge » au contrôleur d'accès, qui le transmet à l'utilisateur. Le challenge est une méthode d'identification. Si le client ne gère pas la méthode, le serveur en propose une autre et ainsi de suite ;
4. L'utilisateur répond au challenge. Si l'identité de l'utilisateur est correcte, le serveur d'authentification envoie un accord au contrôleur d'accès, qui acceptera l'utilisateur sur le réseau ou à une partie du réseau, selon ses droits. Si l'identité de l'utilisateur n'a pas pu être vérifiée, le serveur d'authentification envoie un refus et le contrôleur d'accès refusera à l'utilisateur d'accéder au réseau.

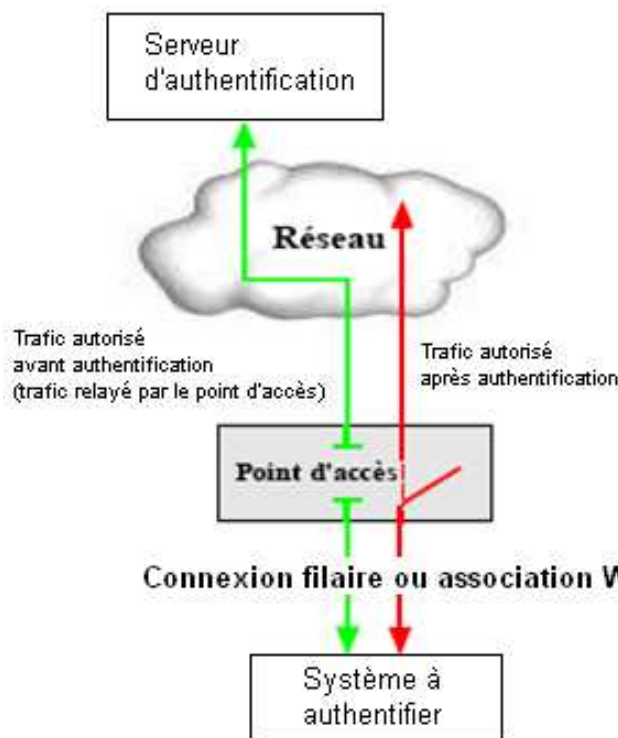


Figure 2.13 : Architecture d'authentification 802.1x

#### 2.3.2.1.2. Echange de clés de chiffrement

Outre l'authentification des utilisateurs, le standard 802.1x est un support permettant de changer les clés de chiffrement des utilisateurs de manière sécurisée, afin d'améliorer la sécurité globale.

### 2.3.3. Autres solutions

#### 2.3.3.1. Mise en place d'un Virtual Private Network

Pour connecter les utilisateurs nomades se branchant au réseau par le biais d'une borne publique, et pour toutes les communications nécessitant un haut niveau de sécurisation, il faut mettre en place un **réseau privé virtuel** (VPN) qui offrira un bon niveau de sécurité et empêchera la plupart des intrusions indésirables.

#### 2.3.3.2. Installer des firewalls

On peut aussi installer un **firewall** comme si le point d'accès était une connexion internet. Ce firewall peut être le serveur **IPsec** (VPN) des clients sans fils. Un réseau WiFi "sécurisé" peut se schématiser comme cela. On considère ici que tout le réseau WiFi est étranger au réseau local, au même titre qu'Internet. L'utilisation d'un pare-feu (firewall) comme pour la connexion Internet, permet de filtrer les adresses MAC associées à des adresses IP fixes. Dans le cas du VPN, le firewall ou un serveur derrière ce dernier fait office de terminal VPN. Certains points d'accès proposent des "petits" firewall permettant de faire un filtrage de plus sur les clients de votre réseau.

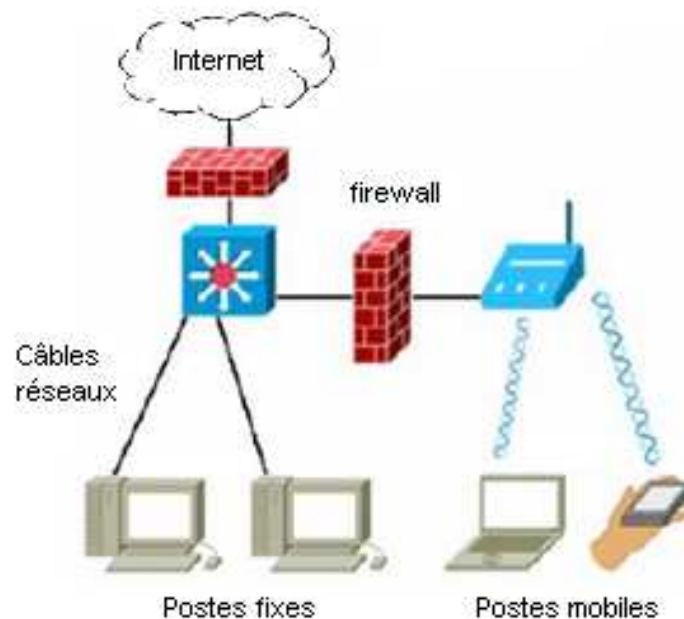


Figure 2.14 : Réseau filaire et WiFi sécurisé

### 2.3.3.3. Le filtrage d'adresse MAC

Chaque *adaptateur réseau* possède une adresse physique qui lui est propre (appelée **adresse MAC**). Cette adresse est représentée par 12 chiffres hexadécimaux groupés par paires et séparés par des tirets.

Les points d'accès permettent généralement, dans leur interface de configuration, de gérer une liste de droits d'accès (appelée *ACL*) basée sur les adresses MAC des équipements autorisés à se connecter au réseau sans fil.

Cette précaution un peu contraignante permet de limiter l'accès au réseau à un certain nombre de machines. En contrepartie cela ne résout pas le problème de la confidentialité des échanges.

## 2.4. Conclusion

Les problèmes de sécurité du WiFi sont liés au support de communication dont on ne peut donner une limite précise dans un espace donné.

Pour sécuriser les réseaux WiFi, les solutions adoptées sont le chiffrement des données et l'authentification de l'utilisateur.

## CHAPITRE 3 : THEORIE DE LA CRYPTOGRAPHIE

Depuis sa création, le réseau Internet a tellement évolué qu'il est devenu un outil essentiel de communication. Cependant, cette communication met de plus en plus en jeu des problèmes stratégiques liés à l'activité des entreprises sur le Web. Les transactions faites à travers le réseau peuvent être interceptées ; il faut donc garantir la sécurité de ces informations, c'est la cryptographie qui s'en charge.

### 3.1. Définitions [20] [21]

Le mot *cryptologie* est un terme utilisé pour désigner la science des messages secrets. Il vient du grec *kryptos* et *logos* qui signifient respectivement « cacher » et « mot ». Longtemps restreinte aux usages diplomatiques et militaires, elle est maintenant une discipline scientifique à part entière dont l'objet est l'étude des méthodes permettant d'assurer les services d'intégrité, d'authenticité et de confidentialité dans les systèmes d'information et de communication.

Les systèmes cryptographiques offrent donc trois importants services :

- Un service de *confidentialité* qui garantit que le contenu d'une communication ou d'un fichier n'est pas accessible aux tiers.
- Un service d'*authenticité* qui garantit l'origine d'une communication ou d'un fichier.
- Un service d'*intégrité* qui garantit que le contenu d'une communication ou d'un fichier n'a pas été modifié.

La cryptologie se partage en deux sous-disciplines également importantes : la *cryptographie* dont l'objet est de proposer des méthodes pour assurer les services définis plus haut, et la *cryptanalyse* qui recherche des failles dans les mécanismes ainsi proposés.

Le mot *cryptographie* désigne l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre inintelligibles sans une action spécifique.

La cryptologie est essentiellement basée sur l'arithmétique : il s'agit, dans le cas d'un texte, de transformer les lettres qui composent le message en une succession de chiffres (sous forme de bits dans le cas de l'informatique), puis ensuite de faire des calculs sur ces chiffres pour : d'une part, les modifier de telle façon à les rendre incompréhensibles. Le résultat de cette modification (le message chiffré) est appelé **cryptogramme** (en anglais, *ciphertext*) par opposition au message initial, appelé **message en clair** (en anglais, *plaintext*) ; d'autre part, de faire en sorte que le destinataire saura les déchiffrer.

Le fait de coder un message de telle façon à le rendre secret s'appelle *chiffrement*. La méthode inverse, consistant à retrouver le message original, est appelée *déchiffrement*. Le chiffrement se fait généralement à l'aide d'une *clé de chiffrement*, le déchiffrement nécessite quant à lui une *clé de déchiffrement*. On distingue généralement deux types de clés :

- Les **clés symétriques** : il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète.
- Les **clés asymétriques** : il s'agit de clés utilisées dans le cas du chiffrement asymétrique (aussi appelé *chiffrement à clé publique*). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

On appelle *décryptement* (ou *décryptage*) le fait d'essayer de déchiffrer illégitimement le message, que la clé de déchiffrement soit connue ou non de l'attaquant. Lorsque la clé de déchiffrement n'est pas connue de l'attaquant, on parle de **cryptanalyse** ou **cryptoanalyse**, ou tout simplement, *cassage*.

## 3.2. Cryptographie conventionnelle [3] [20] [21]

### 3.2.1. Chiffrement et déchiffrement

La cryptographie conventionnelle est principalement liée aux services de confidentialité. Elle réalise sur les données  $m$  une transformation

$$c = E_k(m) \quad (3.01)$$

par l'intermédiaire d'un *algorithme de chiffrement*  $E$ . Cet algorithme prend en entrée le message clair  $m$  et un paramètre secret  $k$  qu'on appelle *clé*. Le message  $m$  varie dans un ensemble  $\mathcal{M}$  et la clé  $k$  dans un ensemble  $\mathcal{K}$ . La restauration du texte clair à partir du *chiffré* ou *cryptogramme*  $c$  se fait par un *algorithme de déchiffrement*  $D_k$ , prenant en entrée le chiffré et la même clé.

On doit avoir :

$$D_k(E_k(m)) = m \quad (3.02)$$

En général, le chiffré prend sa valeur dans le même espace  $\mathcal{M}$  et l'on a aussi

$$E_k(D_k(c)) = c \quad (3.03)$$

c'est-à-dire que les algorithmes  $E_k$  et  $D_k$  réalisent une permutation de  $\mathcal{M}$ .

### 3.2.2. Décryptement

L'opération consiste à calculer le clair  $m$  à partir du chiffré  $c$  (cf. Equation (3.01)), mais sans la connaissance de la clé  $k$  est appelée *décryptement*. La confidentialité est assurée si cette opération est impossible. On distingue divers scénarios possibles d'attaque :

- Les attaques à chiffré seul, où l'adversaire dispose d'un certain nombre de chiffrés  $E_k(m_i)$  ;
- Les attaques à clair connu, où l'adversaire dispose d'un certain nombre de chiffrés  $E_k(m_i)$  et des clairs correspondants  $m_i$  ;
- Les attaques à clair choisi, où l'adversaire dispose d'un certain nombre de chiffrés  $E_k(m_i)$  correspondant à des clairs de son choix  $m_i$  ; si de plus chaque message  $m_i$  est défini en fonction des chiffrés obtenus antérieurement, on parle d'attaque à clair choisi adaptative.

Le but pour l'attaque est la découverte de la clé ou le décryptement d'un chiffré  $c$ , correspondant à un clair dont on ne dispose pas.

### 3.2.3. Chiffrement par bloc

#### 3.2.3.1. Principe

Dans les algorithmes de chiffrement par bloc, l'espace des messages est de la forme  $\{0,1\}_b$ . Autrement dit, le clair (comme le chiffré) est une suite de  $b$  bits. Des messages de taille supérieure à  $b$  sont chiffrés en les complétant à un multiple de  $b$  bits, par une règle de formatage convenue et en chiffrant bloc par bloc. Il existe plusieurs modes d'opération. Le mode ECB (*Electronic Block Chaining*) chiffre successivement chaque bloc. Le mode CBC (*Cipher Block Chaining*) fait le « OU exclusif » de chaque bloc avec le chiffré précédent avant d'appliquer l'algorithme de chiffrement, soit

$$c_i = E_k(c_{i-1} \oplus m_i) \quad (3.04)$$

On peut convenir que pour chiffrer le premier bloc  $m_1$ , on prend  $c_0$  nul ou ajouter un vecteur d'initialisation  $IV$ , transmis en clair, et poser

$$c_0 = IV \quad (3.05)$$

Le déchiffrement calcule  $m_i$  par :

$$c_{i-1} \oplus D_k(c_i) \quad (3.06)$$

DES (*Data Encryption Standard*), Blowfish et AES (*Advanced Encryption Standard*) sont des exemples courants d'algorithmes de chiffrement par bloc.

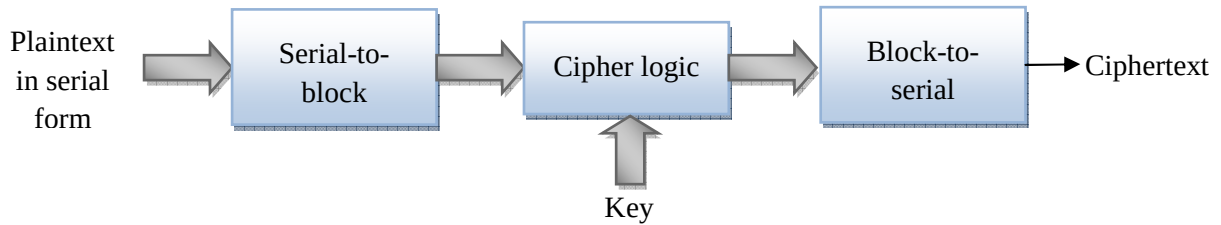


Figure 3.01 : Bloc diagramme d'un chiffrement par bloc

### 3.2.3.2. Mode de chiffrement

Les entiers **b** et **k** sont respectivement la taille en bits d'un bloc et celle de la clé. Un système de chiffrement par bloc est une paire de fonctions **E** et **D** associant à toute clé **k** de  $K = \{0, 1\}^k$  une permutation  $E_k$  de l'espace des blocs  $\{0, 1\}^b$ , et la permutation inverse  $D_k$ .

On appelle *mode d'opération* d'un système de chiffrement par bloc une technique qui permet, avec une unique clé **k**, de chiffrer ou déchiffrer un message dont la longueur n'est pas nécessairement **b**, au moyen d'un certain nombre d'appels de la fonction  $E_k$  ou  $D_k$ .

#### 3.2.3.2.1. Mode ECB (*Electronic CodeBook*)

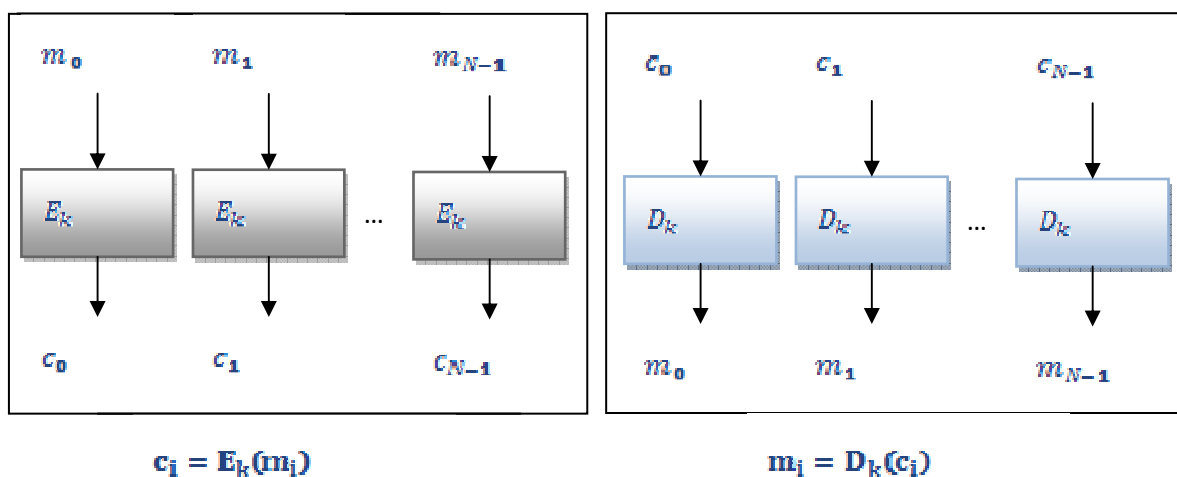


Figure 3.02 : Chiffrement et déchiffrement en mode ECB

Le message est découpé en blocs de taille **b**. Chaque bloc est chiffré par  $E_k$ , le chiffré est la concaténation des blocs obtenus (cf. figure 3.02).

Le déchiffrement se fait de façon similaire. Le message est découpé en blocs de taille **b**. Chaque bloc est déchiffré séparément par  $D_k$ , le clair est la concaténation des blocs obtenus.

#### 3.2.3.2.2. Le Counter – Mode

Un mode très apprécié pour les algorithmes par bloc est le *Counter-Mode* (CM), c'est-à-dire le « mode compteur ». Il est utilisé depuis plus de vingt ans et est considéré comme très sûr. Son principe est légèrement plus complexe que celui d'ECB :

- Un compteur est incrémenté sans arrêt ;
- Ce compteur lui-même est crypté avec l'algorithme de cryptage par bloc choisi ;
- Ceci produit un flux infini de bits pseudo-aléatoires.

Ce flux est simplement combiné avec le message, grâce à l'opération XOR.

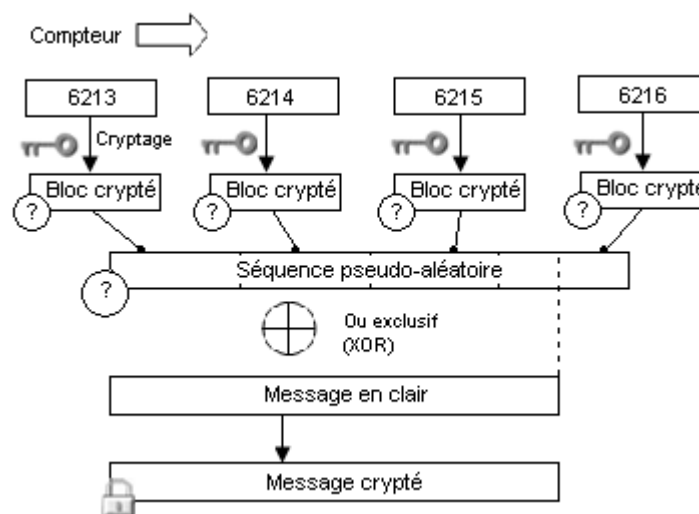


Figure 3.03 : Le Counter-Mode

La taille du message à crypter n'a plus besoin d'être un multiple de la taille du bloc. L'opération de déchiffrement est strictement identique à l'opération de cryptage.

### 3.2.3.2.3. Mode CBC (Cipher Block Chaining)

Les correspondants ont convenu au préalable d'une valeur publique **IV** faisant **b** bits. Le message est découpé en blocs de taille **b**. Le chiffrement d'un bloc se calcule en chiffrant par  $E_k$  le ou bit-à-bit (XOR) du bloc clair et du bloc chiffré précédent. La valeur **IV** sert de chiffré précédent le bloc 0.

Le déchiffrement se fait de façon similaire. Un bloc de clair est obtenu en déchiffrant avec  $D_k$  puis en faisant un ou bit-à-bit avec le bloc chiffré précédent.

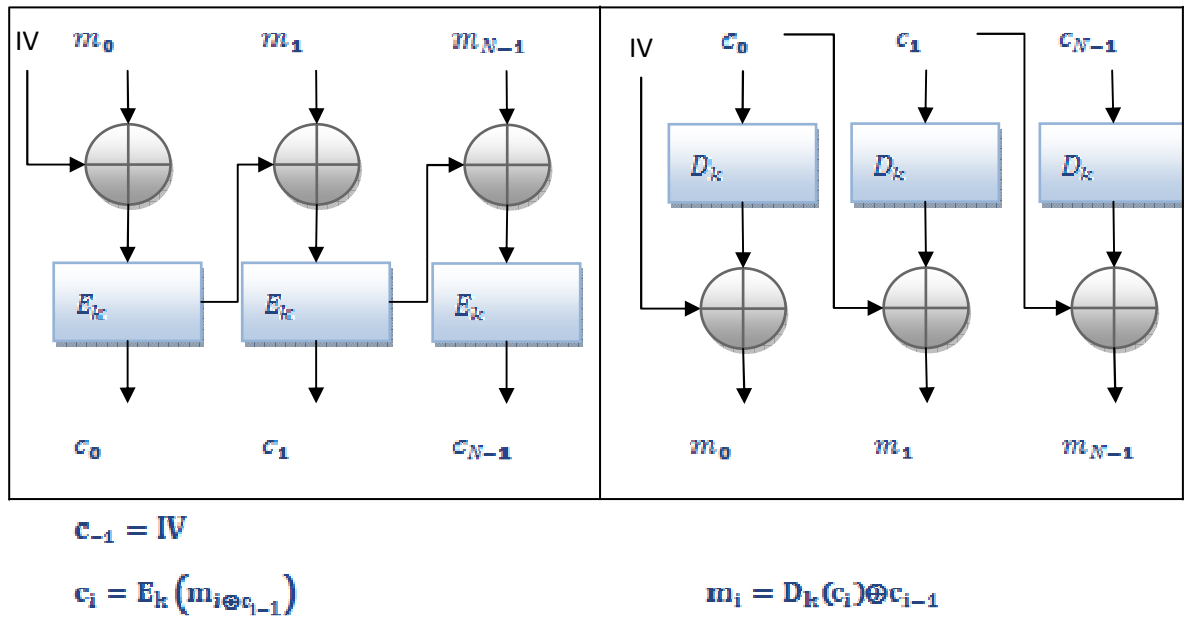


Figure 3.04 : Chiffrement et déchiffrement en mode CBC

### 3.2.3.2.4. Mode CFB (Cipher FeedBack)

Le mode CFB fabrique un registre à décalage en mode CTAK (CipherText Auto Key), pour obtenir un chiffrement de flot.

Le mode CFB est paramétré par un entier **l** inférieur à **b** et utilise un registre de **b** bits, initialisé par une valeur publique **IV** (Initial Value) convenue à l'avance. Le message est découpé en blocs de taille **l**. A chaque coup d'horloge, on calcule l'image du registre par  $E_k$ , dont on extrait **l** bits qu'on appelle  $r_i$ . Le bloc est chiffré par ou exclusif :

$$c_i = m_i \oplus r_i \quad (3.07)$$

La nouvelle valeur du registre est obtenue en faisant un décalage de  $l$  bits et en y entrant la valeur  $c_i$ .

Chiffrement :

$$s_{-1} = IV \quad (3.08)$$

$$r_i = [E_k(s_{i-1})]_{(1..l)} \quad (3.09)$$

$$c_i = m_i \oplus r_i \quad (3.10)$$

$$s_i = [s_{i-1} \ll (l+1 \dots b)] \parallel c_i \quad (3.11)$$

Déchiffrement :

On a toujours les équations (3.08), (3.09) et (3.11), mais l'équation (3.10) devient :

$$m_i = c_i \oplus r_i \quad (3.12)$$

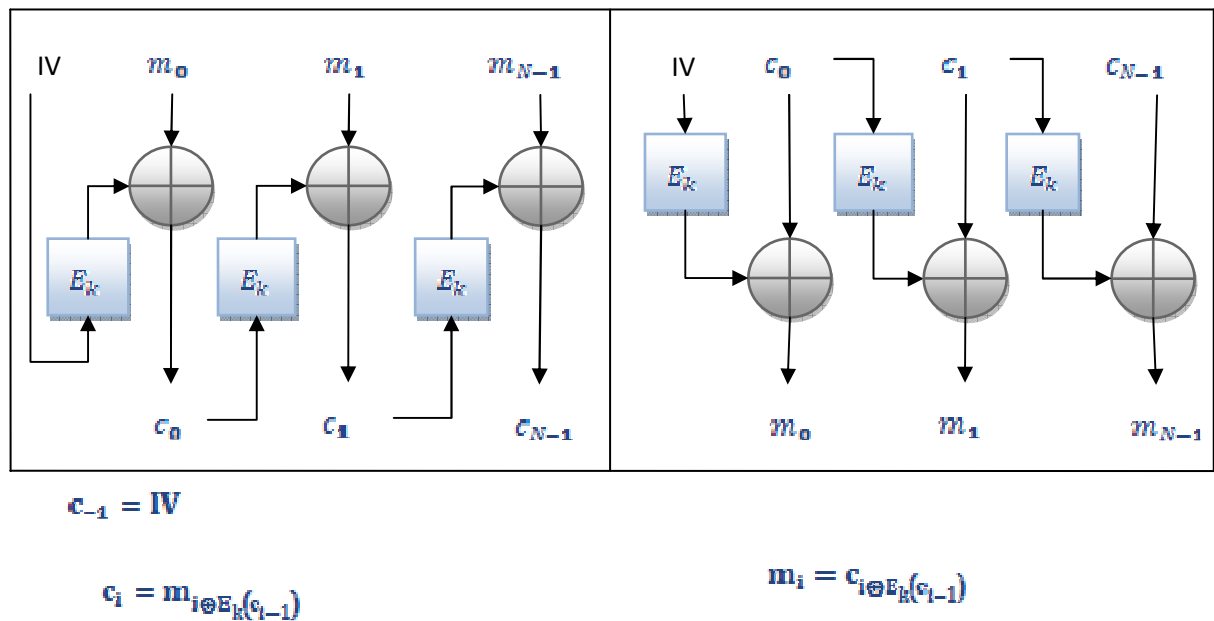


Figure 3.05 : Chiffrement et déchiffrement en mode CFB, avec  $l=b$ .

### 3.2.3.2.5. Mode OFB (Output FeedBack)

Le mode OFB fabrique un registre à décalage en mode KAK (Key Auto Key) pour obtenir un chiffrement de flot.

Le mode OFB est paramétré par un entier  $l$  inférieur à  $b$  et utilise un registre de  $b$  bits, initialisé par une valeur publique IV (Initial Value) convenue à l'avance. Le message est découpé en blocs

de taille  $l$ . A chaque coup d'horloge, on calcule l'image du registre par  $E_k$ , dont on extrait  $l$  bits qu'on appelle  $r_i$ . Le bloc est chiffré par ou exclusif (équation 3.07). La nouvelle valeur du registre est obtenue en faisant un décalage de  $l$  bits, et en y entrant la valeur  $r_i$ .

Pour le chiffrement et le déchiffrement, on a les mêmes équations que (3.08), (3.09), (3.10), (3.11) et (3.12).

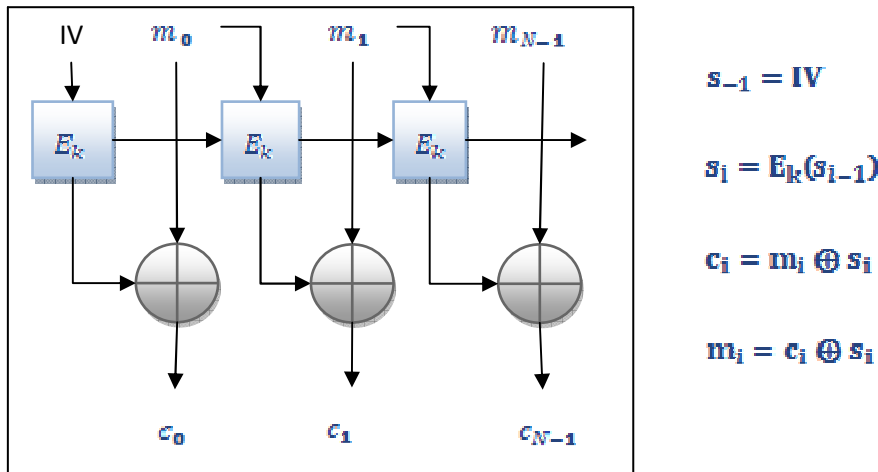


Figure 3.06 : Chiffrement ou déchiffrement en mode OFB

### 3.2.4. Chiffrement par flot

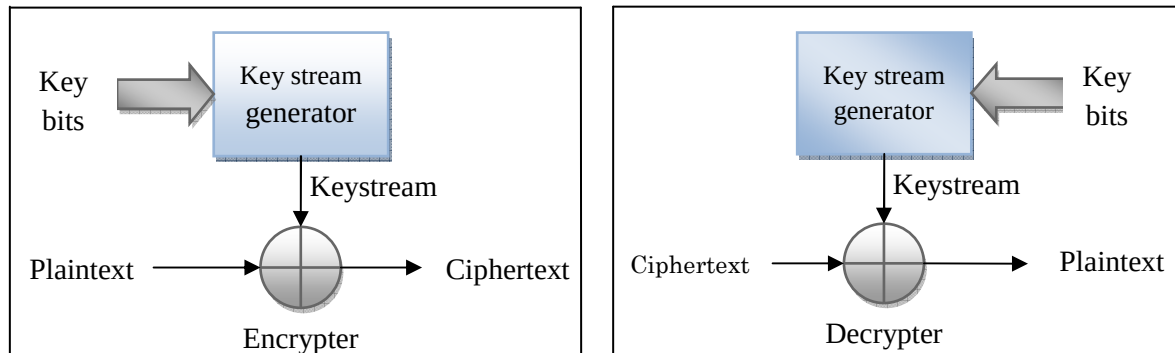


Figure 3.07 : Principe de chiffrement et de déchiffrement par flot

Dans les algorithmes de chiffrement par flot (figure 3.07), une suite d'octets ou de bits  $r_i$  est produite à partir de la clé. Cette suite est combinée aux octets ou aux bits du clair  $m_i$  pour donner les octets ou les bits du chiffré  $c_i$ , suivant la formule

$$c_i = m_i \oplus r_i \quad (3.13)$$

L'algorithme de chiffrement par flot le plus courant est RC4.

### 3.2.5. Intégrité et authenticité

Le service d'intégrité est assuré par un algorithme qu'on peut qualifier de conventionnel même si les définitions qui suivent sont récentes. Une fonction de hachage cryptographique  $H$  calcule un condensé de taille fixe à partir d'un message forme d'une suite de bits quelconque.

On requiert qu'il soit pratiquement impossible à un adversaire de calculer des collisions, c'est-à-dire de produire des messages  $m$  et  $m_0$  différentes mais tels que

$$H(m) = H(m_0) \quad (3.15)$$

Cette propriété empêche la substitution d'un message a un autre, si le condense est conserve séparément. La fonction de hachage la plus répandue est la norme SHA-1. Elle produit des condensés de 20 octets.

Les méthodes conventionnelles assurent des services d'authenticité, mais ne garantissent pas la non-répudiation, puisque les clés secrètes sont partagées. Pour authentifier un message, sans le chiffrer, on peut calculer et transmettre le dernier chiffre dans un chiffrement CBC du message, voire une partie de ce chiffre. Le nom traditionnel de cette méthode est CBC-MAC, le mot MAC étant l'acronyme de *Message Authentication Code*.

### 3.3. Le chiffrement par substitution [23]

A chaque lettre ou groupe de lettres, on substitue une autre lettre ou un autre groupe de lettres.

#### 3.3.1. Chiffrement simple

Pour chaque lettre de l'alphabet de base, on se donne une autre lettre utilisée dans le texte chiffré. Le chiffrement de César en est un exemple historique ; les lettres y sont décalées de trois (3) positions.

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |

Tableau 3.01 : Le chiffrement de César

La forme générale des chiffres par décalage de l'alphabet à 26 lettres :

$$E_k(x) = x + k \bmod 26 \quad (3.16)$$

$$D_k(x) = y - k \bmod 26 \quad (3.17)$$

### 3.3.2. Chiffrement par transposition

On procède à un réaménagement de l'ensemble des caractères (une transposition) qui cache le sens initial.

Le plus souvent, on utilise deux visions géométriquement différentes du texte.

Exemple :

- On enroule une fine feuille de papyrus ou de peau sur un tambour d'un diamètre donné (technique assyrienne, 400 av. J.-C.)
- On écrit horizontalement un texte sur la lamelle enroulée.
- Quand la lamelle est déroulée, les lettres sont incompréhensibles.
- Pour déchiffrer le message, il faut un cylindre de bon diamètre.

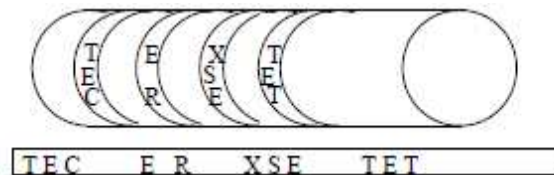


Figure 3.08 : Chiffrement par transposition

### 3.4. Le chiffrement à clé secrète

Le chiffrement symétrique (aussi appelé chiffrement à clé privée ou chiffrement à clé secrète) consiste à utiliser la même clé pour le chiffrement et le déchiffrement.

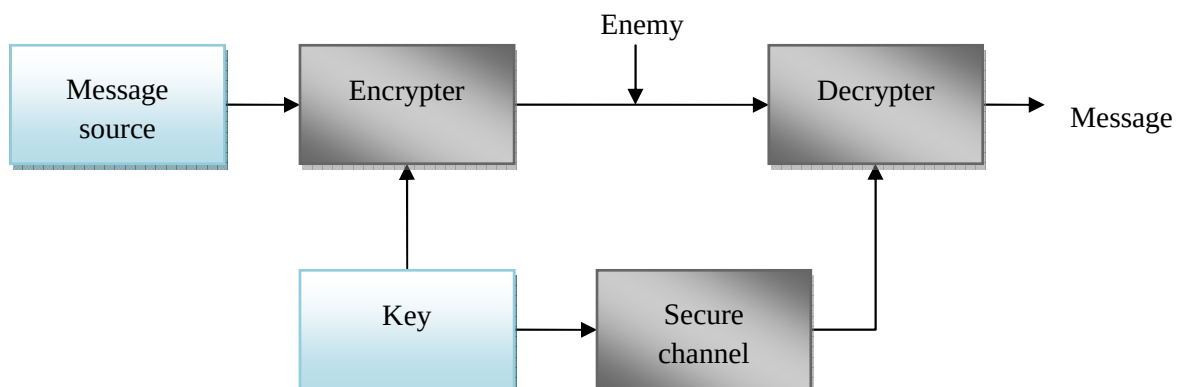


Figure 3.09 : Bloc diagramme d'un système de chiffrement à clé secrète

Le chiffrement consiste à appliquer une opération (algorithme) sur les données à chiffrer à l'aide de la clé privée, afin de les rendre inintelligibles. Ainsi, le moindre algorithme (tel qu'un *OU exclusif*) peut rendre le système quasiment inviolable.

Toutefois, dans les années 40, Claude Shannon démontra que pour être totalement sûr, les systèmes à clés privées doivent utiliser des clés d'une longueur au moins égale à celle du message à chiffrer. De plus le chiffrement symétrique impose d'avoir un canal sécurisé pour l'échange de la clé, ce qui dégrade sérieusement l'intérêt d'un tel système de chiffrement.

Le principal inconvénient d'un cryptosystème à clés secrètes provient de l'échange des clés. En effet, le chiffrement symétrique repose sur l'échange d'un secret (les clés).

D'autre part, un utilisateur souhaitant communiquer avec plusieurs personnes en assurant de niveaux de confidentialité distincts doit utiliser autant de clés privées qu'il a d'interlocuteurs. Pour un groupe de  $N$  personnes utilisant un cryptosystème à clés secrètes, il est nécessaire de distribuer un nombre de clés égal à

$$N * \frac{(N-1)}{2} \quad (3.15)$$

Ainsi, dans les années 20, Gilbert Vernam et Joseph Mauborgne mirent au point la méthode du *One Time Pad* (traduisez *méthode du masque jetable*, parfois appelé « One Time Password » et noté **OTP**), basée sur une clé privée, générée aléatoirement, utilisée une et une seule fois, puis détruite. À la même époque, le Kremlin et la Maison Blanche étaient reliés par le fameux téléphone rouge, c'est-à-dire un téléphone dont les communications étaient cryptées grâce à une clé privée selon la méthode du *masque jetable*. La clé privée était alors échangée grâce à la valise diplomatique (jouant le rôle de canal sécurisé).

### 3.5. Le chiffrement à clé publique [2]

Le principe de chiffrement asymétrique (appelé aussi chiffrement à clés publiques) est apparu en 1976, avec la publication d'un ouvrage sur la cryptographie par Whitfield Diffie et Martin Hellman.

Dans un cryptosystème asymétrique (ou cryptosystème à clés publiques), les clés existent par paires : une clé publique pour le chiffrement et une clé secrète pour le déchiffrement. Ainsi, dans un système de chiffrement à clé publique, les utilisateurs choisissent une clé aléatoire qu'ils sont

seuls à connaître (il s'agit de la clé privée). A partir de cette clé, ils déduisent chacun automatiquement un algorithme (il s'agit de la clé publique). Les utilisateurs s'échangent cette clé à travers un canal non sécurisé.

Lorsqu'un utilisateur désire envoyer un message à un autre utilisateur, il lui suffit de chiffrer le message à envoyer au moyen de la clé publique du destinataire (qu'il trouvera par exemple dans un serveur de clés tel qu'un annuaire LDAP). Ce dernier sera en mesure de déchiffrer le message à l'aide de sa clé privée (qu'il est le seul à connaître).

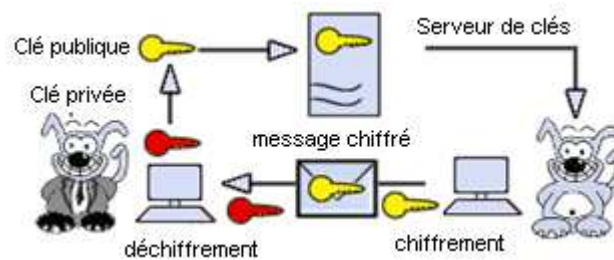


Figure 3.10 : Principe du chiffrement asymétrique

Le problème consistant à se communiquer la clé de chiffrement n'existe plus, dans la mesure où les clés publiques peuvent être envoyées librement. Le chiffrement par clés publiques permet donc à des personnes d'échanger des messages chiffrés sans pour autant posséder de secret en commun.

Ainsi, la notion de clé de session est un compromis entre les chiffrements symétrique et asymétrique permettant de combiner les deux techniques.

### 3.5.1. Notion de clé de session

#### 3.5.1.1. Intérêt d'une clé de session

Les algorithmes asymétriques permettent de s'affranchir de problèmes liés à l'échange de clé via un canal sécurisé. Toutefois, ces derniers restent beaucoup moins efficaces (en terme de temps de calcul) que les algorithmes symétriques.

Le principe de la clé de session est simple : il consiste à générer aléatoirement une clé de session de taille raisonnable, et de chiffrer celle-ci à l'aide d'un algorithme de chiffrement à clé publique (plus exactement à l'aide de la clé publique du destinataire).

Le destinataire est en mesure de déchiffrer la clé de session à l'aide de sa clé privée. Ainsi, expéditeur et destinataires sont en possession d'une clé commune dont ils sont seuls connaisseurs.

Il leur est alors possible de s'envoyer des documents chiffrés à l'aide d'un algorithme de chiffrement symétrique.

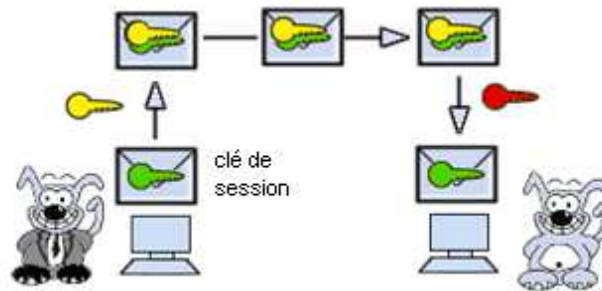


Figure 3.11 : Clé de session

### 3.5.1.2. Algorithme de Diffie-Hellman

L'algorithme de Diffie-Hellman a été mis au point en 1976 afin de permettre l'échange de clés à travers un canal non sécurisé.

## 3.6. L'algorithme RC4 [22]

RC4 est un chiffrement par flot ; c'est-à-dire qu'une suite d'octets ou de bits  $\mathbf{r_i}$  est produite à partir de la clé. Cette suite est combinée aux octets ou bits du message clair  $\mathbf{m_i}$  pour donner les octets ou les bits du message chiffré  $\mathbf{c_i}$ , suivant la formule :

$$\mathbf{c_i = m_i \oplus r_i} \quad (3.17)$$

### 3.6.1. Initialisation de l'algorithme RC4

A partir de la clé K (décomposable en une suite  $\mathbf{k_0 \dots k_{l-1}}$  de  $\mathbf{l}$  mots de  $\mathbf{n}$  bits [ $\mathbf{n=8}$ ]) de longueur **128 bits**, un tableau S de 256 octets est initialisé et deux compteurs  $\mathbf{i}$  et  $\mathbf{j}$ , mis à zéro :

Pour z de 0 à  $2^n-1$

$\mathbf{K[z] = k[z] \bmod l}$

$\mathbf{S[z] = z}$

$\mathbf{j = 0}$  // initialisation de j

Pour i de 0 à  $2^n-1$ ,

$\mathbf{S[i] = i}$

$\mathbf{K[i] = cle}$

$\mathbf{j = (j + S[i] + K[i]) \bmod 2^n-1}$

Echanger S[i] et S[j]

$i = 0, j = 0$  // réinitialisation de  $i$  et  $j$  avant le calcul des clés de session.

### 3.6.2. Génération des clés RC4

Pour générer un nouvel octet aléatoire, on applique les opérations suivantes :

$i = (i + 1) \bmod 2^n$

$j = j + S[i] \bmod 2^n$

$S[i] \leftarrow S[j]$

$S[j] \leftarrow S[i]$

$t = S[i] + S[j] \bmod 2^n$

Retourner  $S[t]$

La suite  $S[t]$  représente les octets de la clé de session.

### 3.6.3. Clés faibles générées par RC4

En septembre 1995, Andrew Roos publia dans le sci.crypt newsgroup un article sur une classe de clés faibles pour l'algorithme RC4. Ces clés sont celles dont les premiers octets présentent une forte corrélation selon la relation suivante :

$$k_0 + k_1 = 0 \bmod 2^n \quad (3.18)$$

En effet, une entrée donnée  $S[z]$  ne subit au maximum qu'une seule permutation. La probabilité que cette permutation ait lieu pour une entrée  $S[z]$  de la s-box  $S$  calculée par Andrew Roos est de  $1/e$ . Selon cet article, la complexité d'une attaque serait réduite de  $2^{5,1}$  ce qui ne représente pas un gain énorme dans un espace de clés de complexité  $2^{128}$  si le vecteur d'initialisation n'est pas connu. En revanche, la complexité se réduirait de  $2^{18}$  pour Oscar, s'il a déjà mené les attaques décrites ci-dessus et possède une clé de session.

## 3.7. L'algorithme AES (Advanced Encryption Standard) [22]

AES, par contre, est un algorithme de chiffrement par bloc. La clé de cryptage est de 128, 192 ou 256 bits.

### 3.7.1. Fonctionnement

Il y a une suite de **r tours**, chacun réalisant une suite de permutations et de substitutions dépendant d'une clé de tour et opérant sur une matrice de 4 x 4, 4 x 6 ou 4 x 8 octets selon la longueur de la clé.

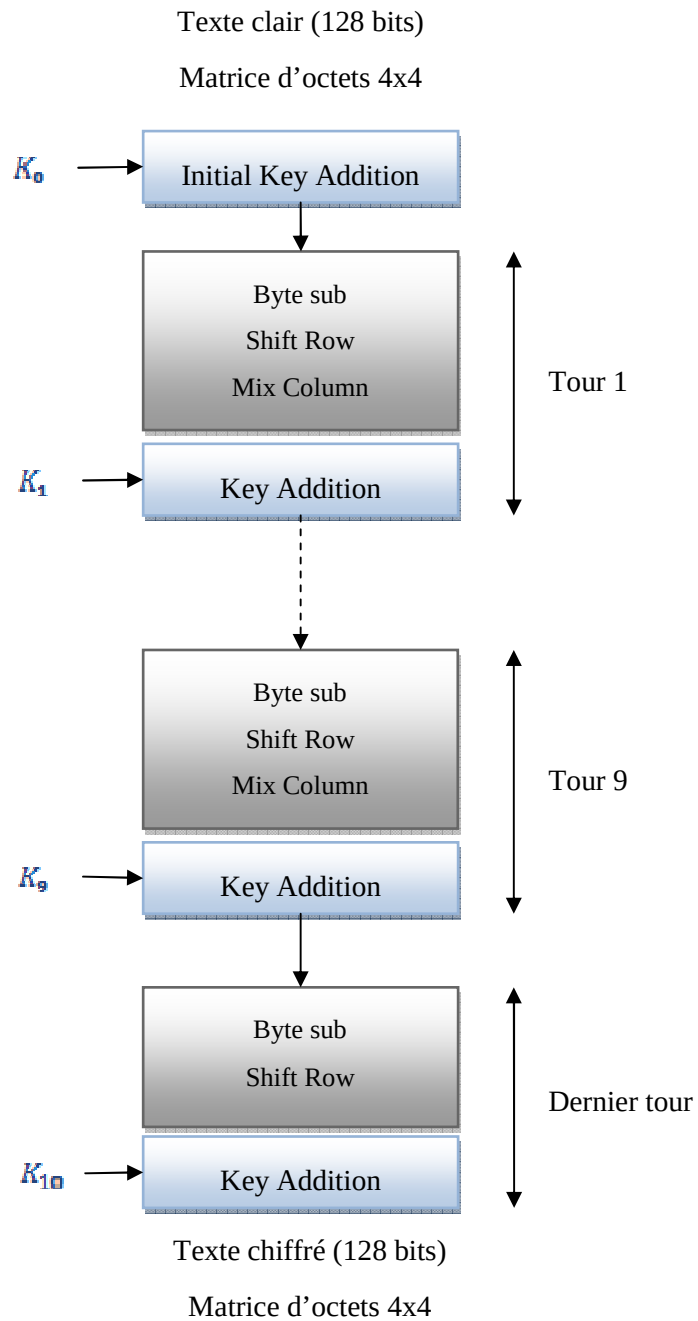


Figure 3.12 : Chiffrement AES

La figure (3.10) illustre le fonctionnement du chiffrement AES pour  $r = 10$ , avec une clé de chiffrement de 128 bits et une matrice d'octets  $4 \times 4$ .

Les fonctionnements des différents blocs (*Byte sub*, *Shift row*, *Mix column* et *Key addition*) sont respectivement décrits dans les figures 3.12, 3.13, 3.14 et 3.15.

### 3.7.2. Algorithme pour le calcul de contrôle d'intégrité Michael utilisé par AES

#### Procedure Michael $((K_0, K_1), (M_0, \dots, M_{N-1}))$

Input: Key  $(K_0, K_1)$ , Padded message (as 32 bitwords:  $M_0, \dots, M_{N-1}$ )

Output: MIC value  $(V_0, V_1)$

$(L, R) \leftarrow (K_0, K_1)$

For  $i = 0$  to  $N-1$  do

$L \leftarrow L \oplus M_i$

$(L, R) \leftarrow B(L, R)$

Return  $(L, R)$

#### Procedure B $(L, R)$

$R \leftarrow R \oplus (L \ll 17)$

$L \leftarrow (L + R) \bmod 2^{32}$

$R \leftarrow R \oplus \text{XSWAP}(L)$

$L \leftarrow (L + R) \bmod 2^{32}$

$R \leftarrow R \oplus (L \ll 3)$

$L \leftarrow (L + R) \bmod 2^{32}$

$R \leftarrow R \oplus (L \ll 2)$

$L \leftarrow (L + R) \bmod 2^{32}$

Return  $(L, R)$

$\ll$  indique une rotation à droite de 32 bits

XSWAP est une fonction qui échange la position des deux LSB (*Least Significant Bytes*) et la position des deux MSB (*Most Significant Bytes*) dans un mot, c'est-à-dire,

**XSWAP(ABCD) = BADC**

où A, B, C et D sont des octets.

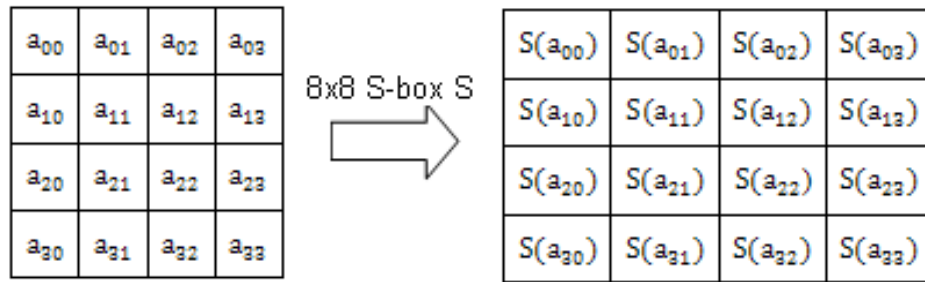


Figure 3.13 : Byte substitution

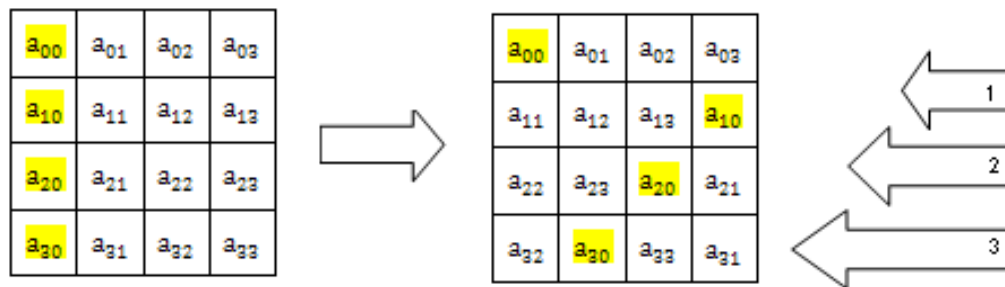


Figure 3.14 : Shift row

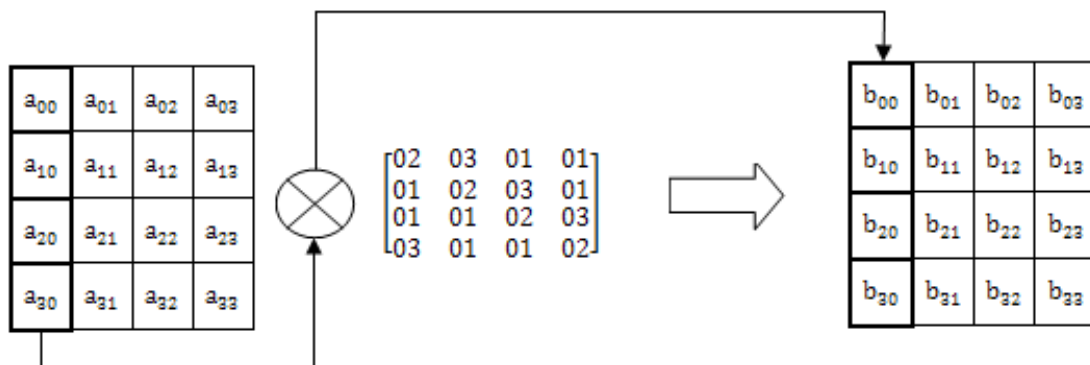


Figure 3.15 : Mix column

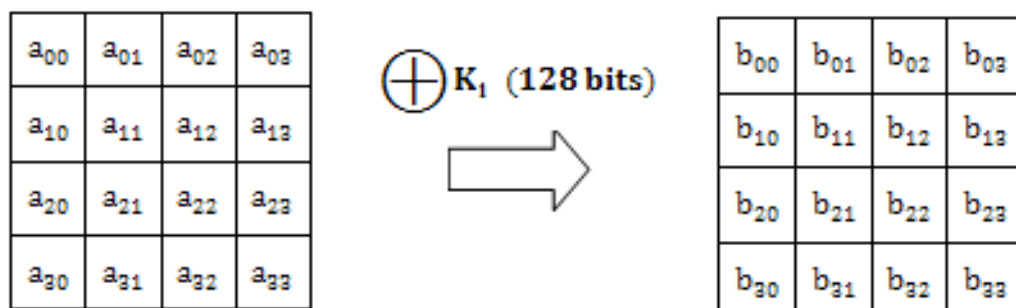


Figure 3.16 : Key addition

### **3.8. Conclusion**

Le chiffrement des données à transmettre sur un réseau est utilisé pour interdire l'accès ou la modification des informations et garantir la confidentialité dans les applications informatiques. Cependant, ce chiffrement n'est qu'un élément dans l'ensemble des dispositifs d'un système complexe.

## CHAPITRE 4 : SIMULATION

Dans ce chapitre, nous allons mettre en œuvre les algorithmes de chiffrement utilisés par le WiFi dans le WEP, le WPA et le 802.11i (WPA2), c'est-à-dire le RC4 et l'AES. Pour ce faire, nous avons élaboré un petit logiciel, réalisé avec le logiciel Eclipse 3.2, et permettant non seulement de chiffrer un fichier texte mais aussi de montrer la structure du paquet selon l'algorithme utilisé.

Les différentes classes que nous avons utilisées sont illustrées dans la figure 4.01 suivante :

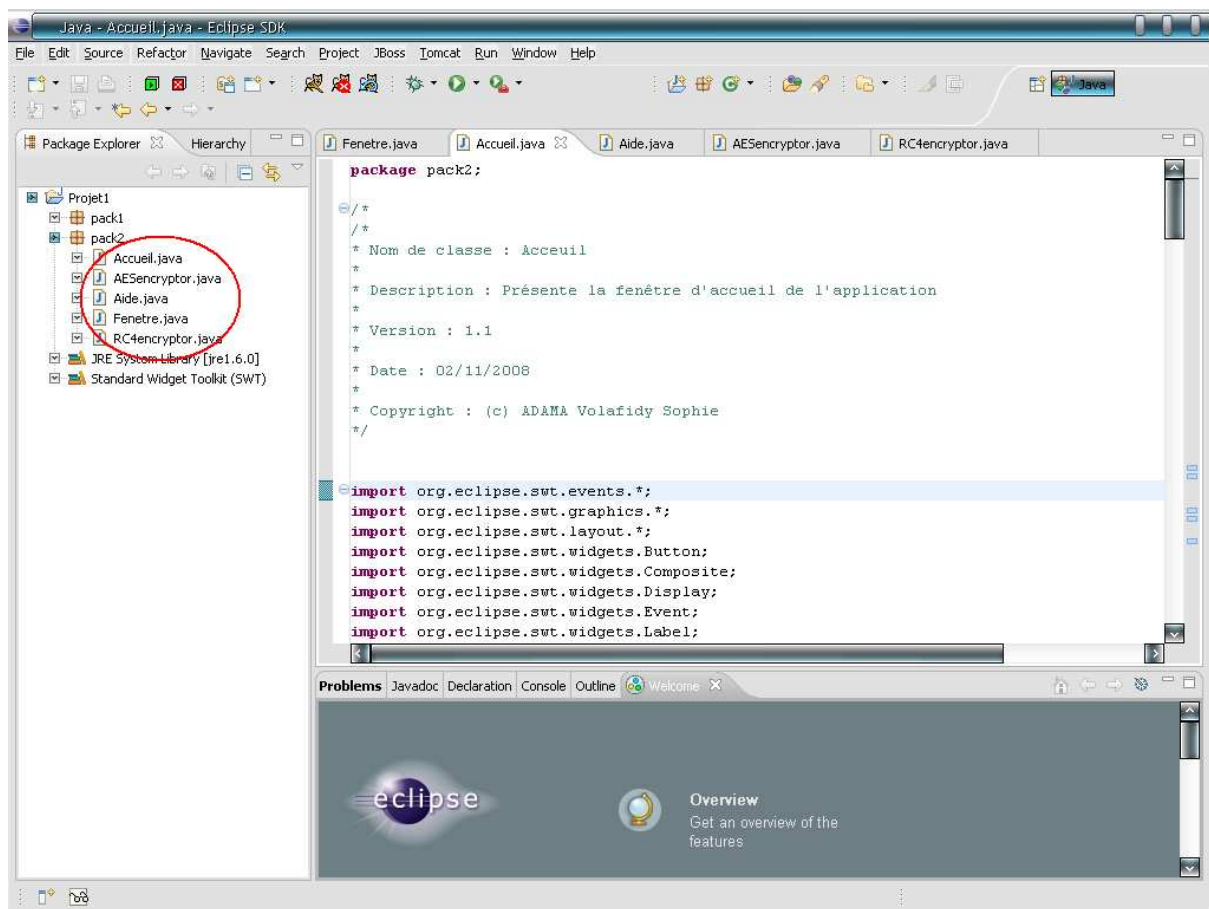


Figure 4.01 : Interface utilisateur du logiciel Eclipse montrant les différentes classes utilisées

Nous avons donc utilisé les classes :

- **Accueil** : classe qui permet d'afficher la fenêtre d'accueil lors du lancement du logiciel ;
- **AESencrytor** : classe qui permet de chiffrer/déchiffrer en utilisant l'algorithme de chiffrement AES ;
- **Aide** : classe qui permet d'afficher l'aide ;
- **Fenetre** : classe qui permet d'afficher la fenêtre principale ;

- **LancerSim** : classe principale ;
- **RC4encryptor** : classe qui permet de chiffrer/déchiffrer en utilisant l'algorithme de chiffrement RC4 ;

#### 4.1. Fonctionnement du logiciel

En lançant le logiciel, la fenêtre d'accueil illustrée par la figure 4.02 apparaît.

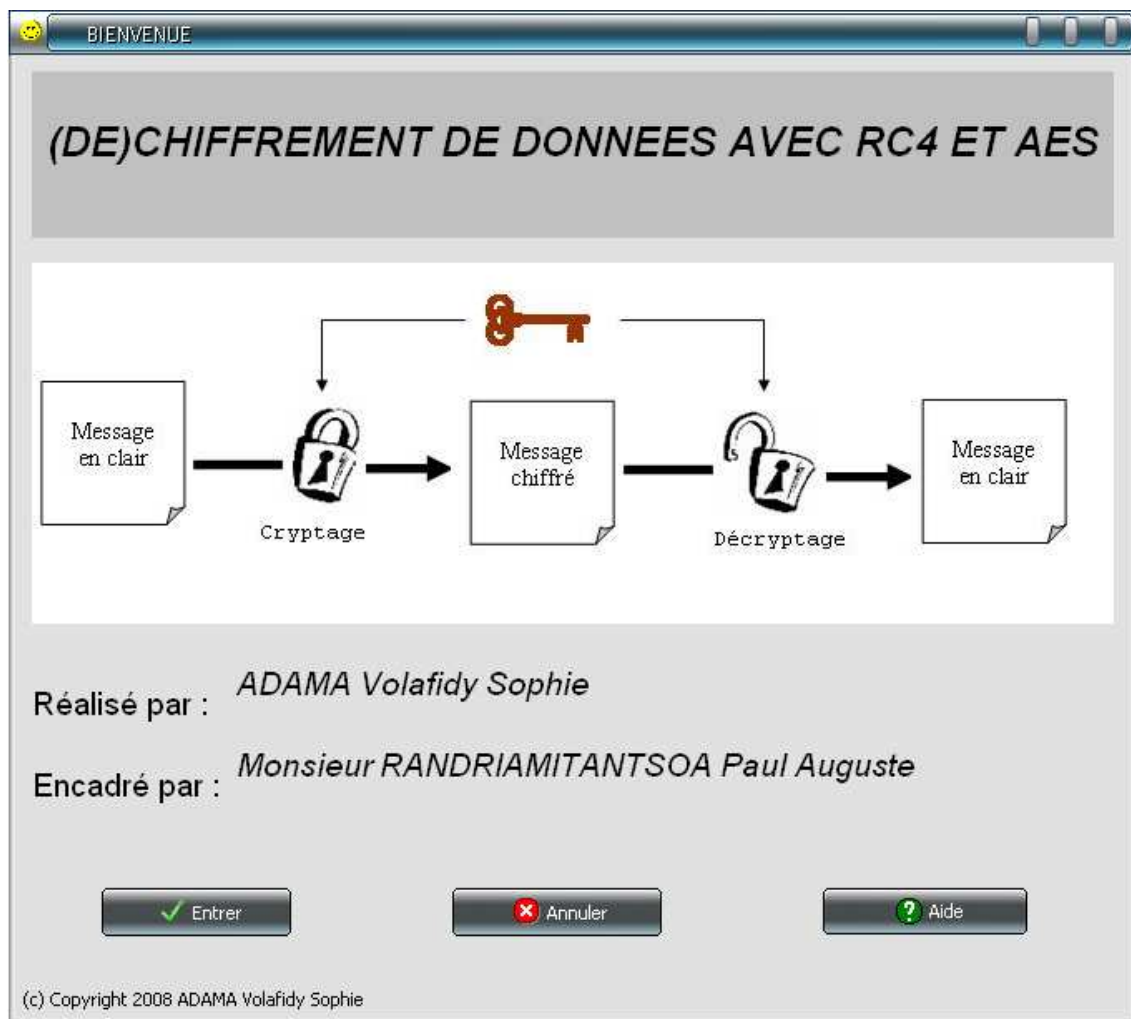


Figure 4.02 : Fenêtre d'accueil

Pour sortir du logiciel, il suffit simplement d'appuyer sur le bouton Annuler ; et pour afficher l'aide, il faut appuyer sur le bouton Aide. En appuyant sur le bouton Entrer, la fenêtre qui s'affiche est illustrée dans la figure 4.03. Remarquez que l'algorithme de chiffrement RC4 est sélectionné par défaut.

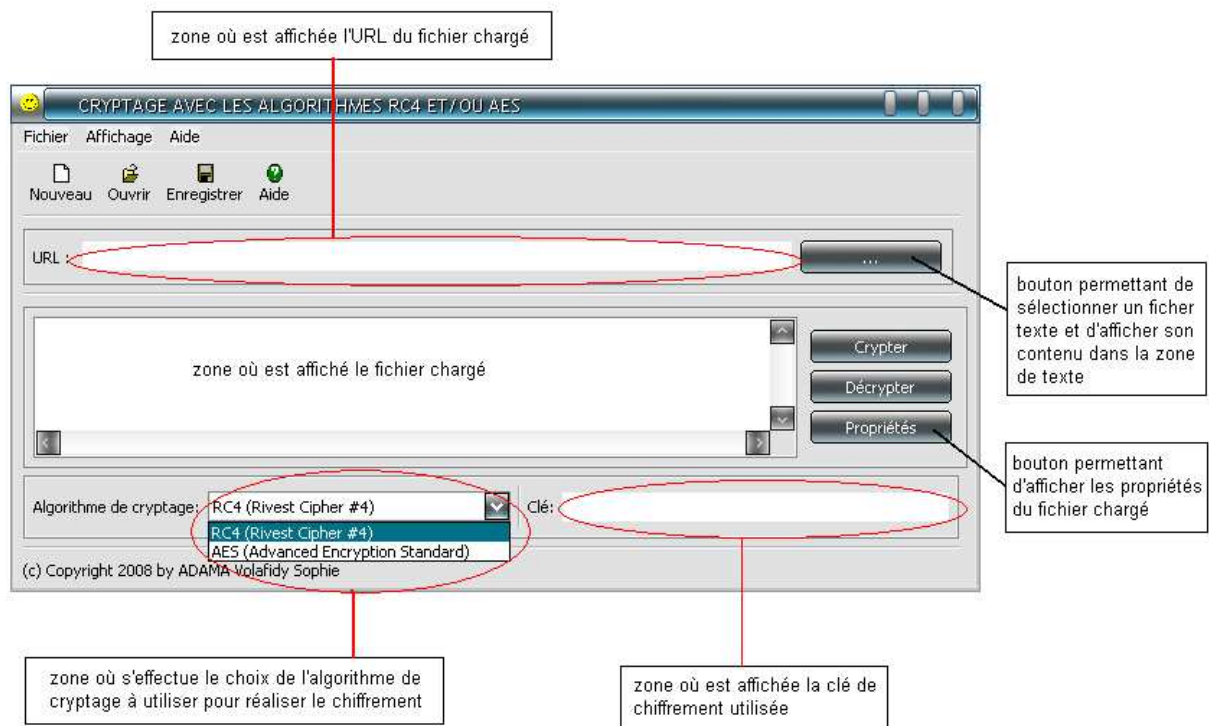


Figure 4.03 : Fenêtre principale

#### 4.1.1. Ouverture d'un fichier

Afin de pouvoir effectuer un chiffrement, il faut tout d'abord charger le fichier concerné ; la zone de texte ne doit pas être vide. Pour ce faire, il existe différentes méthodes :

- Cliquer sur le menu Fichier / Ouvrir ...
- Cliquer sur le bouton 
- Cliquer sur le bouton 

La fenêtre d'ouverture de fichier (figure 4.04) apparaît alors.

Nous avons limité l'extension du fichier à \*.txt, c'est-à-dire à des fichiers texte uniquement. Nous avons choisi de chiffrer le fichier *exemple1.txt*, il suffit ensuite de valider en cliquant sur le bouton Ouvrir. Le contenu du fichier *exemple1.txt* est alors affichée dans la zone de texte réservée à cet effet tandis que son arborescence est affichée juste au-dessus, à côté du texte 'URL' (figure 4.05).

On peut aussi taper directement le texte que nous voulons chiffrer à l'intérieur de la zone de texte.

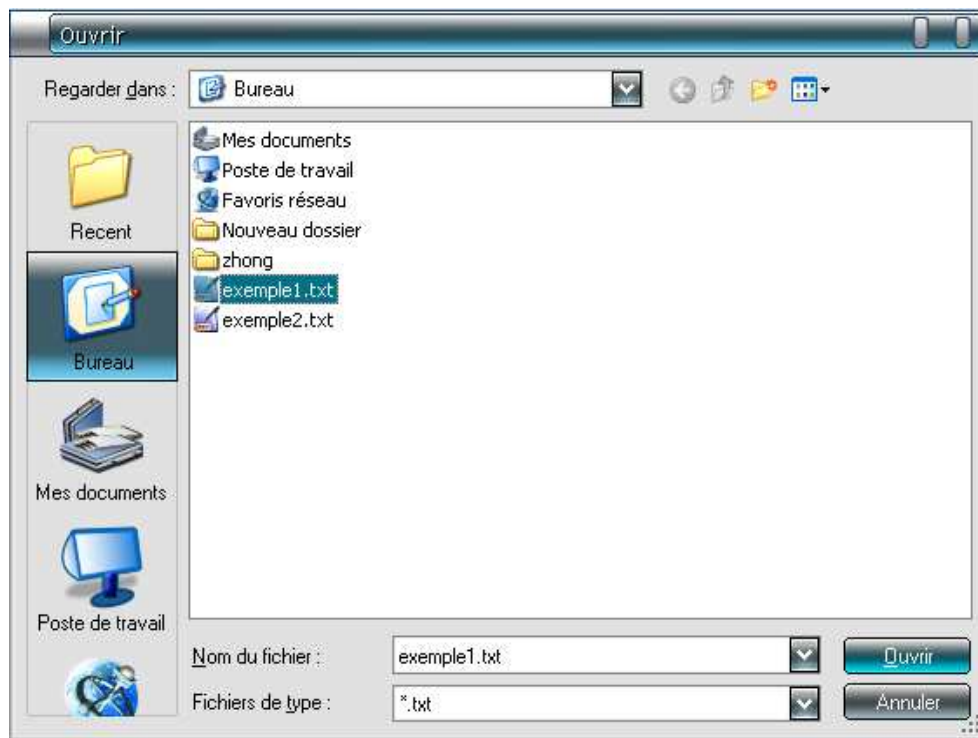


Figure 4.04 : Choix du fichier à chiffrer

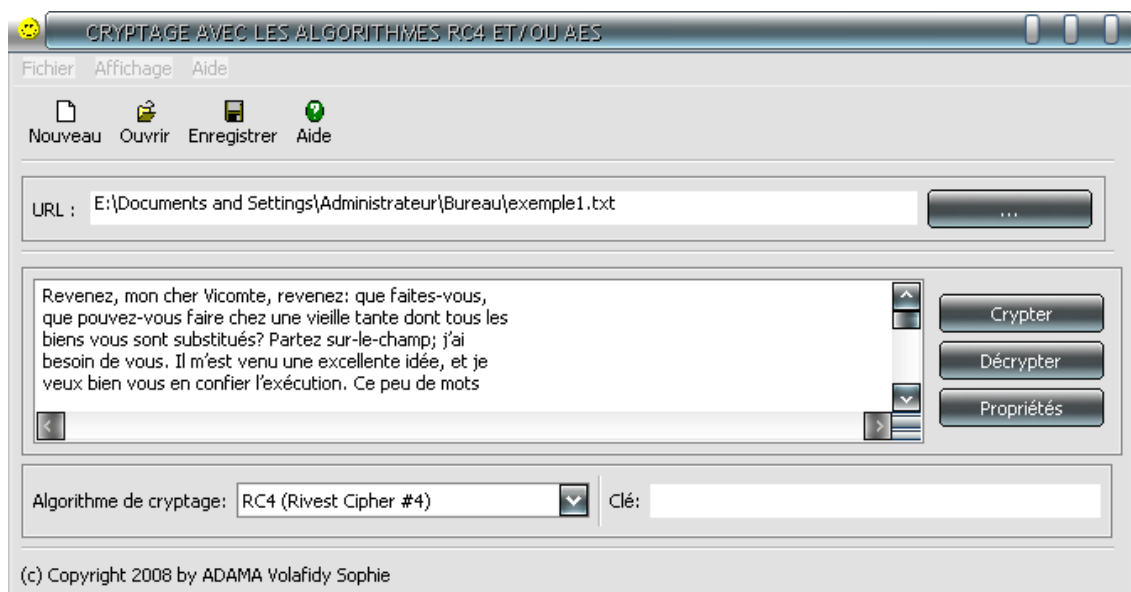


Figure 4.05 : Texte à chiffrer

#### 4.1.2. Affichage des propriétés du fichier

Les propriétés du fichier *exemple1.txt* sont obtenues, soit :

- En cliquant sur le bouton Propriétés qui se situe à droite de la zone de texte ;
- En cliquant sur le menu Afficher/Propriétés.

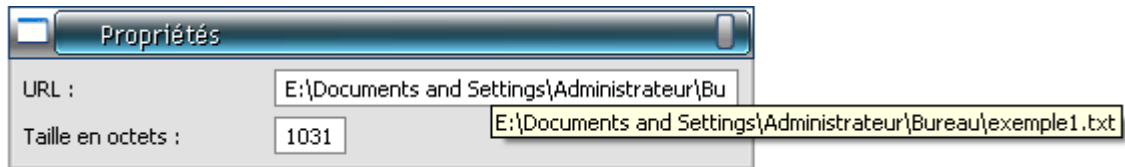


Figure 4.06 : Propriétés de fichier1.txt

La figure 4.06 montre que le fichier *exemple1.txt* se trouve dans « E:\Documents and Settings\Administrateur\Bureau » et que sa taille est de 1031 octets.

#### 4.1.3. Chiffrement en utilisant l'algorithme de cryptage RC4

Nous allons maintenant chiffrer le fichier que nous venons de charger. Pour ce faire, il suffit d'appuyer sur le bouton Crypter.

Afin de prendre en compte le fait que ce fichier allait être émis à travers le canal de transmission et être déchiffré à la réception, le logiciel permet l'enregistrement du fichier crypté (figure 4.07). Nous avons choisi l'extension *\*.crypt* pour représenter les fichiers cryptés.

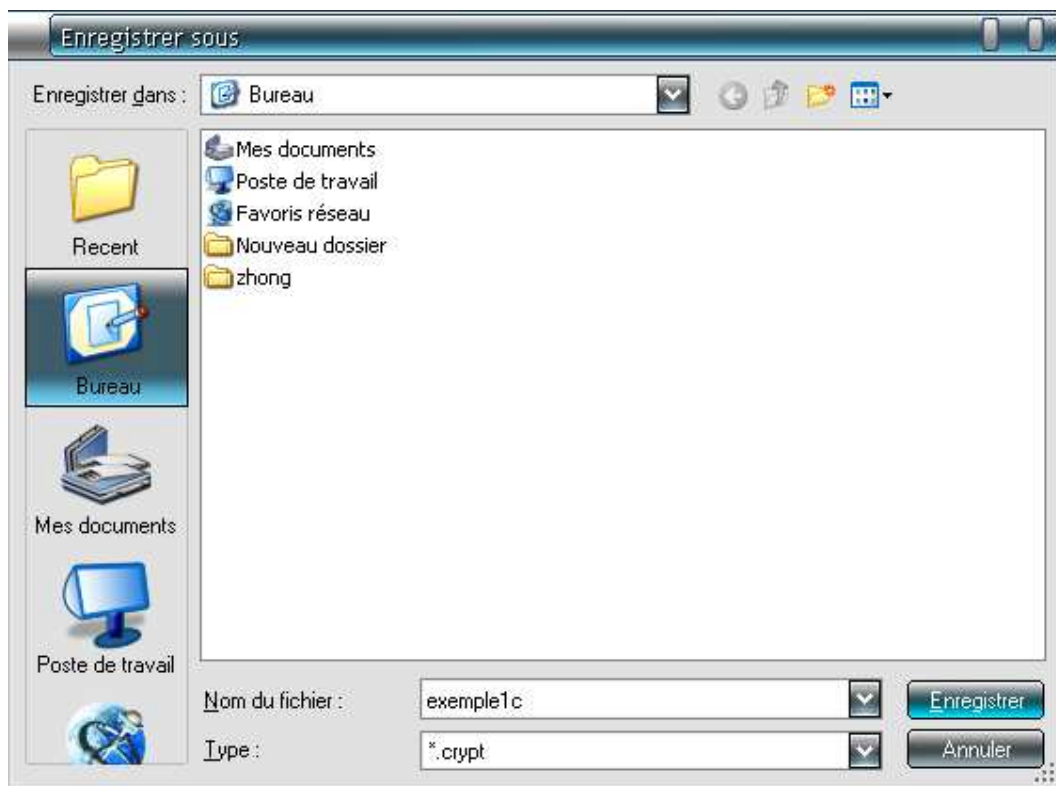


Figure 4.07 : Sauvegarde d'un fichier chiffré

Ensuite, on charge le fichier crypté. On obtient la fenêtre illustrée par la figure 4.08. Comme précédemment, l'arborescence du fichier est aussi affichée dans la fenêtre principale.

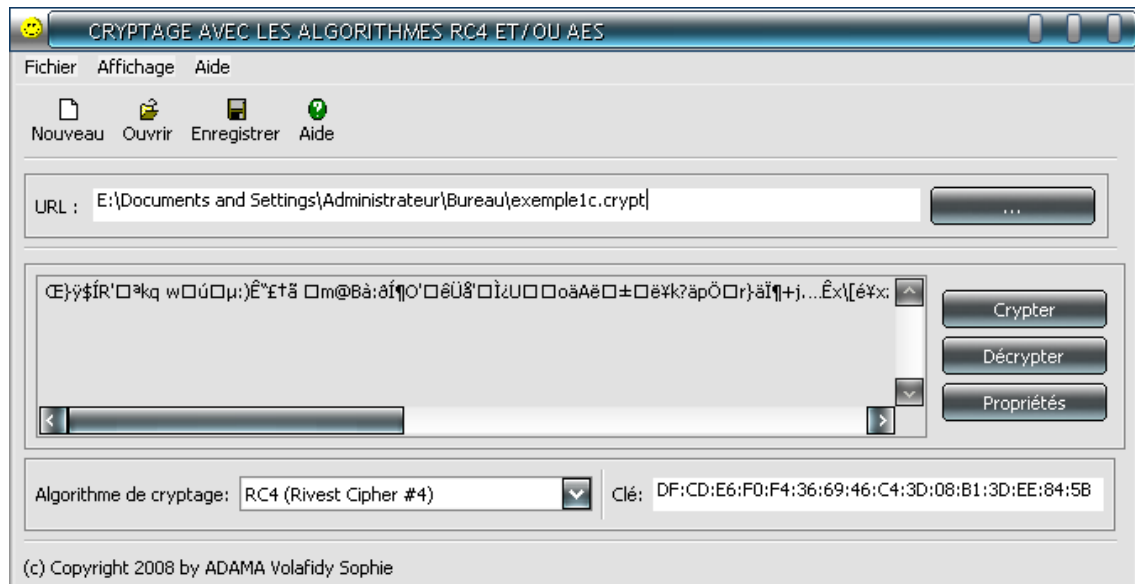


Figure 4.08 : Chiffrement du fichier original

Remarques :

- La clé est affichée en format hexadécimal.
- La même clé de chiffrement est utilisée pour le cryptage et le décryptage.
- La clé RC4 a une longueur de 16 octets soit 128 bits ; 24 de ces bits représentent le vecteur d'initialisation.

#### 4.1.4. Affichage de la structure du paquet

Pour afficher la structure du paquet crypté, il faut cliquer sur le menu Affichage / Structure du paquet.

Nous avons vu dans les chapitres précédents que l'algorithme de chiffrement RC4 est utilisé aussi bien par le WEP que par le WPA. Ceci étant, notre logiciel prend en compte ce fait. Donc, après avoir cliqué sur Structure du paquet, une fenêtre permet à l'utilisateur de choisir le paquet qu'il veut voir s'afficher. Ceci est illustré par la figure 4.09.



Figure 4.09 : Choix du paquet à afficher

#### 4.1.4.1. Paquet (MPDU) WEP

Le MPDU est constitué de l'en-tête MAC (30 octets), d'un champ IV de 6 octets, d'un champ ID de 2 octets, des données cryptées (0 – 2300 octets) et de l'ICV crypté (4 octets).

| Structure du paquet crypté avec RC4 |          |    |                                        |            |  |
|-------------------------------------|----------|----|----------------------------------------|------------|--|
| En-tête MAC                         | IV       | ID | Données cryptées                       | ICV crypté |  |
| -                                   | DF:CD:E6 | 01 | Œ}ÿ\$IR'□%kq□w□ú□μ;)É%t\$ □m@Bà:ðI¶... | ð'YÓP~à□   |  |

Figure 4.10 : Structure du paquet WEP chiffré avec l'algorithme RC4

#### 4.1.4.2. Paquet (MPDU) WPA

Le MPDU est formé par l'en-tête MAC, un champ de 3 octets formé par les 2 derniers octets de l'IV + 1 octet, suivi du champ ID (1 octet) puis des 4 octets formant l'IV. Ensuite viennent les données cryptées (0 – 2300 octets) et enfin de l'ICV crypté (4 octets). Le champ ID est toujours égal à 1.

| Structure du paquet crypté avec RC4 |          |    |             |                                        |            |
|-------------------------------------|----------|----|-------------|----------------------------------------|------------|
| En-tête MAC                         | IV1      | ID | IV2         | Données cryptées                       | ICV crypté |
| -                                   | CD:E6:4F | 01 | DF:CD:E6:4F | Œ}ÿ\$IR'□%kq□w□ú□μ;)É%t\$ □m@Bà:ðI¶... | ð'YÓP~à□   |

Figure 4.11 : Structure du paquet WPA chiffré avec l'algorithme TKIP

#### 4.1.5. Déchiffrement avec l'algorithme RC4

Le déchiffrement s'effectue en appuyant sur le bouton Décrypter ; on retrouve le fichier original (figure 4.12).

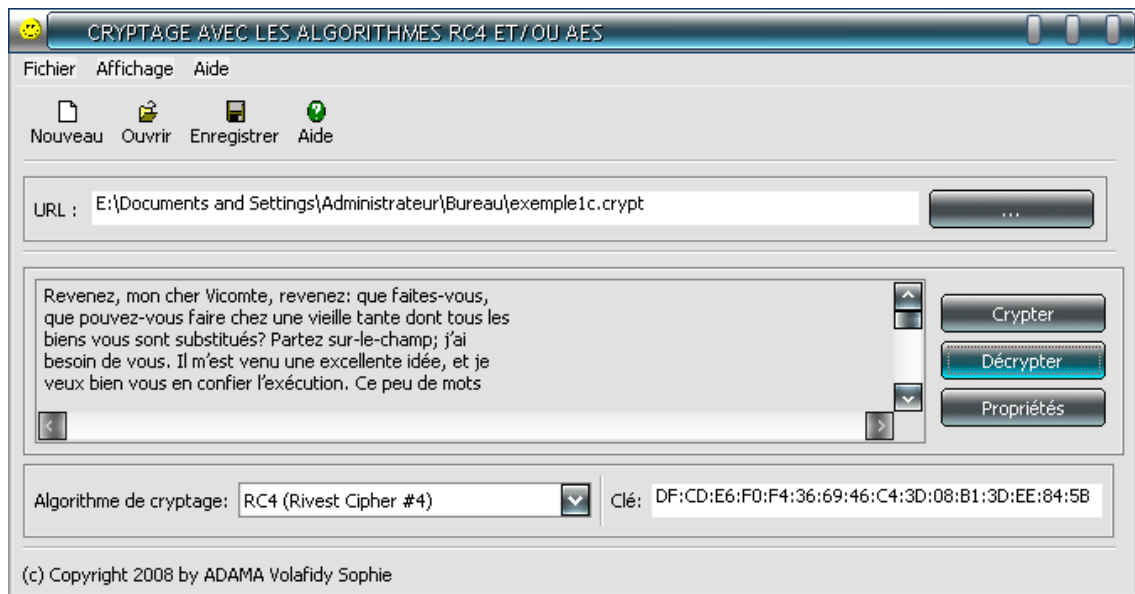


Figure 4.12 : Résultat du déchiffrement avec l'algorithme de chiffrement RC4

#### 4.1.6. Chiffrement/déchiffrement d'un fichier (texte) en utilisant l'algorithme de cryptage AES

Pour le chiffrement/déchiffrement avec l'algorithme AES, le processus est le même que pour RC4 sauf qu'il faut choisir **Algorithme de cryptage : AES (Advanced Encryption Standard)** au lieu de **RC4 (Rivest Cipher #4)**.

Nous allons choisir ici de déchiffrer le fichier *exemple2.txt* (figure 4.13):

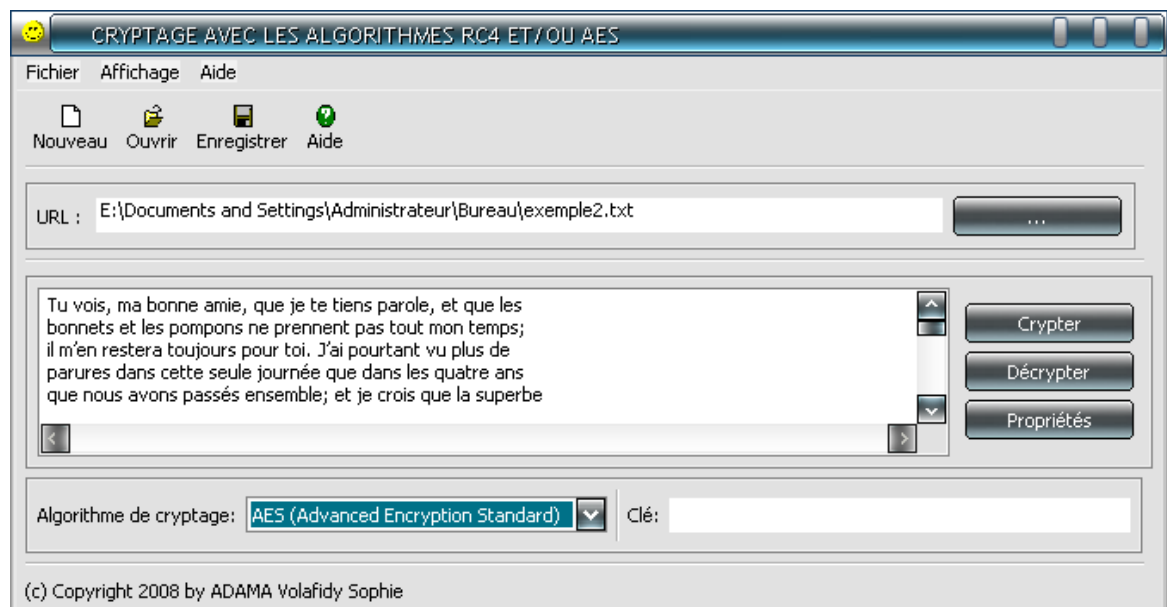


Figure 4.13 : Fichier à chiffrer

Nous savons que l'AES est un algorithme de chiffrement par bloc. Pourtant, le fichier *exemple2.txt* fait 1318 octets ; sa taille n'est donc pas un multiple de 8 octets. Le chiffrement/déchiffrement est rendu possible grâce au mode que nous avons utilisé pour le cryptage ; il s'agit du mode CBC.

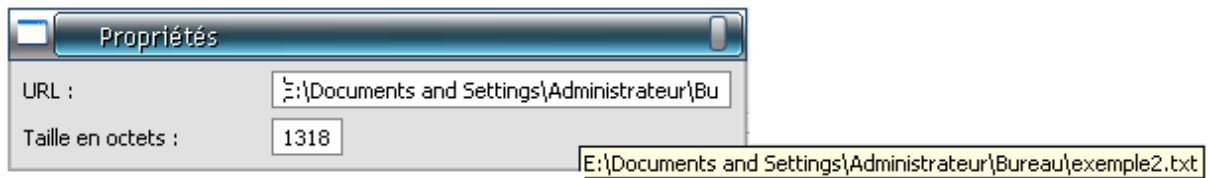


Figure 4.14 : Propriétés du fichier *exemple2.txt*

Un clic sur le bouton Crypter fait apparaître la fenêtre suivante :

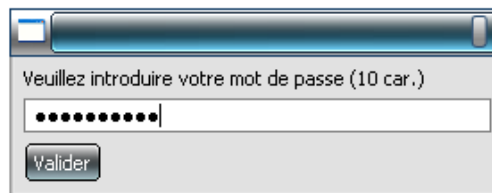


Figure 4.15 : Fenêtre de saisie du mot de passe

Elle permet de saisir le mot de passe qui va servir à chiffrer le fichier. Elle doit être composée de 8 caractères au minimum et de 63 caractères au maximum. Nous avons choisi 10 caractères, et le mot de passe que nous avons utilisé est : « 1234567890 ».

En chiffrant le fichier et après l'avoir enregistré, on obtient (figure 4.16)):

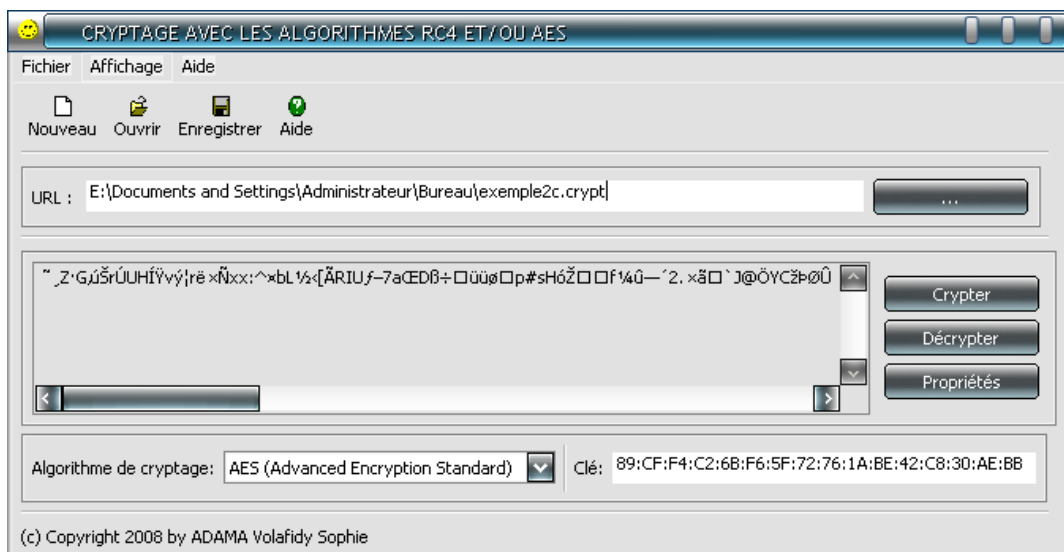


Figure 4.16 : Chiffrement du fichier original *exemple2.txt*

La trame est constituée d'un en-tête MAC (30 octets), d'un en-tête CCMP (8 octets), des données cryptées (0 – 2296 octets), du MIC crypté (8 octets) et du CRC (4 octets).

| Structure du paquet crypté avec AES |              |                                           |               |          |
|-------------------------------------|--------------|-------------------------------------------|---------------|----------|
| En-tête MAC                         | En-tête CCMP | Données cryptées                          | MIC crypté    | CRC      |
| -                                   | 3e8          | ~_Z'GJ5rÚUHIÿvÿ rè xNxx:~*bL½<[ÄRIUf-7... | □w@u,t¥ø□Ä... | ed81809a |

Figure 4.17 : Structure du paquet chiffré avec l'algorithme AES

Le déchiffrement se fait aussi par une saisie de mot de passe. Si ce dernier ne correspond pas à celui saisi à l'émission, le fichier ne pourra pas être déchiffré. On obtient :

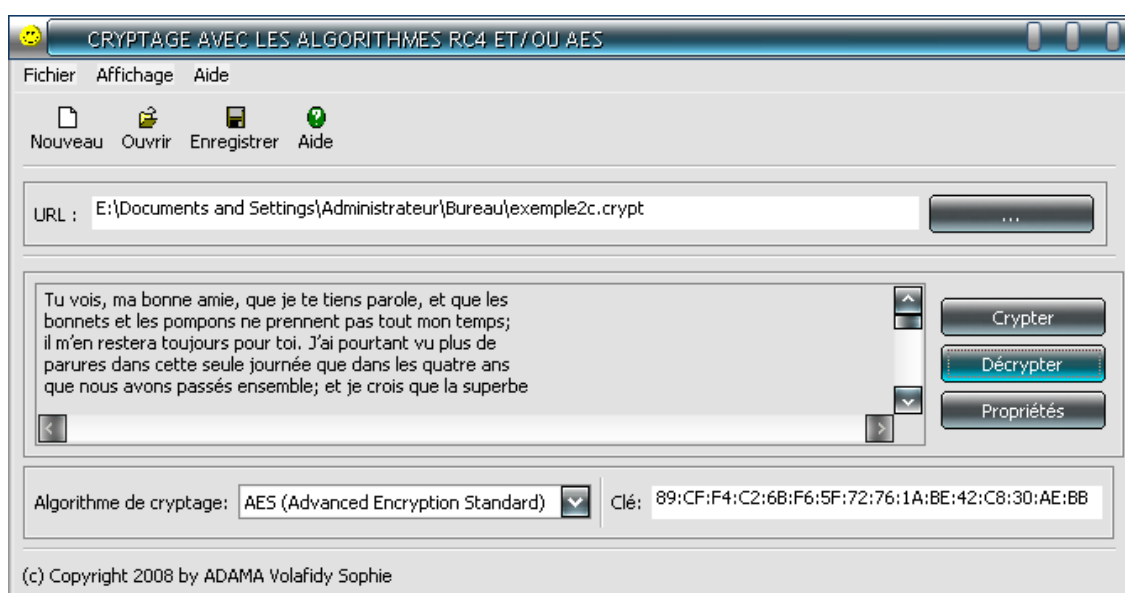


Figure 4.18 : Résultat du déchiffrement avec l'algorithme de chiffrement AES

#### 4.1.7. Quitter le logiciel

Pour sortir du logiciel, il faut cliquer sur le menu Fichier/Quitter ou tout simplement « Ctrl+'Q' ».

## 4.2. Conclusion

Ce logiciel permet de visualiser la structure du paquet WiFi au niveau de la couche MAC. Cette structure est différente selon la solution de sécurité utilisée (WEP ou WPA).

## CONCLUSION GENERALE

Le WiFi (Wireless Fidelity) est un réseau local sans fil à peu près équivalent à un réseau local d'entreprise du point de vue zone de couverture. Cependant, le support de transmission est l'atmosphère. Les ondes se propageant librement dans l'air, des personnes mal intentionnées peuvent en tirer profit et se procurer des informations auxquelles ils n'ont pas accès.

Pour empêcher ceci, diverses solutions de sécurité ont été mises en œuvre ; plus particulièrement l'authentification de l'utilisateur ainsi que le chiffrement de données transmises sur le réseau. La première génération des solutions de sécurité à avoir été intégrée dans le 802.11 est le WEP (*Wired Equivalent Privacy*) qui utilise l'algorithme de chiffrement RC4. Au début, tout fonctionnait à merveille jusqu'à ce que les pirates trouvent le moyen de casser la clé WEP. La faiblesse du WEP résidait surtout dans la longueur de son vecteur d'initialisation qui ne faisait que 24 bits. Désormais, le WEP ne constitue plus une protection efficace contre les malfaiteurs. Une seconde solution a donc vu le jour, le WPA (*WiFi Protected Access*) qui utilise le TKIP (*Temporary Key Integrity Protocol*). Ce dernier ne possède pas les faiblesses du WEP ; son vecteur d'initialisation est de 48 bits, ce qui évite la répétition des clés. Cependant, la première version du WPA ne permet de sécuriser que les réseaux de type Infrastructure. WPA mais n'est qu'une solution temporaire en attendant la ratification de la norme 802.11i (WPA2). Le WPA2, quant à lui, est identique au WPA mais permet aussi de protéger tous les types de réseau.

Il est donc important de protéger son réseau contre les accès non autorisés. La question de sécurité est très évolutive. Ceci étant, il faut se mettre à jour afin de connaître quelles sont les possibilités d'attaques et de se protéger en conséquence.

Les solutions de sécurité énoncées précédemment ne sont pas totalement efficaces contre les pirates ; il est peut être préférable d'ajouter à ces dernières d'autres méthodes telles que la biométrie qui est d'actualité de nos jours (empreinte digitale ou rétinienne, ...) pour l'authentification des utilisateurs.

## ANNEXE 1 : PRESENTATION DU LOGICIEL ECLIPSE

### 1. Présentation du logiciel Eclipse [22]

Eclipse est un environnement de développement intégré (*Integrated Development Environment*) dont le but est de fournir une plateforme modulaire pour permettre de développer des développements informatiques.

I.B.M. est à l'origine du développement d'Eclipse qui est d'ailleurs toujours le cœur de son outil *Websphere Studio Workbench* (WSW), lui même à la base de la famille des derniers outils de développement en Java d'I.B.M. Tout le code d'Eclipse a été donné à la communauté par I.B.M afin de poursuivre son développement.

Eclipse utilise énormément le concept de modules nommés "plug-ins" dans son architecture. D'ailleurs, hormis le noyau de la plate-forme nommé "Runtime", tout le reste de la plate-forme est développé sous la forme de plug-ins. Ce concept permet de fournir un mécanisme pour l'extension de la plate-forme et ainsi fournir la possibilité à des tiers de développer des fonctionnalités qui ne sont pas fournies en standard par Eclipse.

Les principaux modules fournis en standard avec Eclipse concernent Java mais des modules sont en cours de développement pour d'autres langages notamment C++, Cobol, mais aussi pour d'autres aspects du développement (base de données, conception avec UML, ...). Ils sont tous développés en Java soit par le projet Eclipse soit par des tiers commerciaux ou en open source.

Les modules agissent sur des fichiers qui sont inclus dans le workspace. Le workspace regroupe les projets qui contiennent une arborescence de fichiers.

Bien que développé en Java, les performances à l'exécution d'Eclipse sont très bonnes car il n'utilise pas Swing pour l'interface homme-machine mais un toolkit particulier nommé SWT associé à la bibliothèque JFace. SWT (*Standard Widget Toolkit*) est développé en Java par IBM en utilisant au maximum les composants natifs fournis par le système d'exploitation sous jacent. JFace utilise SWT et propose une API pour faciliter le développement d'interfaces graphiques.

Eclipse ne peut donc fonctionner que sur les plateformes pour lesquelles SWT a été porté. Ainsi, Eclipse 1.0 fonctionne uniquement sur les plateformes Windows 98/NT/2000/XP et Linux.

SWT et JFace sont utilisés par Eclipse pour développer le Workbench qui organise la structure de la plate-forme et les interactions entre les outils et l'utilisateur. Cette structure repose sur trois concepts : la perspective, la vue et l'éditeur. La perspective regroupe des vues et des éditeurs pour offrir une vision particulière des développements. En standard, Eclipse propose huit perspectives.

Les vues permettent de visualiser et de sélectionner des éléments. Les éditeurs permettent de visualiser et de modifier le contenu d'un élément du workspace.

## **2. Le langage de programmation Java [22] [23]**

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche de C. Il existe deux types de programmes en Java : les applets et les applications. Une application autonome (*stand alone program*) est une application qui s'exécute sous le contrôle direct du système d'exploitation. Une applet est une application qui est chargée par un navigateur et qui est exécutée sous le contrôle du plug in de ce dernier.

### **2.1. JCE (Java Cryptography Extension)**

JCE fournit un framework et des implémentations pour le chiffrement, la génération de clés et les clés partagées, et les algorithmes MAC (*Message Authentication Code*). Les supports pour le chiffrement incluent les chiffrements symétriques, asymétriques, par bloc et par flux.

## ANNEXE 2: CODE SOURCE

```
package pack2;

import java.util.Arrays;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

/** class RC4encryptor */
public final class RC4encryptor
{
    private static final String KEY_ALGORITHM = "RC4";
    private static final String CIPHER_ALGORITHM = "RC4";

    /** constructeur */
    private RC4encryptor(){}

    public static byte[] encrypt(final byte[] cleartext) throws Exception
    {
        /* initialization de la clé de 16 octets */
        final byte[] key = new byte[16];

        /* cryptage RC4-128 */
        final Cipher cipher = Cipher.getInstance(CIPHER_ALGORITHM);

        cipher.init(
            Cipher.ENCRYPT_MODE,
            new SecretKeySpec(key, KEY_ALGORITHM));
        Arrays.fill(key, (byte) 0x00);
        return cipher.doFinal(cleartext);
    }

    public static byte[] decrypt(final byte[] ciphertext) throws Exception
    {
        final byte[] key = new byte[16];

        /* decryptage RC4-128 */
        final Cipher cipher = Cipher.getInstance(CIPHER_ALGORITHM);

        cipher.init(
            Cipher.DECRYPT_MODE,
            new SecretKeySpec(key, KEY_ALGORITHM));
        Arrays.fill(key, (byte) 0x00);
        return cipher.doFinal(ciphertext);
    }
}

/** class AESencryptor */
public final class AESencryptor
{
    private static final String CHARSET_NAME = "UTF-8";
    private static final String RNG_ALGORITHM = "SHA1PRNG";
    private static final String DIGEST_ALGORITHM = "SHA-256";

    private static final String KEY_ALGORITHM = "AES";
    private static final String CIPHER_ALGORITHM = "AES/CBC/PKCS5Padding";

    /** constructeur */
    private AESencryptor(){}
    public static byte[] encrypt(
        final String password,
        final byte[] cleartext)
        throws Exception
    {
        /* calcule la clé et le vecteur d'initialisation */
        final MessageDigest shaDigest = MessageDigest.getInstance(DIGEST_ALGORITHM);
        byte[] pw = password.getBytes(CHARSET_NAME);
```

```

    for (int i = 0; i < iterations; i++)
    {
        /* ajout du salt */
        final byte[] salted = new byte[pw.length + salt.length];
        System.arraycopy(pw, 0, salted, 0, pw.length);
        System.arraycopy(salt, 0, salted, pw.length, salt.length);
        Arrays.fill(pw, (byte) 0x00);
    }

    /* extraction de la clé et du vecteur d'initialisation */
    final byte[] key = new byte[16];
    final byte[] iv = new byte[16];
    System.arraycopy(pw, 0, key, 0, 16);
    System.arraycopy(pw, 16, iv, 0, 16);
    Arrays.fill(pw, (byte) 0x00);

    /* chiffrement AES-128 */
    final Cipher cipher = Cipher.getInstance(CIPHER_ALGORITHM);

    cipher.init(
        Cipher.ENCRYPT_MODE,
        new SecretKeySpec(key, KEY_ALGORITHM));

    Arrays.fill(key, (byte) 0x00);
    return cipher.doFinal(cleartext);
}

public static byte[] decrypt(
    final String password,
    final byte[] ciphertext)
    throws Exception
{
    /* calcule la clé et le vecteur d'initialisation */
    final MessageDigest shaDigest = MessageDigest.getInstance(DIGEST_ALGORITHM);
    byte[] pw = password.getBytes(CHARSET_NAME);

    for (int i = 0; i < iterations; i++)
    {
        /* ajout du salt */
        final byte[] salted = new byte[pw.length + salt.length];
        System.arraycopy(pw, 0, salted, 0, pw.length);
        System.arraycopy(salt, 0, salted, pw.length, salt.length);
        Arrays.fill(pw, (byte) 0x00);
    }

    /* extraction de la clé et du vecteur d'initialisation */
    final byte[] key = new byte[16];
    final byte[] iv = new byte[16];
    System.arraycopy(pw, 0, key, 0, 16);
    System.arraycopy(pw, 16, iv, 0, 16);
    Arrays.fill(pw, (byte) 0x00);

    /* dechiffrement AES-128 */
    final Cipher cipher = Cipher.getInstance(CIPHER_ALGORITHM);

    cipher.init(
        Cipher.DECRYPT_MODE,
        new SecretKeySpec(key, KEY_ALGORITHM),
        new IvParameterSpec(iv));

    Arrays.fill(key, (byte) 0x00);
    Arrays.fill(iv, (byte) 0x00);

    return cipher.doFinal(ciphertext);
}
}

import java.io.*;
import java.security.Key;

import org.eclipse.swt.widgets.*;

```

```

import org.eclipse.swt.events.*;
import org.eclipse.swt.graphics.*;
import org.eclipse.swt.layout.*;
import org.eclipse.swt.SWT;

import javax.crypto.*;

public class Fenetre {

    /** Déclaration des variables utilisées */
    ...

    private Shell shell;

    public static void centrerSurEcran(Display display, Shell shell) {
        Rectangle rect = display.getClientArea();
        Point size = shell.getSize();
        int x = (rect.width-size.x) / 2;
        int y = (rect.height-size.y) / 2;
        shell.setLocation(new Point(x, y));
    }

    /** constructeur */
    public Fenetre() {

        final Display display = new Display();
        shell = new Shell(display);
        Image img1 = new Image(
            display,
            "D:\\Eclipse_Workspace\\Projet1\\pack2\\icon.ICO");
        shell.setImage(img1);

        shell.setText("CRYPTAGE AVEC LES ALGORITHMES RC4 ET/OU AES");
        centrerSurEcran (display, shell);
        shell.setLayout(new GridLayout());

        /** Mise en forme de la barre des tâches */
        final ToolBar toolbar = new ToolBar (shell, SWT.HORIZONTAL);
        ...

        /** Ajout des les composantes de la fenêtre */
        ...

        /** Mise en forme de la barre des menus */
        Menu menu = new Menu (shell, SWT.BAR);
        /** Menu Fichier */
            /** Sous-menu Fichier */ ...
            /** Sous-menu Ouvrir */ ...
            /** Sous-menu Enregistrer le fichier */ ...
            /** Sous-menu Enregistrer le fichier crypté */ ...
            /** Sous-menu Quitter */ ...
        /** Menu Affichage */
            /** Sous-menu Structure du paquet */ ...
            /** Sous-menu Propriétés */ ...
        /** Menu Aide */
            /** Sous-menu Aide */ ...
            /** Sous-menu A propos de ... */ ...
        shell.setMenuBar(menu);
        /** Fin de la mise en forme de la barre des menus */

        shell.pack();
        shell.open();

        while (!shell.isDisposed()) {
            if (!display.readAndDispatch())
                display.sleep();
        }
        display.dispose();
    }

    /** Calcul du CRC32 */

```

```

    private String calculCRC (byte [] bytes) {
        java.util.zip.CRC32 x = new java.util.zip.CRC32();
        x.update(bytes);
        String valCRC = Long.toHexString(x.getValue());
        return valCRC;
    }

    /** Calcul du MIC */
    private String calculMIC (byte [] bytes) {
        ...
    }

/** Les différentes méthodes utilisées */

    /** Nouveau fichier */
    private void newFile() {
        ...
    }

    /** Ouverture de fichier */
    private void openFile(String ext) {
        ...
    }

    /** Enregistrement de fichier */
    private void saveFile(String ext) {
        ...
    }

    /** Affichage des propriétés du fichier chargé */
    private void afficherProprietes() {
        ...
    }

    /** Affichage de A propos de ... */
    private void afficherApropos() {
        ...
    }

/** Affichage de l'aide */
private void afficherAide() {
    ...
}

/** Affichage d'un paquet crypté avec RC4 */
private void afficherPaquetRC4() {
    ...
}

/** Affichage d'un paquet crypté avec AES */
private void afficherPaquetAES() {
    ...
}
}

```

## BIBLIOGRAPHIE

- [1] <http://www.lex-electronica.org/articles/v11-1/mian.pdf>
- [2] J. F. Pillou, *CommentCaMarche.net* .
- [3] A. Géron, *WiFi Déploiement et Sécurité, La QoS et le WPA*, Deuxième édition, Dunod : Paris, 2004, 2006.
- [4] F. Di Gallo, *Essentiel WiFi*, 2003.
- [5] M. Terré, *WiFi : le standard 802.11 – Couche physique et couche MAC*, 2007.
- [6] D. Lalot, *Réseaux informatiques*, <http://ciscam.univ-aix.fr/doctech/reseaux.pdf>
- [7] Y. Lescop, *Les supports de transmission*, 2002.
- [8] B. T. Madiou, R. Gambone, *Les divers moyens de transmission d'une information dans un réseau*, Nov. 2004.
- [9] M. Duchateau, *Analyse et Simulation du déploiement des réseaux sans fil à l'ULB*, Belgique 2002.
- [10] Cours Sécurité WiFi, The Hackademy School.
- [11] IEEE 802.11
- [12] S. Mazen, *Les réseaux de communication sans fil*.
- [13] P. O. Bourgeois, A. Marcou, *La sécurité dans IEEE 802.11*, Université de Nantes, 2004.
- [14] J. D. Gibson, *The Communications Handbook, Second Edition*, pp. 1242-1253, 1357-1374, 2002.
- [15] P. Latu, « Technologie des réseaux sans fil IEEE 802.11 », [www.linux-france.org/prj/inetdoc/](http://www.linux-france.org/prj/inetdoc/), Fev. 2007.
- [16] *Pratique des solutions VPN sur les réseaux WiFi*, Livre Blanc, première édition, déc. 2005.
- [17] S. Convery, D. Miller et Sri Sundaralingam, « Description détaillée de la sécurité pour les réseaux locaux sans fil », *Livre Blanc - Cisco Safe*, 2003.
- [18] G. Lehembre, « Sécurité WiFi – WEP, WPA, WPA2 », [www.hackin9.org](http://www.hackin9.org), hackin9 N°1, pp. 12 - 26, 2006.

- [19] V. Alvaro, S. Dupuy, *Recommandations d'utilisation des réseaux sans fil*, Centre de calcul Recherche et Réseau Jussieu : 2003.
- [20] S. Haykin, *Communication systems*, 4th edition, Appendix 5, John Wiley & Sons, Inc., 2001.
- [21] J. Stern, L. Granboulan, P.Nguyen, D. PointCheval, *Conception et preuves d'algorithmes cryptographiques*, Edition 2004.
- [22] M. Minier, *Cryptographie : de la théorie à la pratique*, Séminaire 5IF, Novembre 2005.
- [23] G. Florin, S. Natkin, *Les techniques de cryptographie*, Mars 2002.
- [24] J. M. Doudoux, *Développons en Java*, version 0.80.1
- [25] J. M. Doudoux, *Développons en Java avec Eclipse*, version 0.40
- [26] Java Cryptography Extension (JCE) Reference Guide for the Java Platform Standard Edition Development Guide (JDK) 5.0
- [27] 802.11n and WiFi telephony, White Paper, March 2008.
- [28] H. Benchikh, G. Cognard, La norme 802.11n, Décembre 2006.

**PAGE « RENSEIGNEMENTS »**

Nom : ADAMA  
Prénoms : Volafidy Sophie  
Adresse : B.P. 108, 201 – Antsiranana  
Tél : 032 40 853 51  
Courriel : [avsophie@hotmail.com](mailto:avsophie@hotmail.com)

Titre du mémoire : LE CHIFFREMENT DES DONNEES DANS LES RESEAUX WIFI

Nombre de pages : 84

Nombre de tableaux : 3

Nombre de figures : 69

Mots clés : wifi, wpa, wep, réseau, sécurité, cryptographie, chiffrement, déchiffrement, rc4, aes.

Directeur de mémoire : M. RANDRIAMITANTSOA Paul Auguste

## RESUME

Ce travail de mémoire étudie les différents types de chiffrement mis en œuvre pour sécuriser les réseaux locaux sans fil Wifi. Après avoir présenté leurs bases théoriques, nous avons élaboré un petit logiciel qui met en œuvre ces dites bases en permettant de chiffrer et de déchiffrer un fichier avec l'un ou l'autre de ces différents algorithmes, à savoir l'algorithme de chiffrement par flots RC4 et l'algorithme de chiffrement par bloc AES, de visualiser le résultat (le texte crypté) ainsi que la structure du paquet ainsi chiffré.

## ABSTRACT

This work studies the different types of encryption used to safe a wireless local area network (Wi-Fi). After having presented their theoretical bases, we have made a little software which makes use of these said bases by allowing encryption and decryption of a file with one of these algorithms which are the stream cipher algorithm RC4 and the block cipher algorithm AES, then to watch the result (the ciphertext) and the encrypted packet structure.

