REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

MEMOIRE DE MAGISTER EN ELECTRONIQUE

OPTION : ***Télédétection***

Présenté par :

M. BOUKHATEM Mohammed Belkaid

THEME :

Application des techniques de cryptage pour la transmission sécurisée d'images MSG

Devant les membres du jury :

Président :	Mme. AMEUR	Zohra	Professeur	UMMTO
Rapporteur :	Mr. LAHDIR	Mourad	Maître de conférences A	UMMTO
Examinateurs :	Mr. ZIANI	Rezki	Professeur	UMMTO
	Mr. HAMICHE	Hamid	Maître de conférences A	UMMTO

Soutenu le : 11 / 03 / 2015

REMERCIEMENT

Ce mémoire a été réalisé au sein du laboratoire d'Analyse et Modélisation des Phénomènes Aléatoires (LAMPA) de la Faculté de Génie Electrique et Informatique de l'Université Mouloud MAMMERI de Tizi-Ouzou.

Je souhaite adresser ici tous mes remerciements à mon encadreur Mr LAHDIR Mourad, Maitre de Conférences à l'UMMTO, pour l'aide et le temps qu'il a bien voulu me consacrer et sans qui ce mémoire n'aurait jamais vu le jour.

Je remercie vivement Madame AMEUR Zohra, Professeur à l'UMMTO et Directrice du laboratoire LAMPA, d'avoir accepté de présider le jury.

J'exprime mes plus sincères remerciements à Monsieur ZIANI Rezki, Professeur à l'UMMTO d'avoir accepté de faire partie d jury.

Mes remerciements vont aussi à Monsieur HAMICHE Hamid, Maitre de Conférences A à l'UMMTO, pour l'intérêt qu'il a porté à ce travail et d'accepter de faire partie du jury.

J'exprime mes gratitudes à tous les enseignants qui n'ont pas ménagé leurs efforts pour nous assurer une bonne formation. Un grand merci aussi, aux membres du laboratoire LAMPA ainsi à ceux qui assurent son bon fonctionnement.

Laboratoire d'Analyse et Modélisation des Phénomènes Aléatoires, UMMTO, BP 17 RP, 15000, Tizi-Ouzou, Algérie

DEDICACE

Merci au Noble « Allah» Dieu le tout puissant qui m'a donné le courage, l'intelligence, la force et la patience pour réaliser ce travail.

À celle qui m'a indiqué la bonne voie en me rappelant que la volonté fait toujours les grands hommes…

Merci ma Mère

À celui qui a attendu avec patience les fruits de sa bonne éducation…

Merci mon Père.

J'exprime ma gratitude à tous mes frères et à toutes mes sœurs.

J'adresse également mes plus sincères remerciements à tous mes proches et amis qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire

Laboratoire d'Analyse et Modélisation des Phénomènes Aléatoires, UMMTO, BP 17 RP, 15000, Tizi-Ouzou, Algérie

RESUME

***Abstract*—** In this work, we propose a new encryption system for secure transmission of meteorological images from Meteosat Second Generation (MSG). The hybrid encryption scheme is based on AES and RSA algorithms to validate the three security services are authentication, integrity and confidentiality. Data protection is ensured by AES and authenticity is guaranteed by the RSA algorithm. Integrity is assured by the basic function of the correlation between adjacent pixels. Our encryption system generates a unique password every 15 minutes that will be used to encrypt each frame of the image database MSG. Several parameters were used for various tests of our analysis. For the integrity test, we noticed the effectiveness of our system and how changes cryptographic printing at reception if a change affects the image in the transmission channel.

***Résumé*—** Les travaux de recherche de ce mémoire s'inscrivent dans le cadre de la sécurité des images satellitaires, par les deux algorithmes AES et RSA, en particulier les images de satellite météorologique MSG. L'originalité de ce mémoire consiste à proposer un cryptosystème à base deux algorithmes AES et RSA pour assurer les trois services de la sécurité l'authentification, l'intégrité et la confidentialité que l'algorithme AES seul, ne peut assure que la fonction de confidentialité. Dans notre cryptosystème La confidentialité est assurée par l'algorithme AES, l'authenticité est assurée par l'algorithme RSA. Et l'intégrité est assurée à la base de la fonction de corrélation entre les pixels adjacents. Notre cryptosystème génère un mot de passe unique de session tout les 15 minutes utilise pour chiffrer chaque image pour renforcer et garantir la sécurité. Pour l'analyse de sécurité et les résultats des tests, nous avons appliqué plusieurs métriques de test, pour le test d'intégrité nous avons remarqué l'efficacité de notre système et comment l'empreinte cryptographique change à la réception si une modification touche l'image dans le canal de transmission.

Mots-clés : Cryptages symétrique et asymétrique, transfert sécurisé, Image cryptée, Meteosat Second Generation (MSG), sécurité, transmission de données, chiffrement, clés, algorithme DES, AES, RSA, réseau, authentification.

Laboratoire d'Analyse et Modélisation des Phénomènes Aléatoires, UMMTO, BP 17 RP, 15000, Tizi-Ouzou, Algérie

SOMMAIRE

Laboratoire d'Analyse et Modélisation des Phénomènes Aléatoires, UMMTO, BP 17 RP, 15000, Tizi-Ouzou, Algérie

SOMMAIRE

INTRODUCTION

Tout au long de l'histoire, l'humanité a essayé d'envoyer des informations d'une façon sécurisée. Le chiffrement d'information a été utilisé comme instrument de sécurisation pour des stratégies militaires et des échanges de données secrètes. Le transfert sécurisé d'information est nécessaire et énormément utilisée dans le monde numérique. Les réseaux numériques ont fortement évolué ces dernières années et sont devenus inévitables pour la communication moderne. Les images MSG (Météo satellite Second Génération) [1] [2] transmises sur ces réseaux sont des données particulières du fait de leur quantité importante d'information. La transmission des images MSG soulève donc un nombre important de problèmes. Nous citons, par exemple la confidentialité, l'authentification et l'intégrité des données :

- Toute information circulant peut être capturée et lue « Sniffing », La confidentialité se base sur les concepts qui permettent de s'assurer que l'information ne puisse pas être lue par des personnes non autorisées. La confidentialité est fortement liée à la cryptographie.
- Une personne peut falsifier ses informations numériques personnelles « Spoofing », l'authentification est l'ensemble des moyens qui permettent d'assurer que les données reçues et envoyées proviennent bien des entités déclarées.
- Les données peuvent être capturées et modifiées, l'intégrité des données concerne les techniques qui rendent possible la vérification de la non-altération des données, c'est-à-dire le contrôle du contenu.

Plusieurs techniques de chiffrement sont proposées pour répondre à ces exigences; Nous trouvons parmi elles, l'algorithme public symétrique AES (Advanced Encryption Standard) [3] [4] [5] [6,7] [8] qui a prouvé de nos jours sa robustesse contre les différents types d'attaques, l'algorithme asymétrique RSA (Rivest, Shamir and Adleman) [4] [9] [10], et l'algorithme IDEA (International Data Encryption Algorithm)[11]. Mais si nous appliquons ces algorithmes séparément pour la transmission des images MSG, ils ne peuvent assurer que la confidentialité, pour cette raison nous proposons un cryptosystème de chiffrement hybride basé sur les deux algorithmes AES et RSA pour assurer les trois grands axes de sécurité : la confidentialité, l'authenticité, l'intégrité.

Pour assurer la confidentialité dans notre cryptosystème notre choix est tombé sur l'algorithme AES avec ses cinq modes de fonctionnement [4] [11] [12], car il consomme peu de mémoire et n'étant pas basé sur un schéma de Feistel. Sa complexité est moindre et est plus facile à implémenter. Il est très rapide et n'est pas cassé jusqu'à nos jours. Et la principale raison est que le satellite MSG utilise l'algorithme 3-DES pour chiffrer ces images, alors nous utilisons dans notre cryptosystème AES qui est le successeur de DES [10]. Nous analysons les résultats de chiffrement, et l'RSA pour assurer l'authentification et les échanges sécurisés des clés, car il très rapide, et très sure avec l'utilisation de clef de chiffrement supérieur à 2048 bits.

Le cryptosystème inclut aussi une procédure basée sur la corrélation entre les pixels voisins de l'image pour assurer l'intégrité. Cette propriété va nous conduire à une entité ou

empreinte cryptographique unique, utilisée comme clé de chiffrement générée toutes les 15 minutes.

Ce mémoire s'articule autour de quatre chapitres :

Le premier chapitre représente l'état de l'art de la cryptographie : nous aborderons plusieurs types de chiffrements et les différents types d'attaques.

Le deuxième chapitre se compose de deux parties : La première partie traite en détaille l'Algorithme symétrique AES, la deuxième est consacrée pour l'étude détaillée de l'algorithme asymétrique RSA.

Le troisième chapitre est composé de trois parties : la première traite des images MSG, la deuxième explique en détail les cinq modes de fonctionnement pour la transmission des images MSG, et la dernière partie détaille le cryptosystème proposé.

Dans Le dernier chapitre, nous examinerons la robustesse de notre cryptosystème de chiffrement hybride à l'aide d'un ensemble de métriques d'évaluation du degré de cryptage.

Nous finirons notre travail par une conclusion et un ensemble des perspectives ouvertes par ce thème de recherche.

Chapitre I

Généralités sur la cryptographie

I.1. Préambule

La cryptographie est une science très ancienne qui date de 1900 ans avant J.-C. Des recherches indiquent qu'un scribe égyptien a employé des hiéroglyphes non conformes à la langue pour écrire un message. De ce temps-là et au long de l'histoire, la cryptographie a été utilisée exclusivement à des fins militaires. Aujourd'hui, les réseaux informatiques exigent une phase de cryptographie comme mécanisme fondamental afin d'assurer la confidentialité de l'information numérique. En se rapportant à des événements historiques de notre ère informatisée, nous pouvons retenir les suivants : la machine Énigma créée par Dr Arthur Scherbius en 1923 et qui a été utilisée largement dans la seconde guerre mondiale ; la théorie de l'information de Claude Shannon en 1949; l'algorithme lucifer développé par IBM en 1970; la théorie du système à clef publique par Whitfield Diffie et Martin Hellman en 1976 qui a marqué le début de la cryptographie moderne ; le premier standard pour le cryptage, l'algorithme DES en 1976; les résultats de la cryptographie quantique par Charles H. Bennett et Gilles Brassard en 1990, et le standard AES en 2000, qui a remplacé le DES.

Dans ce chapitre, nous énumérons certains termes et terminologies qui faciliteront la compréhension des concepts et des objectifs des travaux de recherche développés dans les sections suivantes.

I.2. Généralités sur le cryptage

Le cryptage peut être défini comme une fonction réversible de transformation des données en envisageant la protection d'information contre toute prise de connaissance du contenu (confidentialité).

un cryptosystème est caractérisé par l'ensemble des éléments (P,C,K,E,D), Où P est l' ensemble des textes clairs possibles, C l'ensemble des textes chiffrés possibles et K l'ensemble des clefs possibles.

Pour chaque clef $k \in K$, il y a une règle de cryptage $ek \in E$ et une règle correspondante de décryptage $dk \in D$.

Chaque $ek : P \rightarrow C$ et $dk : C \rightarrow P$ sont des fonctions telles que $dk(ek(x)) = x$ pour tout texte clair $x \in P$.

I.2.1. Cryptosystèmes classiques

Les cryptosystèmes classiques ont été conçus avant la création des ordinateurs et qui ont donné les concepts et les bases pour l'évolution de plusieurs algorithmes symétriques encore utilisés de nos jours. Les cryptosystèmes classiques sont groupés en chiffrement mono alphabétique et poly alphabétique. Le chiffrement mono alphabétique est très primaire, il s'agit d'une substitution simple. Chaque lettre est remplacée par une autre lettre ou symbole conformément à un certain algorithme [10].

I.2.1.1. Chiffrement par décalage (shift cipher)

Le principe de ce chiffrement est de décaler les lettres de l'alphabet. Les fonctions de Cryptage :

$$ek(x) = x + K \bmod 26 \qquad I.1$$

Et décryptage :

$$dk(y) = y - K \bmod 26 \qquad I.2$$

Sont définies en Z_{26} et K est la clef de décalage [10].

Par exemple, pour le texte clair X= ”APPELLE”, en remplaçant chaque lettre par celle située 3 cases plus loin (K= 3), le texte chiffré ”DSSHOOH” est obtenu.

I.2.1.2. Alphabets désordonnés ou codage par substitution

Ce genre de cryptosystème consiste à remplacer une lettre par une autre en utilisant un alphabet désordonné où K est l'ensemble des permutations aléatoires possibles des 26 lettres. Le chiffrement par décalage est un cas particulier du chiffrement par alphabets désordonnés, où il existe seulement 26 clefs.

I.2.1.3. Chiffrement affine

Il s'agit d'un algorithme qui utilise les fonctions affines pour le cryptage :

$$e_k(x) = ax + b \bmod 26 \qquad I.3$$

et pour le décryptage :

$$d_k(y) = a^{-1}(y - b) \bmod 26 \qquad I.4$$

La constante a est premier avec 26, et a^{-1} désigne l'inverse de a modulo 26. Nous remarquons également que si a= 1, nous avons le chiffrement par décalage.

I.2.1.4. Chiffrement de Vigenère

Le chiffrement de Vigenère utilise une clef qui définit le décalage pour chaque lettre du texte clair [8]. Sa force réside dans l'utilisation non pas de 1, mais de 26 alphabets décalés$(Z_{26})^m$ où m est un nombre entier positif.
La fonction de cryptage est :

$$ek(x_1, \cdots, x_m) = (x_1 + k_1, \cdots, x_m + k_m) \qquad I.5$$

et la fonction de décryptage est :

$d_k(y_1, \cdots, y_m) = (y_1 - k_1, \cdots, y_m - k_m)$ sont faits avec les clefs $K = (k_1, \cdots, k_m)$.

Le chiffrement de Vigenère est considéré comme un chiffrement par flot, avec une période m.

I.2.1.5. Chiffrement de Hill

Le chiffrement de Lester S. Hill est considéré polygamique [10]. Dans un chiffrement polygamique, un groupe de n lettres est chiffré par un groupe de n symboles.

Le principe est de remplacer les lettres par leur rang dans l'alphabet, soit K une matrice inversible, alors :

$$e_k(x) = xK \text{ et } d_K(y) = yK^{-1} \quad \text{I.6}$$

En fait, l'idée est de prendre m combinaisons linéaires de m caractères de l'alphabet.

I.2.1.6. Chiffrement par permutation

Le principe du chiffrement par permutation est de garder le même alphabet et de changer seulement l'ordre des lettres. Soit π toutes les permutations $\{1, \ldots, m\}$

Pour une clef et m un entier positif, alors les fonctions de cryptage et décryptage sont décrites par :

$$e_\pi(x_1, \ldots, x_m) = (x_{\pi(1)}, \ldots, x_{\pi(m)}) \quad \text{I.7}$$

$$d\pi(y_1, \ldots, y_m) = (y_{\pi-1}(1), \ldots, y_{\pi-1}(m)) \quad \text{I.8}$$

Le chiffrement par permutation est un cas spécial du chiffrement de Hill pour une matrice de permutation $K_\pi = (k_{i,j})$.

I.2.2. Classification des algorithmes

Les cryptosystèmes peuvent être classé conformément à différentes caractéristiques :

- selon les types de clefs : symétrique, asymétrique ou hybride ;
- selon les techniques de chiffrement : par bloc ou par flot ;
- ou selon les corrélations entre le flux de clefs (key, stream) et les textes clairs et chiffrés : synchrone ou asynchrone.

Le chiffrement symétrique : l'émetteur et le destinataire partagent une clef unique.

Le chiffrement asymétrique : également appelé chiffrement à clef publique est un système où chaque interlocuteur dispose d'un couple de clefs, la clef publique pour crypter et la clef privée pour décrypter.

Le chiffrement hybride fait appel aux deux techniques en même temps, symétrique et asymétrique.

Le chiffrement par flot ou par flux traite les caractères comme une suite de bits, Ils sont considérés comme synchrones, si le flux de clefs est produit indépendamment du texte clair et du texte chiffré. Si le flot de chiffrement est produit à partir de la clef et d'un nombre fixe de caractères du flot chiffré, alors le chiffrement par flot est dit asynchrone.

Le chiffrement par bloc est le type de cryptosystème dont le texte clair est découpé en blocs de tailles fixes et chaque bloc est crypté séparément. Un cryptosystème par bloc peut être symétrique AES ou asymétrique et peut avoir aussi divers modes d'opérations.

I.2.3. Notion de sécurité

Il est possible de classer les cryptosystèmes en trois catégories par rapport à la sécurité : parfaite, calculatoire et sémantique.

I.2.3.1. Sécurité parfaite

La notion de sécurité parfaite ou inconditionnelle a été créée par Shannon. Un cryptosystème est à sécurité parfaite si aucune information ne peut être obtenue sur le texte clair, à partir du texte crypté correspondant.

En réalité, cette approche est impraticable dans la plupart des scénarios. Diffie et Hellman [16] ont proposé le remplacement de la sécurité parfaite par le concept de sécurité calculatoire pour une meilleure évaluation des cryptosystèmes.

I.2.3.2. Sécurité calculatoire

La sécurité calculatoire est liée à la quantité de ressources informatiques. Elle suppose que si on ne dispose que d'une puissance de calcul limitée, alors il est impossible de déduire le texte clair. C'est-à-dire, qu'il est impossible de résoudre certains problèmes de calculs très complexes (la factorisation de grands nombres ou l'extraction de logarithme discret par exemple) dans un temps raisonnable.

La sécurité des algorithmes est démontrée en montrant que la capacité d'un adversaire à casser le procédé avec les ressources existantes et avec une probabilité significative est impossible. Shordans [17], a proposé une méthode basée sur un ordinateur quantique (théorique) pour résoudre les problèmes difficiles. Cette méthode suscite que la sécurité calculatoire est un fait directement lié aux avancements technologiques.

I.2.3.3. Sécurité sémantique

La sécurité sémantique, introduite par Goldwasser et Micali dans [18], s'adresse aux cryptosystèmes asymétriques, et coïncide avec la notion de la sécurité parfaite limitée aux chiffrements probabilistes. L'exigence de la sécurité sémantique est très forte et dit qu'on ne peut pas extraire une information, même partielle, sur le texte clair à partir du texte chiffré et de la clef publique. La sécurité sémantique est considérée comme insuffisante parce qu'elle prévoit que l'attaquant a connaissance seulement du texte chiffré et de la clef publique. Elle ne prend pas en compte les autres types d'attaques comme l'attaque à texte chiffré choisi.

I.3. Algorithmes de chiffrement et clefs

I.3.1. Chiffrement asymétrique

La théorie du système asymétrique, publiée par Diffie et Hellman en 1976 a marqué le début de la cryptologie moderne. Ils ont bouleversé la façon de crypter avec l'idée que chaque utilisateur a deux clefs, une privée et une publique (figure I.1). Dans leur approche, deux nombres premiers p et g (g < p) sont rendus publics. Chaque utilisateur U choisit un numéro secret α tel que {α∈Z, 0≤α≤p} et calcule $U_\beta = g^{U\alpha} \bmod p$. Si l'utilisateur A(Alice) veut communiquer avec l'utilisateur B(Bob), par exemple, il doit alors choisir son numéro secret A_α, calculer A_β et l'envoyer à B.

L'utilisateur B, à son tour, fait la même procédure et envoie B_β à A. Un numéro commun $K=B_\beta^{A\alpha} \bmod p = A_\beta^{B\alpha} \bmod p$ est calculé de chaque côté et utilisé comme clef de cryptage.

La notion fondamentale sur laquelle repose les concepts des principaux algorithmes asymétriques est celle de fonction à sens unique à brèches secrètes (Trapdoor one-way function). Les brèches secrètes ou trappes sont des informations qui permettent d'inverser facilement les fonctions à sens unique. Pour les algorithmes qui utilisent des fonctions à sens unique, nous citons : le RSA (Rivest Shamir Adelman), fondé sur la difficulté de factorisation de grands nombres entiers.

Du fait de l'utilisation de grands nombres premiers, les cryptosystèmes asymétriques nécessitent une quantité de calcul importante, ce qui les rend très lents par rapport aux systèmes symétriques. En pratique, ils ne sont pas adaptés au chiffrement d'un volume important de données comme des images. Le RSA, le crypto système asymétrique le plus populaire, est 1500 fois plus lent que l'algorithme symétrique comme le DES (Data Encryption Standard) [10].

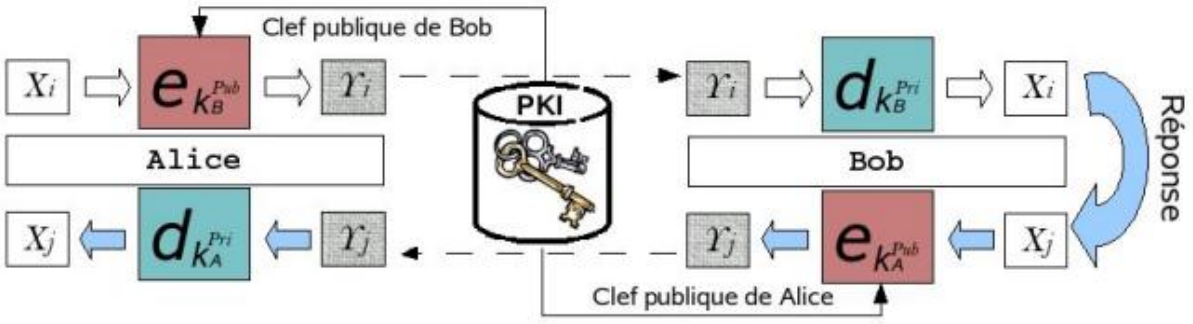


Figure I.1 Principe du chiffrement asymétrique

I.3.1.1. Théorie des nombres

Des nombreuses fonctions utilisées dans les cryptosystèmes asymétriques viennent des problèmes de la théorie des nombres. L'algorithme RSA, par exemple, utilise :

- le théorème des restes chinois pour simplifier les calculs d'arithmétique modulaire;
- le petit théorème de Fermat pour aider dans l'obtention des fonctions à sens unique;

- l'algorithme d'Euclide pour calculer le PGDC (plus grand diviseur commun)
- et l'algorithme d'Euclide étendu pour calculer la clef privée [10].

I.3.1.2. Théorème des restes chinois

C'est un théorème qui permet de résoudre certains systèmes de congruence.
Soit m_1, ..., m_k k entiers positifs deux à deux premiers entre eux et a_1, . . . ,a_k k entiers tels que $0 \leq a_i < n_i$, pour i = 1, . . . , k.
Nous calculons alors l'unique entier z tel que $0 \leq z < n$ et $z = a_i \bmod n_i$ pour t ou t_i= 1, ..., k
Où $n = \Pi_i n_i$.

I.3.1.3. Algorithme d'Euclide étendu

Si d est le plus grand diviseur commun de a et b, il y a donc deux entiers u et v tels Que :

$$d = au + bv \qquad \text{I.9}$$

L'algorithme d'Euclide étendu permet de calculer les coefficients u et v.

I.3.1.4. Petit théorème de Fermat

Soit p un nombre premier et b $\in$Zp, alors :

$$b^p \equiv b \ (\text{mod} p) \qquad \text{I.10}$$

C'est à-dire que si un entier b est multiplié par lui-même p fois, et si b lui est soustrait, le résultat est divisible par p.

I.3.1.5. Algorithme RSA

L'algorithme RSA (nommé par les initiales de ses trois inventeurs: Ronald Rivest, Adi Shamir et Leonard Adleman) est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Dans notre cryptosystème que nous avons conçu, cet algorithme est utilisé pour l'échange sécurisé des clefs, et pour assurer la fonction d'authentification. L'algorithme RSA sera détaillé dans le chapitre II.

I.3.1.6. PKI - Infrastructure de gestion de clef

Le chiffrement asymétrique a éliminé le problème de diffusion d'une clef secrète, mais il a posé le problème de la certification de la clef publique. Un utilisateur qui veut chiffrer un message avec un cryptosystème asymétrique nécessite la clef publique du destinataire.

Le problème majeur réside dans la nécessité d'associer une clef publique à l'identité de son détenteur légitime. Il est possible de créer une fausse signature numérique en remplaçant la clef publique d'une personne. La certification des clefs a alors un rôle très important dans les systèmes asymétriques.

Il existe deux modèles pour certifier une clef publique :

- Le premier modèle est fondé sur une relation de confiance directe avec son détenteur. Par exemple, la logique web of trust associée à PGP (Pretty Good Privacy) [19].
- Le second modèle repose sur le fait que tous les interlocuteurs ont confiance en un tiers : les institutions d'infrastructure de gestion des clefs (IGC) en anglais PKI - (Public Key Infrastructure).

Le mécanisme de certification des clefs est très lourd à mettre en œuvre, mais il est fondamental pour sécuriser les systèmes asymétriques de chiffrement. Il existe un nombre important d'institutions PKIs privées dans le monde, particulièrement en Europe, l'institut européen des normes de télécommunication (ETSI) règle leurs créations.

I.3.2. Chiffrement symétrique

Les systèmes de chiffrement symétrique, disposent de trois éléments : une clef secrète K, une fonction de cryptage e_K et une fonction de décryptage d_K. Contrairement aux cryptosystèmes asymétriques, les symétriques ont une faible consommation de ressources de calcul et utilisent une clef unique K pour e_K et d_K. Par contre, le désavantage du chiffrement symétrique est l'imposition d'un canal sécurisé pour l'échange de la clef.

Pour la Distribution de la clef, on distingue deux principales approches d'établissement de clefs en systèmes symétriques :

- l'échange de clef (key exchange) où un interlocuteur produit une clef secrète et la transmet à l'autre interlocuteur.
- Et le partage de clef (key agreement) où les interlocuteurs s'entendent sur une clef secrète en utilisant une source aléatoire de création.

L'échange de clef nécessite l'existence d'un tiers de confiance TA (Trusted Authentication Authority).

Si nous ne souhaitons pas utiliser un serveur de clefs en ligne, nous devons utiliser une technique de partage de clefs. La plus ancienne et la plus connue est la technique de Diffie-Hellman basée sur leur algorithme asymétrique.

I.3.3. Chiffrement hybride

Le concept de chiffrement hybride fait appel aux deux techniques, symétrique et asymétrique, comme présenté sur la figure I.2, Il a été mis en œuvre par Zimmermann pour le PGP (Pretty Good Privacy) en 1991.

L'idée d'un système hybride est d'utiliser la rapidité de l'algorithme symétrique et la sécurité de l'asymétrique. Une clef secrète K de 128bits est générée automatiquement pour la session. Le message m est chiffré avec cette clef K en utilisant un chiffreur symétrique, $m' = e_K(m)$. La clef K est alors chiffrée avec un chiffreur asymétrique en utilisant la clef publique du destinataire B, $K' = e\, k_B^{Pub}(K)$. Ensuite, le message entier $M = m' + K'$ (message chiffré symétriquement et clef asymétriquement) est envoyé au destinataire. De l'autre côté, B utilise sa clef privée k_B^{Pri} pour décrypter la clef K' et ensuite déchiffrer le message. Un exemple d'un

système hybride est le protocole SSL (Secure Socket Layer) développé par les sociétés Netscape et RSA Security, cette dernière est responsable de l'algorithme RSA.

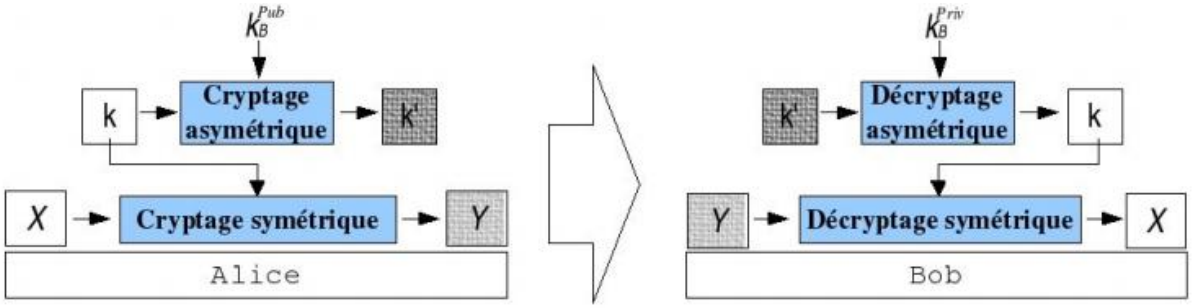


Figure I.2 Cryptage hybride

I.3.4. Quelques remarques sur la taille des clefs

Un paramètre essentiel pour la sécurité d'un cryptosystème est la longueur de la clef. Une clef de longueur de 128 bits est considérée comme sûre pour les systèmes symétriques, néanmoins pour les asymétriques cette longueur est considérée comme extrêmement faible.

Il existe plusieurs façons d'essayer de casser une clef. La manière la plus simple de trouver une clef de 512 bits, dans le chiffrement RSA par exemple, est d'essayer la factorisation du nombre n. Actuellement, les algorithmes peuvent factoriser des nombres avec 154 chiffres décimaux (512 bits). Cet exploit a été réalisé en 1999 par onze équipes scientifiques et la factorisation a pris deux mois de calculs répartis sur 300 ordinateurs. De plus, Adir Shamir, un des créateurs du RSA, a proposé une machine théorique qui peut factoriser des nombres premiers de 1024 bits [20]. De ce fait une clef de 1024 bits n'est plus estimée comme sûre.

Les normes NIST FIPS 140-26 indiquent les longueurs des clefs pour les cryptosystèmes les plus utilisés. A titre de comparaison, pour obtenir un niveau de sécurité équivalent à l'algorithme AES avec une clef de 128 bits, l'algorithme RSA doit employer une clef de 3072 bits. Pour une clef de 256 bits en AES, il est nécessaire d'utiliser une clef de 15360 bits en RSA.

I.4. Chiffrement par flot

Le chiffrement par flot (stream cipher) crypte séparément un caractère individuel (une suite de bits), en utilisant une transformation qui varie au fur et à mesure du temps. L'idée principale du chiffrement par flot est de produire un flux de clefs (key stream) $z=z_1, , \ldots, z_n$ et d' utiliser z, conjointement avec le texte clair $x=x_1, x_2, \ldots, x_n$ pour générer le texte chiffré $y=y_1, y_2, \ldots, y_n$.

Le chiffrement par flot est mieux implanté en hardware parce qu'il demande des circuits peu complexes et avec de petites zones mémoire (buffer). Il est ainsi conseillé pour les environnements bruités avec des ressources limitées. Ceci est le cas de la téléphonie GSM (Global System for Mobile Communications) qui se sert du chiffrement par flot A5/1/2.

I.4.1. Chiffrement de Vernam (One-time pad)

Malgré sa simplicité, le chiffrement de Vernam est la seule méthode sûre de façon inconditionnelle. Celui-ci est impossible à décrypter même avec une puissance de calcul infinie. La fonction de cryptage de Vernam est yi =xi⊕zi, où ⊕ est un « ou exclusif » (XOR), et de décryptage xi =yi ⊕zi. Le chiffrement de Vernam est appelé one-time pad, quand le flux de clefs z est généré aléatoirement et indépendamment.

I.4.2. Chiffrement synchrone

Un chiffrement par flot est considéré synchrone si le flux de clefs z est produit indépendamment du texte clair x et du texte chiffré y (figure I.3). Le chiffrement synchrone peut être décrit par les équations suivantes :

$$\sigma_{i+1}=f(\sigma_i, K), \qquad z_i =g(\sigma_i, K), \qquad yi =h(z_i, x_i) \qquad \text{I.11}$$

Avec σ_0 le vecteur d'initialisation, f la fonction d'état, g une fonction GNPA pour produire le key stream z, et h la fonction de sortie qui produit le texte chiffré y.

Les éléments du procédé de déchiffrement sont identiques, il suffit de remplacer

$y_i =h(z_i, x_i)$ par $x_i =h^{-1}(z_i,y_i)$.

Un exemple classique de chiffrement synchrone, les modes OFB et CTR des chiffrements par bloc. Le chiffrement synchrone est appelé chiffrement par flot additif quand la fonction h utilise un « ou exclusif » pour le chiffrement h= (zi⊕xi).

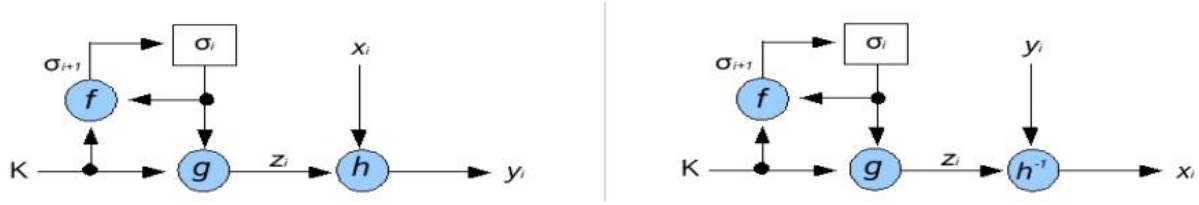


Figure I.3 Cryptage et décryptage synchrone

Dans un chiffrement synchrone, l'émetteur et le récepteur doivent être synchronisés. Si un bit est perdu ou ajouté dans le texte chiffré pendant la transmission, alors il ne peut pas être décrypté. Ce fait demande une resynchronisation qui peut être acquise par une réinitialisation ou par insertion de marques synchronisantes dans le texte chiffré.

La non-propagation d'erreur: si y_i est modifié lors de la transmission, il ne perturbera pas le déchiffrement des textes chiffrés suivants :

Attaques actives: la perte de synchronisation (première propriété) sera immédiatement détectée par le récepteur. Par contre, la deuxième propriété peut passer inaperçue auprès du récepteur. Il est donc important d'introduire des mécanismes garantissant l'authenticité et l'intégrité des données.

I.4.3. Chiffrement asynchrone

La figure I.4 illustre un système de chiffrement asynchrone (ou auto-synchronisant).

Les équations I.12 montrent que z est produit à partir de la clef K et d'un nombre fixe de caractères du flot chiffré y. Soit les équations suivantes :

$$\sigma_i = (y_{i-t}, y_{i-t+1}, \ldots, y_{i-1}), \qquad z_i = g(\sigma_i, K), \qquad yi = h(z_i, x_i) \quad \text{I.12}$$

avec $\sigma_0 = (y_{-t}, y_{-t+1}, \ldots, y_{-1})$ l'état initial, K la clef, g la fonction qui produit le flot de clefs, et h la fonction de sortie.

Actuellement, le chiffrement auto-synchronisant le plus employé est celui basé sur le chiffrement par bloc en mode CFB en utilisant un bit à la fois [21].

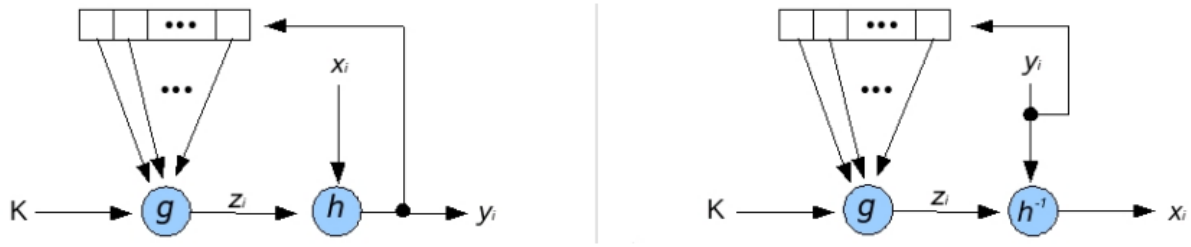


Figure I.4 Cryptage et décryptage asynchrone

Auto-synchronisation: si des bits sont perdus ou ajoutés dans y_i, le procédé se resynchronise au bout de t textes chiffrés. En effet, le déchiffrement dépend uniquement des t précédents y.

Propagation d'erreur: si y_i est modifié, le déchiffrement des t suivants y sont corrompus.

Diffusion des propriétés statistiques: Du fait que le x_i influe la suite des textes chiffrés suivants, les propriétés statistiques sont dispersées. Cela rend plus difficile les attaques basées sur une analyse statistique utilisant les redondances du texte clair.

Attaques actives: la modification d'un chiffre se répercute sur les t chiffres suivants, elle a moins de chances de passer inaperçue auprès du récepteur. Par contre, l'ajout ou la suppression de chiffres seront moins détectés que pour les procédés synchrones.

I.5. Chiffrement par bloc

Contrairement au chiffrement par flot qui utilise des transformations qui varient, le chiffrement par bloc utilise des transformations fixes sur tous les blocs de taille également fixe. En fait, le chiffrement par bloc est un cas spécial de chiffrement par flot où $z_i = K \ \forall_i \geq 1$.

Nous revenons aux théories de Shannon pour citer les propriétés de confusion et diffusion qui sont très liées au chiffrement par bloc. La confusion est l'acte de rendre la corrélation entre la clef de chiffrement et le texte chiffré la plus complexe possible. La substitution ou le remplacement d'un symbole par un autre en utilisant une boîte-S (S- BOX) est un mécanisme

pour accomplir la confusion. La diffusion est l'acte de dissiper la redondance des statistiques du texte clair dans les statistiques du texte chiffré. Elle est associée à la dépendance des bits d'entrée et de sortie. L'effet avalanche, la transposition et le réarrangement de l'ordre des symboles sont des techniques de diffusion.

La plupart des chiffrements symétriques par bloc (DES, 3DES, TEA (Tiny Encryption Algorithm) et IDEA (International Data Encryption Algorithm)) sont construits sur l'approche du réseau de Feistel.

I.5.1. Réseau de Feistel

Il s'agit d'une construction qui s'appuie sur des principes simples d'opérations répétées de permutations et substitutions des blocs de données (figure I.5), La clef K est utilisée pour générer une séquence de n sous clefs qui seront employées dans chaque ronde. Le bloc d'entrée est séparé en deux parties A et B. Une fonction f est appliquée à une des deux moitiés. Une ronde de Feistel calcule A_iB_i à partir de $A_{i-1}B_{i-1}$ selon Ai $=B_{i-1}$ et Bi $=A_{i-1}+f$ (B_{i-1}, ki). Le résultat est alors combiné avec l'autre moitié à l'aide d'un ou exclusif $\oplus$. L'inversion est très simple et il suffit d'appliquer la même transformation dans l'ordre inverse des sous-clés [21].

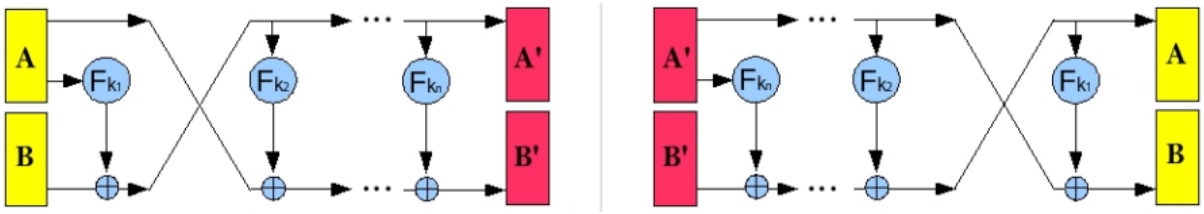


Figure I.5 Cryptage et décryptage du réseau de feistel

I.5.2. Algorithme DES et 3-DES

L'algorithme DES (Data Encryption Standard) a été adopté comme standard en 1976 par le NBS. Le DES est un réseau de Feistel à 16 rondes, à clef k de 56 bits diversifiée en 16 clefs de 48 bits, codant des blocs de 64 bits. Il utilise des tables de substitution (boite-S) fixes pour rendre la confusion.

Le Triple DES ou 3DES a été proposé par Walter Tuchman. Il consiste à appliquer les DES trois fois à l'aide de différentes clefs. Plusieurs approches sont possibles en fonction de la quantité de clefs et du type d'opérations. Un chiffrement avec une clef de 112 bits est obtenu par l'application du DES trois fois avec 2 clefs différentes de 56 bits, soit $Y_i = e_{k1}(e_{k2}(e_{k1}(X_i)))$.

Le DES-EEE3 a la même approche, mais utilise trois clefs distinctes $Y_i = e_{k3}(e_{k2}(e_{k1}(X_i)))$.

Le DES-EDE emploie deux clefs pour les opérations cryptage/décryptage/cryptage,

$Y_i = e_{k1}(d_{k2}(e_{k1}(X_i)))$.

Le triple DES le plus sûr est le DES-EDE3 $Y_i = e_{k3}(d_{k2}(e_{k1}(X_i)))$.

Aujourd'hui, l'algorithme DES n'est plus recommandé à cause de la longueur trop petite de clef et de sa lenteur d'exécution [22].

I.5.3. L'algorithme standard AES

AES (Advanced Encryption Standard), aussi connu sous le nom de Rijndael, est un algorithme de chiffrement symétrique. Il remporta en octobre 2000, le concours AES, lancé en 1997 par le NIST et devient le nouveau standard de chiffrement pour les organisations du gouvernement des États-Unis. Nous utilisons dans notre cryptosystème, l'AES pour le chiffrement des images Météosat MSG, pour les nombreux avantages qu'il représente en comparaison avec d'autres algorithmes symétriques. Il fera l'objet de notre étude dans le chapitre II.

I.5.4. Modes d'opération

Un mode de fonctionnement est une technique visant à accroitre l'effet d'un algorithme de cryptage ou l'adapter pour une application. Dans notre cas, c'est la transmission des images MSG. Ces modes sont destinés à être utilisés avec n'importe quel algorithme de chiffrement symétrique, y compris le triple DES et l'AES. Ces différents modes d'opération seront développés dans le chapitres III.

I.6. Cryptanalyse [10]

La cryptanalyse ou l'attaque regroupe tous les moyens de déchiffrer un texte crypté sans avoir connaissance de la clef. Les procédés de cryptanalyse pour le chiffrement symétrique sont très nombreux et la plupart des attaques sont spécifiquement adaptées aux techniques de chiffrement. Les différents types d'attaques sont en fonction des données supposées connues par les attaquants, parmi elles on distingue :

- **L'attaque à texte chiffré seulement (Ciphertext-onlyattack) :** L'attaquant a connaissance du texte chiffré de plusieurs messages.
- **L'attaque à texte clair connu (Known-plain textattack) :** Le cryptanalyste a accès à plusieurs textes chiffrés ainsi qu'aux textes clairs correspondants.
- **L'attaque à texte clair choisi (Chosen-plain textattack) :** L'attaquant a accès à l'algorithme de chiffrement. Il l'utilise pour générer des couples (Xi, Yi) de son choix. La différence principale par rapport l'attaque à texte clair connu est que le cryptanalyste peut choisir le texte à chiffrer.
- **L'attaque à texte chiffré choisi (Adaptative-plain textattack) :** Le cryptanalyste a accès à l'algorithme de décryptage. Il peut choisir les textes à déchiffrer sans connaître la clef.
- **L'attaque par force brute (Brute-force attack) ou l'attaque exhaustive :** L'attaquant essaie toutes combinaisons de clefs possibles jusqu'à l'obtention du texte clair.
- **L'attaque par canaux auxiliaires :** Toutes les façons d'analyser les propriétés inattendues d'un algorithme sont prises en compte pour réussir à casser le cryptosystème. Dans les algorithmes de chiffrement implémentés en hardware, par

exemple, la consommation électrique pour chaque type de calcul du chiffrement peut être utile pour déduire certaines informations de la clef.

- **L'attaque algébrique :** Les attaques algébriques sont des attaques à clair connu qui exploitent des relations algébriques entre les bits du clair, ceux du chiffré et ceux de la clef secrète. La connaissance de plusieurs couples clairs-chiffrés fournit donc un système d'équations dont les inconnues sont les bits de la clef secrète. Ces derniers peuvent alors être retrouvés en résolvant le système, ce qui est possible s'il est de degré faible, de petite taille ou qu'il possède une structure particulière.

La création de techniques modernes de chiffrement a fait ressortir des nouvelles méthodes de cryptanalyse. Le souci le plus grand des cryptographes alors est le classement de ces schémas cryptographiques selon leur niveau de sécurité face aux attaquants. Nous pouvons grouper les diverses techniques de cryptanalyse en deux grandes familles.

I.6.1. Cryptanalyse différentielle [10]

Elle a été proposée par Eli Biham et Adi Shamir en 1991. Elle permet de trouver la clef en utilisant une quantité de textes clairs. L'idée est de fournir comme entrée des textes clairs avec de légères différences (un bit par exemple). Ensuite, on analyse statistiquement le comportement des sorties selon les entrées pour retrouver la clef. En regardant comment les différences en entrée affectent les sorties, on peut établir des règles statistiques. Il existe plusieurs variantes des cryptanalyses différentielles, nous distinguons : différentielle tronquée, différentielle d'ordre supérieur et différentielles impossibles.

I.6.2. Cryptanalyse linéaire

Elle a été inventée par Mitsuru Matsui en 1993 [10]. Elle nécessite une quantité n de couples (texte clair, texte chiffré), tous chiffrés avec la même clef. Le principe est que le même message soit chiffré plusieurs fois avec des clefs différentes pour construire une immense table (téraoctet) qui contient toutes les versions chiffrées de ce message. Lors d'une interception d'un message chiffré, on peut le retrouver dans la table et obtenir la clef qui avait été utilisée pour le cryptage. Cette attaque n'est bien sûr pas faisable car nous aurions besoin d'une table trop importante. Le génie d'Hellman a été de trouver un moyen pour réduire cette table, processus réalisable. Celui-ci consiste à faire une approximation linéaire de l'algorithme pour le simplifier.

I.7. Discussion

Dans ce chapitre, nous avons présenté plusieurs techniques et quelques théories de la cryptographie qui vont nous permettre de comprendre cet axe de recherche. Nous avons évoqué les notions formelles de sécurité et leurs implications. Nous avons ensuite abordé les différents types de classifications des algorithmes de chiffrement et leurs contextes d'applications. Nous avons aussi observé que les clefs ont un rôle important et le choix de leur longueur est cruciale pour rendre sûrs les cryptosystèmes. Ce chapitre a introduit aussi, les principaux algorithmes de cryptage symétrique, asymétrique, par flot et par bloc, et a présenté également les différentes formes d'attaques et leurs classifications.

Nous détaillons dans le prochain chapitre les deux algorithmes standards AES et RSA qui seront utilisés dans notre système de transmission sécurisé. Il est important de comprendre le fonctionnement de ces deux algorithmes, pour adapter ces méthodes de chiffrement à la transmission des images satellitaires MSG.

Chapitre II

Les algorithmes de cryptages AES et RSA

II.1. Préambule

Dans ce chapitre, nous détaillons les deux algorithmes AES et RSA qui serons utilisés dans notre cryptosystème de transmission.

« Advanced Encryption Standard » ou AES, aussi connu sous le nom de Rijndael, est un algorithme de chiffrement symétrique. Il remporta en octobre 2000 le concours AES, lancé en 1997 par le NISTet devient le nouveau standard de chiffrement pour les organisations du gouvernement des États Unis. Il a été également approuvé par la NSA (National Security Agency) pour les informations top secrètes. Il est issu d'un appel à candidatures international lancé en janvier 1997 et ayant reçu 15 propositions. Parmi ces 15 algorithmes, 5 furent choisis pour une évaluation plus poussée en avril 1999 : MARS, RC6, Rijndael, Serpent, et Twofish [33]. Au bout de cette évaluation, ce fut finalement le candidat Rijndael, du nom de ses deux concepteurs Joan Daemen et Vincent Rijmen (tous les deux de nationalité belge) qui a été choisi. Ces deux experts en cryptographie étaient déjà les auteurs d'un autre algorithme: Square. AES est un sous-ensemble de Rijndael : il ne travaille qu'avec des blocs de 128 bits alors que Rijndael offre des tailles de blocs et de clefs qui sont des multiples de 32 (compris entre 128 et 256 bits). Ce faisant, l'AES remplace le DES (choisi comme standard dans les années 1970) qui de nos jours devenait obsolète, car il utilisait des clefs de 56 bits seulement. L'AES a été adopté par le NIST (National Institute of Standards and Technology) en 2001. De plus, son utilisation est très pratique, car il consomme peu de mémoire et n'étant pas basé sur un schéma de Feistel, sa complexité est moindre et il est plus facile à implémenter.

RSA (Rivest Shamir Adleman) est le premier système à clé publique solide à avoir été inventé, et le plus utilisé actuellement, est le système RSA Publié en 1978 par Ron Rivest, Adi Shamir et Leonard Adleman. Le RSA est fondé sur deux principes mathématiques fondamentaux : la difficulté de factoriser des grands nombres, et l'arithmétique des congruences.

II.2.Structure de l'AES [3] [4]

La figure II.1 montre la structure de l'AES. La taille de bloc en clair est de 128 bits, soit 16 octets. La longueur de la clef peut être 16, 24 ou 32octets (128, 192, ou 256 bits). L'algorithme est appelé AES-128, AES-192, ou AES-256, en fonction de la longueur de la clé.

Ce bloc est copié dans une matrice d'état, qui est modifié à chaque étape de chiffrement ou de déchiffrement. Après la dernière étape, la matrice d'état est copiée dans une matrice de sortie. Ces opérations sont représenté sur la figure II.2 a. De même, la clef est représentée comme une matrice carrée d'octets 4×4. La figure II.2 b montre l'expansion de la clé de 128 bits. Les quatre premiers octets de la clé forment un mot.

Le chiffrement se compose de tours, où le nombre des tours dépend de la longueur de la clef: 10 tours pour une clé de 16 octets, 12 tours pour une clé de 24 octets, et 14 tours pour une clé de 32 octets (tableau II.1).

Les premiers tours se composent de quatre fonctions: « SubBytes, ShiftRows, MixColumns et AddRoundKey » qui sont décrits par la suite. Le tour final ne contient que trois transformations, et il ya une seule transformation initiale (AddRoundKey) avant le premier tour, Chaque transformation prend un ou plusieurs matrices 4×4 en entrée et produit en sortie une matrice 4×4. La figure II.1 montre que le sortie de chaque tour est une matrice 4×4, à la sortie de la ronde finale étant la partie chiffrée.

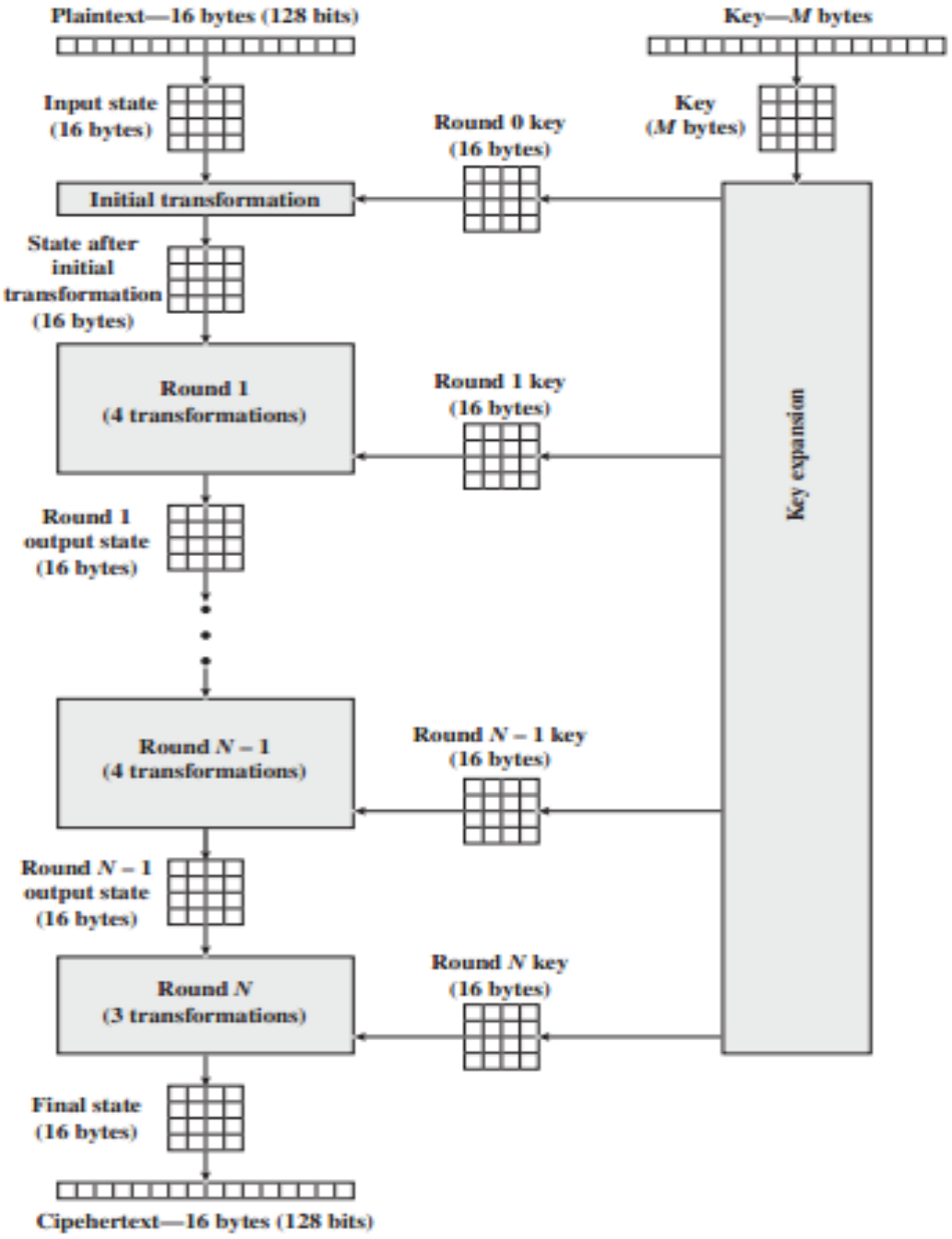


Figure II.1 Structure général de l'AES

Numéro des tours	La longueur de clef (octets)
10	16
12	24
14	32

Tableau II -1a longueur de clef en fonction du nombre des tours

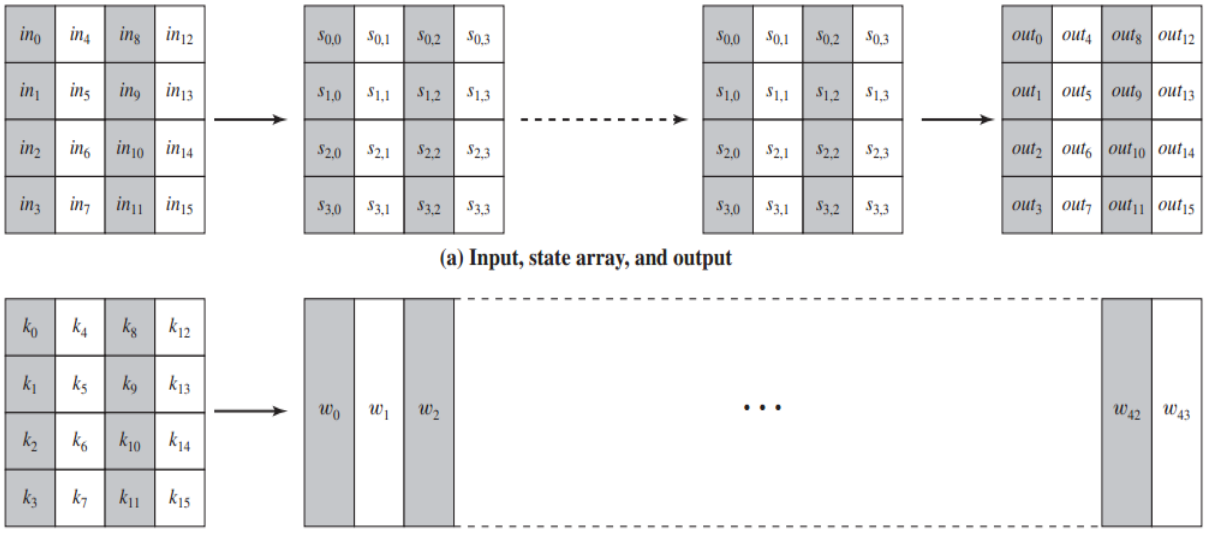


Figure II.2 a) Forme matricielle de l'AES, b) la clef étendue

La taille des clefs (mots/octets/bits)	4/16/128	6/24/192	8/32/256
La taille de texte clair en bloc (mots/octets/bits)	4/16/128	4/16/128	4/16/128
Le nombre des tours	10	12	14
La taille de clef pour chaque tour (mots/octets/bits)	4/16/128	4/16/128	4/16/128
La taille de clef étendu (mots/octets/bits)	44/176	52/208	60/240

Tableau II-2Longueur de la clef

La figure II.3montre la structure de l'algorithme de chiffrement AES de façon plus détaillée. Avant d'entrer dans les détails, nous pouvons faire quelques remarques sur La structure de l'algorithme :

- Une caractéristique remarquable de cette structure est qu'elle n'est pas une structure de Feistel.
- La clef qui est fourni en entrée est élargie en une matrice de quarante-quatre
  Mots 32 bits, w [i]. Quatre mots distincts (128 bits) servent pour assurer le chiffrement pour chaque tour; elles sont indiquées dans la figure II.3.

- Quatre phases différentes sont utilisées pour chaque tour : une de permutation et trois de substitution:
  - « Substitute bytes » utilise une boîte S pour effectuer une substitution octet par octet
  - « ShiftRows » Une simple permutation entre les octets
  - « MixColumns » Une substitution qui rend l'utilisation de l'arithmétique autour GF (2^8)
  - « AddRoundKey » une simple XOR entre le bloc courant avec une partie de la clef étendue
- L'étape de « AddRoundKey » est la seule qui utilise la clef. Pour cette raison, le procédure de chiffrement commence et se termine par une étape de « AddRoundKey ».
- Chaque étape est facilement réversible. Pour « Substitute bytes », « ShiftRows », « MixColumns » une fonction inverse est utilisée dans l'algorithme de déchiffrement. pour l'étape de « AddRoundKey », l'inverse est réalisé par XOR : $A \oplus B \oplus B = A$.
- L'algorithme de chiffrement n'est pas identique à l'algorithme de déchiffrement, ceci est une particularité de l'AES.
- La tour finale de chiffrement et le déchiffrement se compose seulement de trois étapes. Encore une fois, ceci est une particularité de la structure de l'AES.

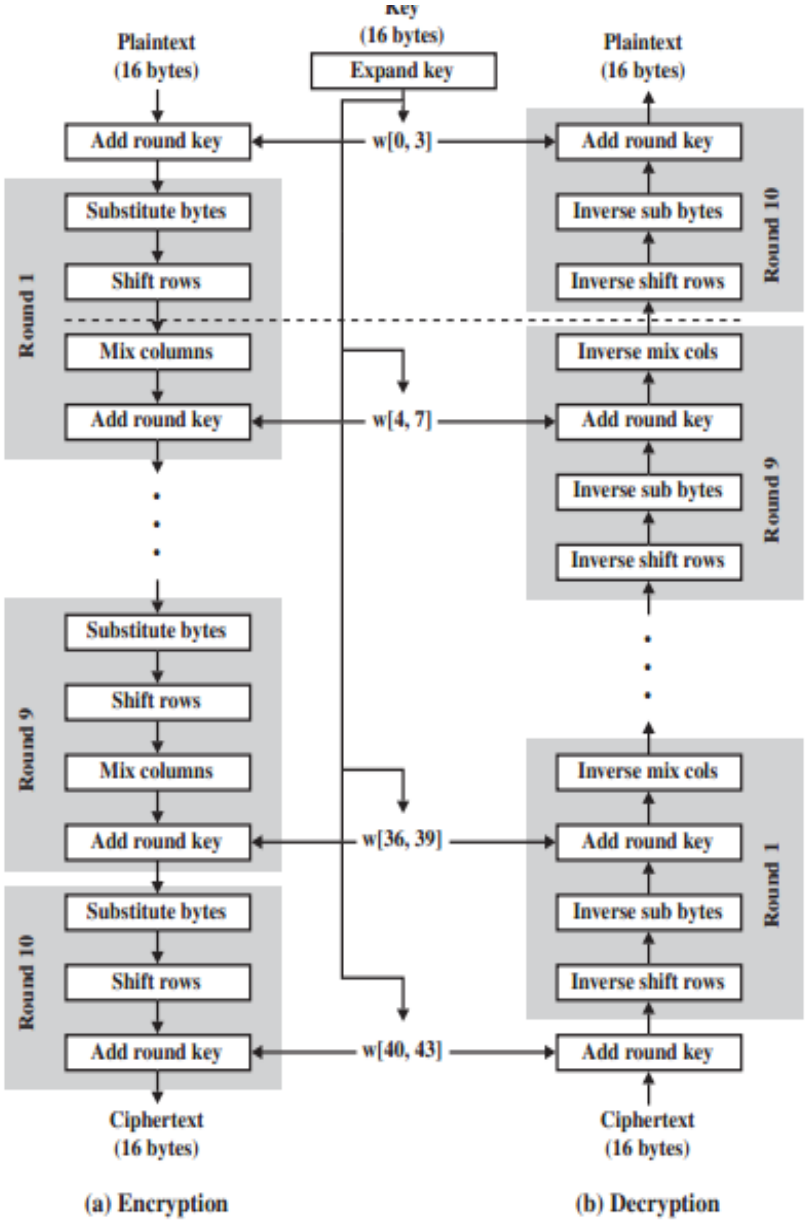


Figure II.3 AES : chiffrement et déchiffrement

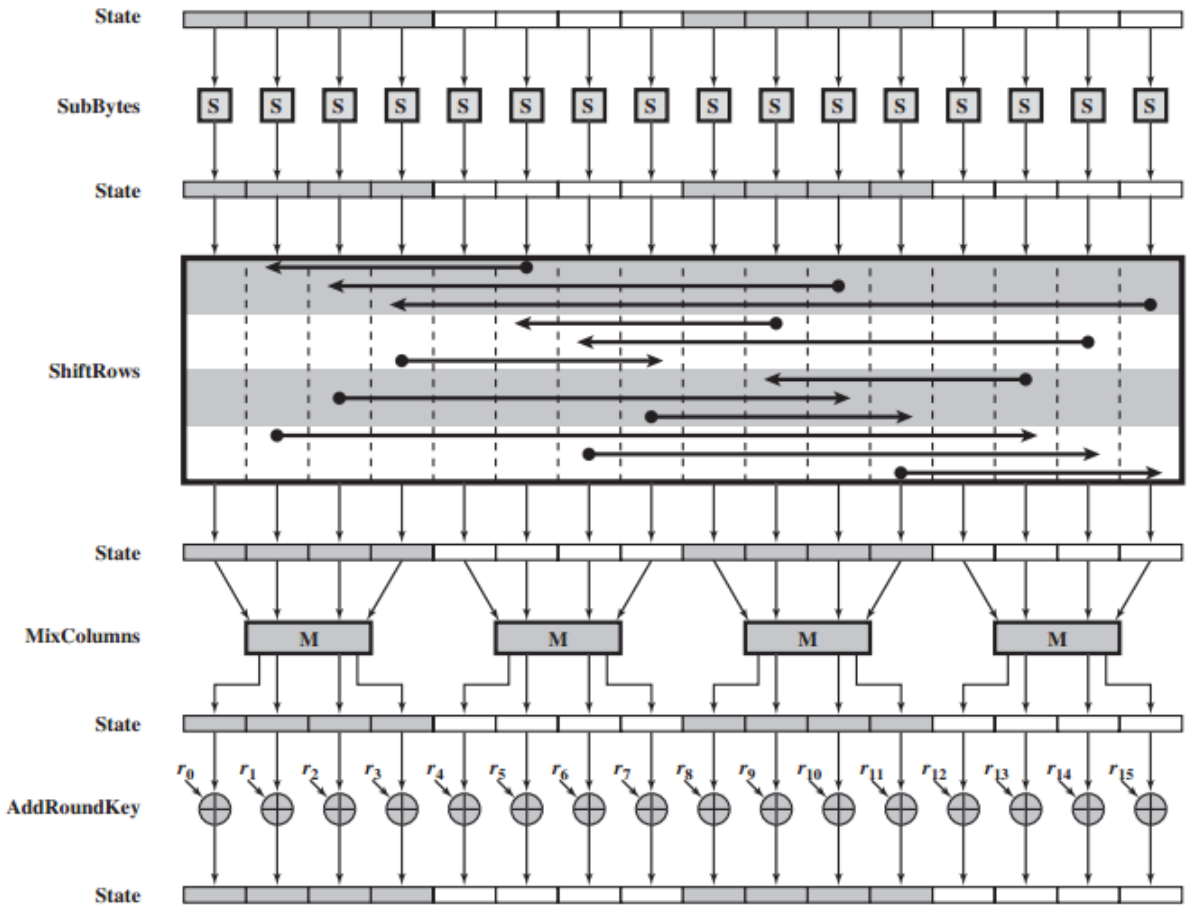


Figure II.4 les tours de chiffrement de l'AES

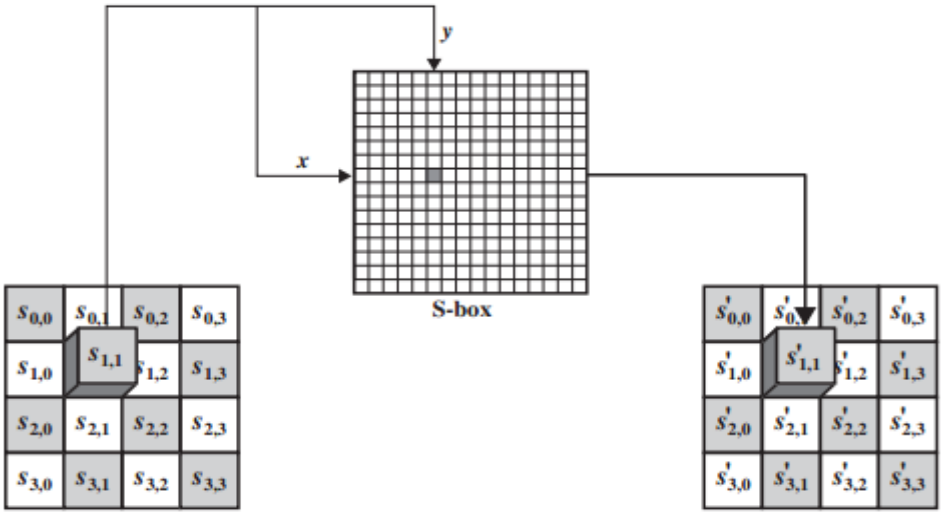
II.2.1. Les fonctions de transformation de l'AES

Nous passons maintenant à la discussion de chacun des quatre transformations utilisées en AES.

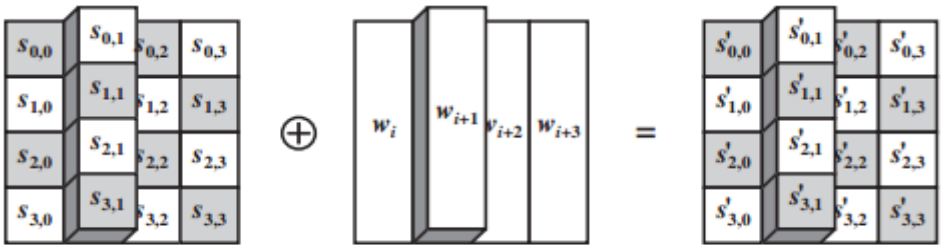
II.2.1.1. « Substitute Bytes Transformation »

Nommée aussi « SubBytes », cette transformation est une simple consultation de table (II.5 a). AES définit une matrice d'octets 16×16, appelé une boîte-S (Tableau II.2 a), qui contient l'ensemble des 256 valeurs possibles de 8 bits.

Chaque octet individuel est remplacé par un nouvel octet de la façon suivante: Les 4 bits les plus à gauche de l'octet sont utilisés en tant que valeur de ligne et les 4 bits les plus à droite sont utilisés en tant que valeur de colonne. Ces valeurs de rangée et de colonne servent d'index dans la boîte-S pour sélectionner le nouvel octet. Par exemple, la valeur hexadécimale 95 est référencée ligne 9, colonne 5 dans S-box, qui contient la valeur 2A, la valeur 95 est remplacée par la valeur 2A.



(a) Substitute byte transformation



(b) Add round key transformation

Figure II.5 AES : Opérations aux niveaux d'octets

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

(a) S-box

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

(b) Inverse S-box

Tableau II.2 S-Box AES a) Direct, b) Inverse

Nous donnons ci-après un exemple de la transformation des « SubBytes » :

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

→

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

La boîte S est construite de la façon suivante :

1- Initialisation de la boîte S avec les valeurs d'octets dans l'ordre croissant ligne par ligne .la première ligne contient {01} {02} {03}…..{0F}, la deuxième ligne contient {10} {11} {12}…..{1F} …ect

Ainsi, la valeur de l'octet à la ligne x, et la colonne y est {yx}.

2- remplacer chaque octet dans S-Box par l'inverse multiplicatif dans le corps fini GF(2^8), la valeur {00} ne change pas.

3- Considérons que chaque octet dans S-Box se compose de 8 bits (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0), Appliquer la transformation suivante pour chaque bit de chaque octet :

$$b'_i = b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus c_i \qquad \text{II.1}$$

avec c_i est le bit numéro i de l'octet c de la valeur {63}, la sous forme matricielle de transformation est comme suit :

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad \text{II.2}$$

« Inverse substitute byte transformation » est nommée InvSubBytes, utilise la Table II.2 b pour trouver l'inverse

La boîte de S-inverse (figure II.6 b) est construite en appliquant l'inverse de la transformation de l'équation (II.1), puis en prenant l'inverse multiplicatif autour de GF(2^8).

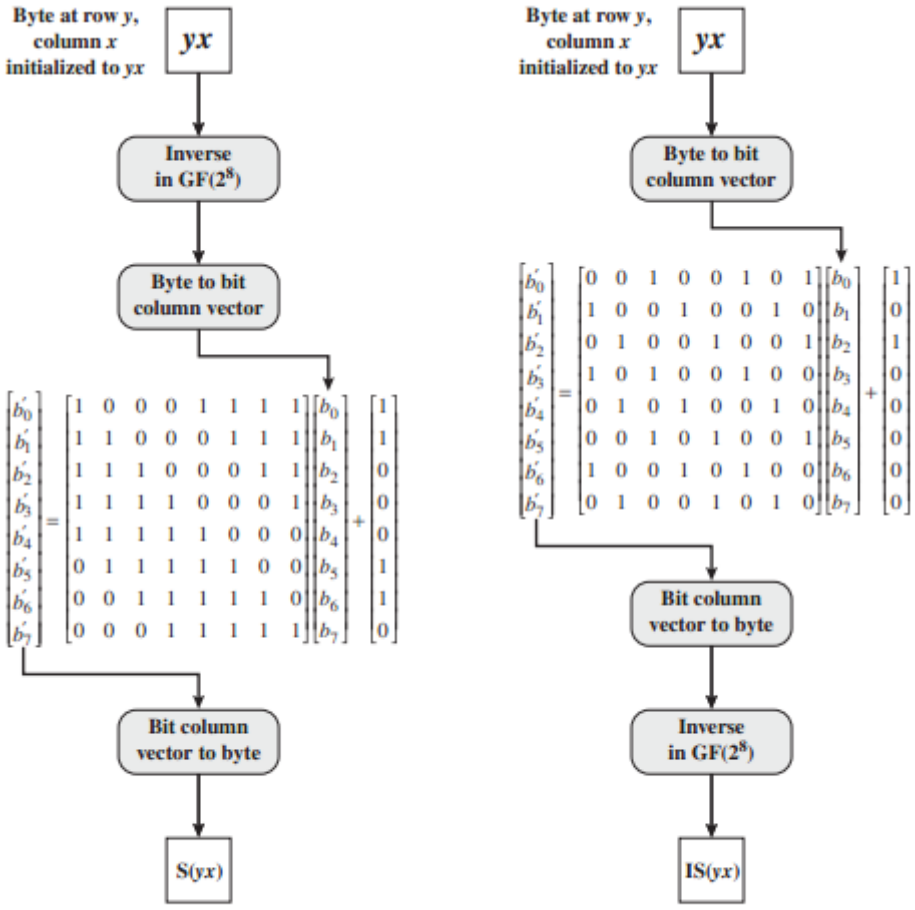
La transformation inverse est :

$$b'_i = b_{(i+2)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+7)mod8} \oplus d_i \;/\; d_i = \{05\} \qquad \text{II.3}$$

Nous pouvons illustrer cette transformation comme suit :

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{II.4}$$

Nous résumons les étapes de construction de la boite S-Box et son inverse dans la figure suivante :



(a) Calculation of byte at row y, column x of S-box

(a) Calculation of byte at row y, column x of IS-box

Figure II.6 Construction des boite de AES S-Box et IS-Box

II.2.1.2. « ShiftRows Transformation »

La transformation « ShiftRows » est représentée dans la Figure II.7. La première ligne de la matrice d'état n'est pas modifiée. Pour la deuxième ligne, un décalage circulaire de gauche du premier octet est réalisé. Pour la troisième ligne, un décalage circulaire de gauche de deuxième octet est réalisé. Pour la quatrième ligne aussi un décalage circulaire de gauche de troisième octet est réalisé. Ce qui suit, est un exemple de « ShiftRows » :

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

→

87	F2	4D	97
6E	4C	90	CE
46	E7	A4	C3
A6	8C	D8	95

« Inverse shift row transformation », appelé « ShiftRows », effectue les déplacements circulaires dans le sens inverse pour chacune des trois dernières lignes.

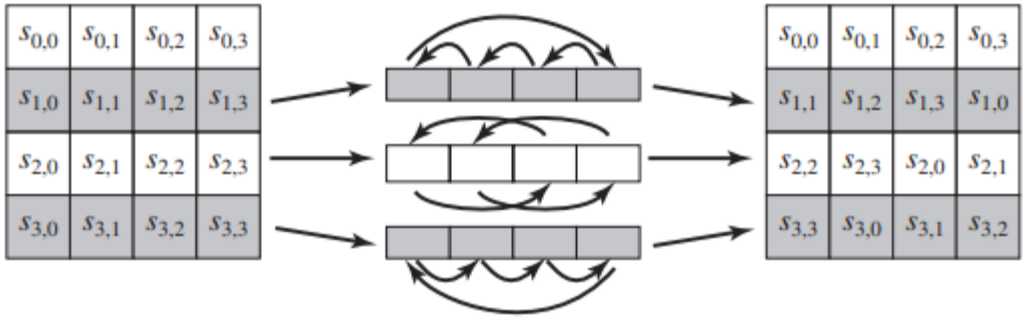


Figure II.7 ShiftRows Transformation

II.2.1.3. « MixColumns Transformation »

Appelé « MixColumns », cette transformation agit sur chaque colonne individuellement. Chaque octet d'une colonne est remplacé par une nouvelle valeur qui est en fonction de l'ensemble des quatre octets de cette colonne. La transformation peut être définie par la matrice d'état (figure II.8):

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,1} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,1} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,1} & S'_{3,3} \end{bmatrix} \quad \text{II.5}$$

Le MixColumns sur une seule colonne de l'État peut être exprimé comme suit :

$$s'_{0,j} = (2 \bullet S_{0,j}) \oplus (3 \bullet S_{1,j}) \oplus S_{2,j} \oplus S_{3,j}$$

$$s'_{1,j} = S_{0,j} \oplus (2 \bullet S_{1,j}) \oplus (3 \bullet S_{2,j}) \oplus S_{3,j} \quad \text{II.6}$$

$$s'_{2,j} = S_{0,j} \oplus S_{1,j} \oplus (2 \bullet S_{2,j}) \oplus (3 \bullet S_{3,j})$$

$$s'_{3,j} = (3 \bullet S_{0,j}) \oplus S_{1,j} \oplus S_{2,j} \oplus (2 \bullet S_{3,j})$$

« • » pour indiquer la multiplication sur le corps fini GF(2^8).

Ce qui suit est un exemple de MixColumns :

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

⟶

47	40	A3	4C
37	D4	70	9F
94	EA	3A	42
ED	A5	A6	BC

« Inverse mix colum n transformation », est défini par la multiplication matricielle suivante :

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,1} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,1} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,1} & S'_{3,3} \end{bmatrix} \quad \text{II.7}$$

Il n'est pas très clair que l'équation (II.5) est l'inverse de l'équation (II.7).

Nous devons montrer ce qui est équivalent à :

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} \quad \text{II.8}$$

On obtient :

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{II.9}$$

Pour vérifier la première colonne de l'équation (II.9), nous avons besoin de montrer :

$$(\{0E\} \bullet \{02\}) \oplus \{0B\} \oplus \{OD\} \oplus (\{09\} \bullet \{03\}) = \{01\}$$

$$(\{09\} \bullet \{02\}) \oplus \{0E\} \oplus \{0B\} \oplus (\{0D\} \bullet \{03\}) = \{00\} \quad \text{II.10}$$

$(\{0D\} \bullet \{02\}) \oplus \{09\} \oplus \{OE\} \oplus (\{0B\} \bullet \{03\}) = \{00\}$

$(\{0B\} \bullet \{02\}) \oplus \{0D\} \oplus \{O9\} \oplus (\{0E\} \bullet \{03\}) = \{00\}$

Les autres équations peuvent être vérifiées de façon similaire

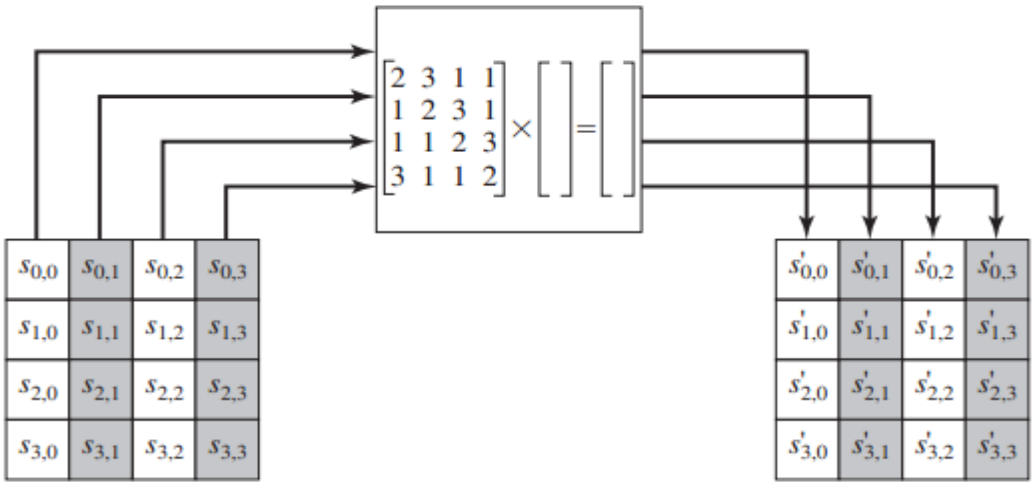


Figure II.8 MixColumns Transformation

II.2.1.4. « AddRoundKey Transformation »

Appelé « AddRoundKey », les 128 bits de la matrice d'état de texte claire sont XORed avec 128 bits de la clef. Comme le montre la figure II.9, Un exemple de AddRoundKey :

47	40	A3	4C
37	D4	70	9F
94	EA	3A	42
ED	A5	A6	BC

$\oplus$

AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

=

EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6

La première matrice est la matrice d'état
La deuxième matrice est la matrice des clefs inverse add round key transformation est identique à AddRoundKey.

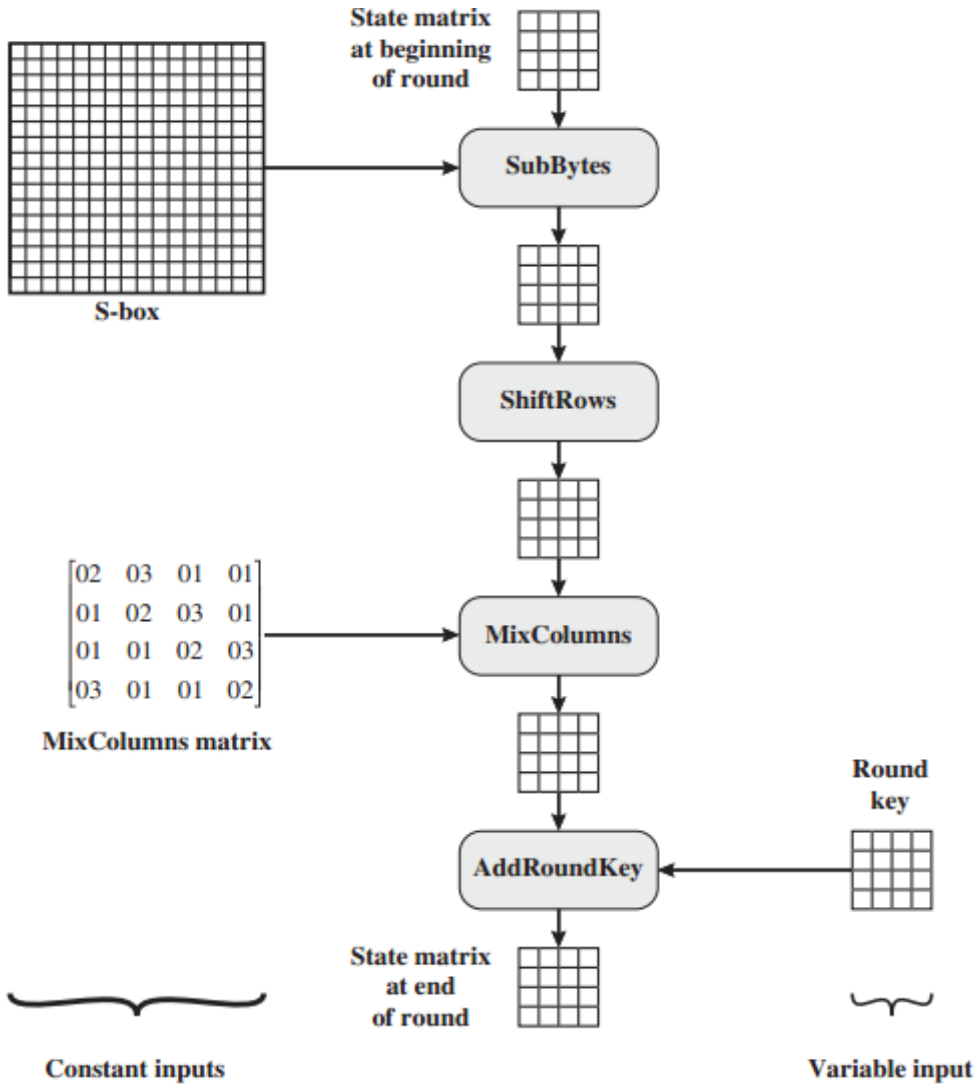


Figure II.9 « AddRoundKey Transformation »

II.2.2. La clef de l'AES étendu

L'algorithme d'extension des clefs AES prend en entrée quatre mots (16 octets) et produit 44 mots (176 octets). Ceci est suffisant pour fournir les clefs nécessaires pour 10 tours de l'algorithme AES

La Figure II.10 illustre la génération de la clef étendue à l'aide de la fonction complexe g**,** la fonction g est constituée de sous-fonctions.

```
KeyExpansion (byte key[16], word w[44])
{
    word temp
    for (i = 0; i < 4; i++)    w[i] = (key[4*i], key[4*i+1],
                                       key[4*i+2],
                                       key[4*i+3]);

    for (i = 4; i < 44; i++)
    {
     temp = w[i - 1];
     if (i mod 4 = 0)    temp = SubWord (RotWord (temp))
                                ⊕ Rcon[i/4];

     w[i] = w[i-4] ⊕ temp
    }
}
```

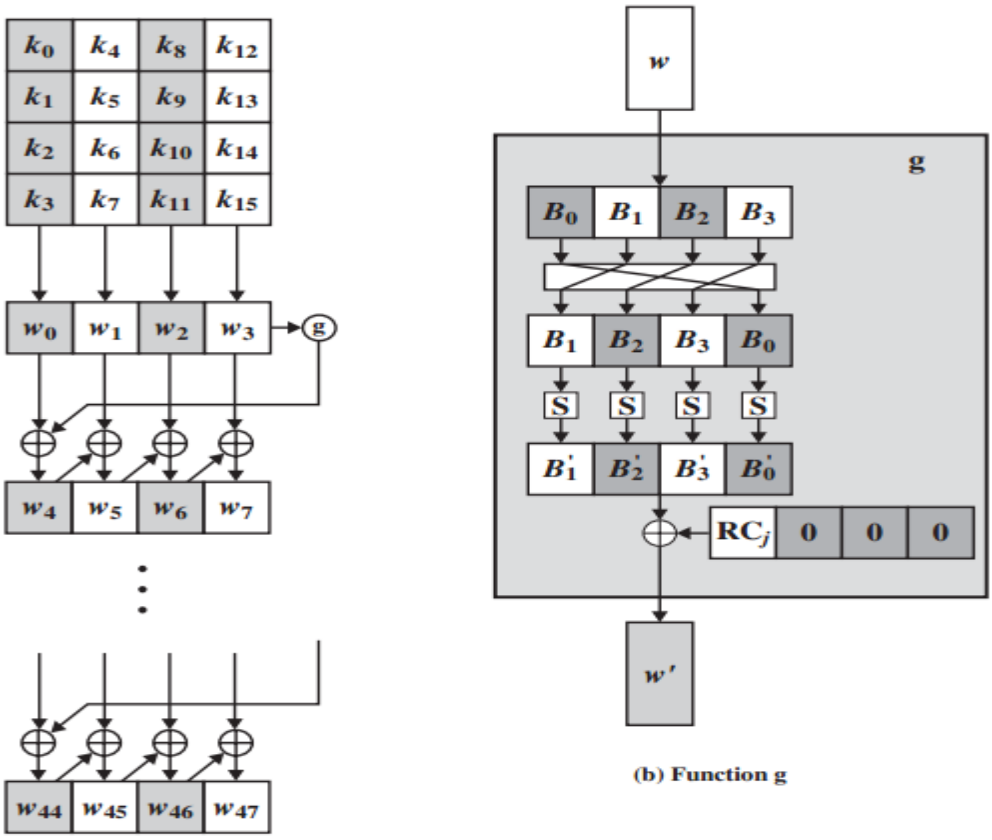


Figure II.10 La clef de l'AES étendu

« RotWord » effectue un décalage d'un octet circulaire gauche sur un mot. Cela signifie que le mot d'entrée est $[B_0,B_1,B_2,B_3]$ transformé en $[B_1,B_2,B_3,B_0]$.

« SubWord » effectue une substitution sur chaque octet, en utilisant S-Box.

Le résultat des étapes 1 et 2 est XORed avec la constante Rcon[j].

La constante est un mot dons les trois octets les plus à droite sont toujours à 0. Ainsi, l'effet d'un « OU exclusif » d'un mot avec Rcon est d'effectuer seulement une opération XOR sur l'octet le plus à gauche de ce mot. La constante est différente pour chaque tour et est définie comme Rcon[j]= (RC[j],0,0,0)

Avec RC [1]=1, RC[j]= 2×RC[j-1], et avec la multiplication définie sur le domaine GF(2^8)

La valeur de RC[j] en hexadécimale est :

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

Tableau II.3 La valeur de RC[j]

II.3. Algorithme asymétrique RSA [4] [9]

Le chiffrement RSA est un algorithme de cryptographie asymétrique, très utilisé et très efficace pour l'échange des clefs des algorithmes symétriques comme l'AES. Cet algorithme a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman.

II.3.1. Cryptosystème a clé publique

Les algorithmes asymétriques sont basés sur une clef pour le chiffrement et une clef associée différent, pour le déchiffrement. Ces algorithmes ont la caractéristique importante suivante. Il est impossible de trouver la clef de déchiffrement malgré la connaissance de l'algorithme cryptographique et la clef de chiffrement.

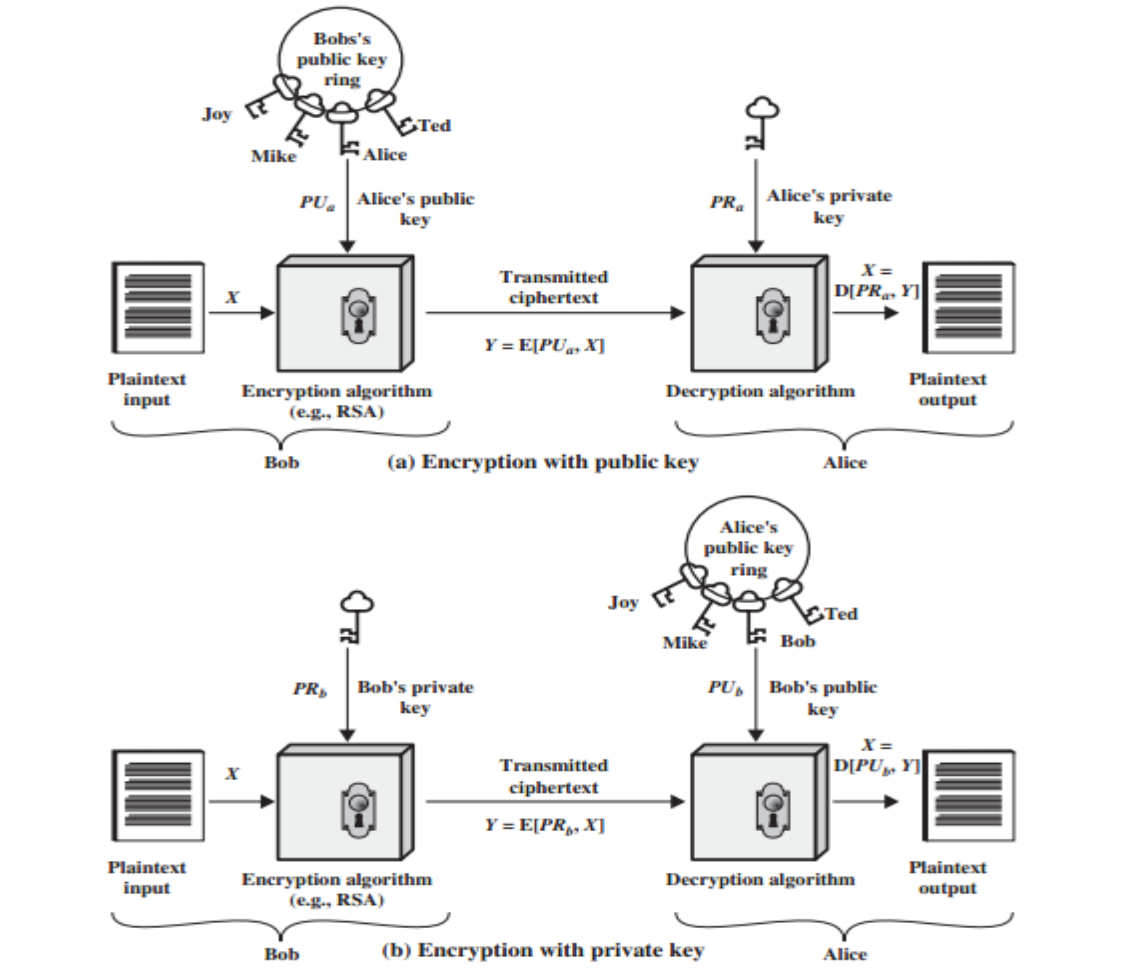


Figure II.11 Cryptosystème à clé publique

D'après la figure II.11, nous résumons les étapes d'un cryptosystème à clef asymétrique:

1. Chaque utilisateur génère une paire de clés à utiliser pour le chiffrement et le déchiffrement des messages.

2. Chaque utilisateur place sa clef publique dans un registre public ou autre fichier accessible.

3. Si Bob veut envoyer un message confidentiel à Alice, Bob chiffre le message utilisant la clé publique d'Alice.

4. Quand Alice reçoit le message, elle le déchiffre à l'aide de sa clef privée.

Il est, cependant, possible d'assurer à la fois la fonction d'authentification et de confidentialité par une double utilisation du système à clé publique comme montre la figure II.12:

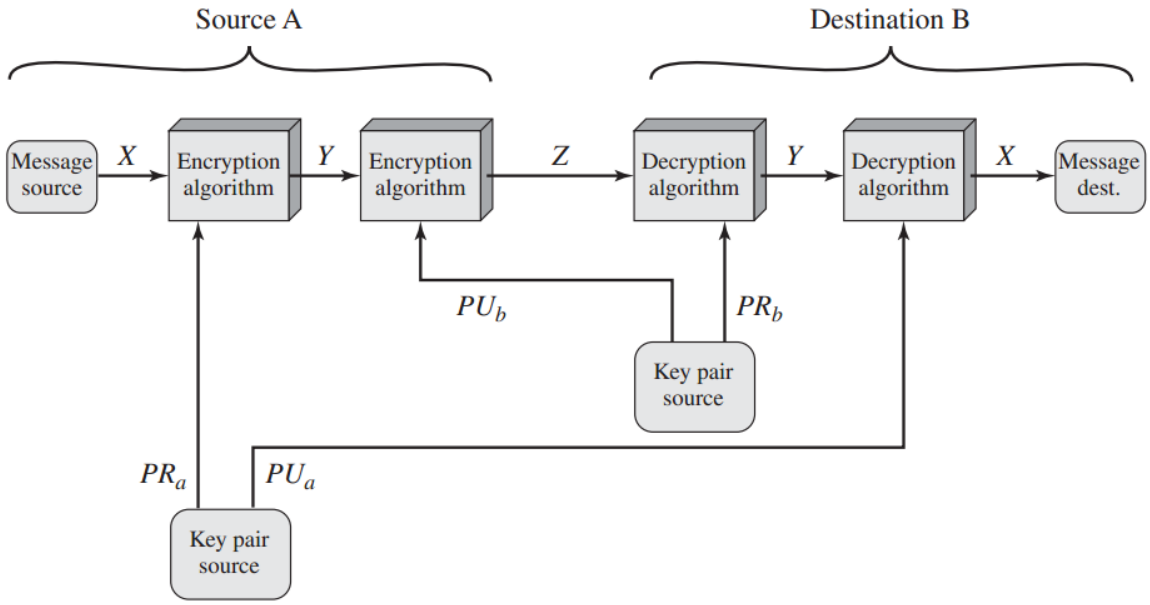


Figure II.12 Cryptosystème à clef publique (authentification et confidentialité)

Dans ce cas, nous commençons par chiffrer un message, en utilisant la clef privé de l'expéditeur. Ceci permet d'obtenir la signature numérique. Ensuite, nous chiffrons à nouveau, en utilisant la clef publique de récepteur. Le texte chiffré final peut être déchiffré que par le destinataire qui possède la clé privée correspondante. Ainsi, la confidentialité est assurée.

II.3.2. L'algorithme RSA

Le premier système à clé publique solide à avoir été inventé, et le plus utilisé actuellement, est le système RSA. Le RSA est fondé sur deux principes mathématiques fondamentaux : la difficulté de factoriser des grands nombres, et l'arithmétique des congruences.

II.3.3. Principe de fonctionnement de l'algorithme RSA

Pour qu'Alice puisse échanger sa clé avec Bob, elle doit d'abord la calculer en mettant en œuvre des notions mathématiques remarquables par leur simplicité. L'algorithme RSA est ainsi défini par 03 phases :

- Génération des clés (effectué par la destinataire Alice)
- Chiffrement (effectué par l'expéditeur Bob)
- Déchiffrement (effectué par la destinataire Alice)

II.3.3.1. Génération des clés :

Cette phase peut se résumer en 03 étapes :

1er étape : Alice choisit au hasard deux grands nombres entiers, naturels, premiers, p et q, ont environ 100 chiffres chacun ou plus pour rendre la factorisation hors de la portée. Dans notre exemple simplifié elle choisit :

p = 31 et q = 53

Et fait leur produit :

n = p×q = 1643

2er étape : Alice détermine la fonction d'Euler associée à n déjà calculé en utilisant la formule :

$\phi(n) = (p-1) \times (q-1)$

$\phi(n) = 30 \times 52 = 1560$

Une fois que la fonction d'Euler déterminée, Alice choisie au hasard sa clé publique « e », cette clé est un nombre premier compris entre 1 et $\phi(n)$ et premier relativement à $\phi(n)$ c'est-à-dire le pgcd $(e, \phi(n))=1$. Alice fait : e = 11

D'où le couple (e, n) constitue la clé publique.

$$\begin{cases} n : c'est\ le\ module \\ e : c'est\ l'exposant \end{cases}$$

La clé publique est donc (11, 1643)

3er étape : Cette dernière étape consiste à trouver la clé privée « d » qui correspond à la clé publique choisie précédemment avec d compris entre 1 et phi(n), pour se faire, il faut résoudre l'équation suivante :

$e.d \bmod \phi(n) = 1$

c.à.d : $e.d \equiv 1[\phi(n)]$

Donc : $e.d = k\,\phi(n) + 1$

Selon notre exemple on aura :

$e.d = k\,\phi(n) + 1$

$11.\ d = k\ .\ 1560 + 1$

Pour k= 6 on aura : $11.d = k.\ 1560 + 1$ on aura d = 851

Le couple (d, n) constitue la clé privée

La clé privée est donc (851, 1643)

$$\begin{cases} n : c'est\ le\ module \\ d : c'est\ l'exposant \end{cases}$$

En fin, Alice et Bob disposent toutes les clés indispensables au chiffrement et au déchiffrement des messages après la transmission ou la publication de sa clé publique (e, n).

Maintenant, il faut qu'elle conserve sa clé privée (d, n) et qu'elle n'oublie jamais les nombres p et q.

II.3.3.2. Chiffrement

Bob veut donc transmettre le message M «ANEMONE» à Alice. Il cherche dans l'annuaire la clé de chiffrement qu'Alice a déjà publiée. Il sait maintenant qu'il doit utiliser le système RSA avec les deux entiers n et e (dans notre exemple n= 1643 et e=11).

Il va procéder au cryptage de la manière suivante :

- Il transforme en nombres son message en remplaçant par exemple chaque lettre par son rang dans l'alphabet : a = 01, b = 02 ……..z = 26. il résulte :

  M = ANEMONE

  M = A N E M O N E

  M = 01 14 05 13 15 14 05

- Il découpe son message numérisé en blocs de même longueur représentant chacun une taille égale ou inférieure à celle de n ce qui empêche la simple substitution. Dans notre exemple la taille de n est 3, ce qui donne des tranches m_i de 03 chiffres chacune, le message devient :

  M = 001 140 513 151 405

  M = m_1 m_2 m_3 m_4 m_5

Chaque bloc m_i est chiffré par l'équation :

$$\mathbf{C_i = M_i^e \bmod n}$$

Ce qui donne :

$$C_1 = m_1^{11} \bmod 1643 = 001$$

$$C_2 = m_2^{11} \bmod 1643 = 109$$

.. . . .

.. . . .

$$C_5 = m_5^{11} \bmod 1643 = 374$$

$001^{11} \bmod 1643 =$	1
$140^{11} \bmod 1643 =$	109
$513^{11} \bmod 1643 =$	890
$151^{851} \bmod 1643 =$	1453
$405^{11} \bmod 1643 =$	374

Alors le message chiffré C sera:

$$C = \quad c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5$$

$$C = \ 0001 \ \ 0109 \ \ 0890 \ \ 1453 \ \ 0374$$

Enfin, Bob a son message chiffré, il peut donc l'envoyer à Alice.

II.3.3.3. Déchiffrement :

Alice reçoit le message de Bob, à partir de p et q, qu'elle a gardés secrets elle calcule la clef d de déchiffrement (c'est sa clef privée). Celle-ci doit satisfaire l'équation :

$$e \cdot d \bmod ((p-1)(q-1)) = 1$$

Chacun des blocs c_i du message chiffré sera déchiffré par l'équation :

$$M_i = C_i^d \bmod n$$

Ce qui lui donne :

$1^{851} \bmod 1643 =$	1
$109^{851} \bmod 1643 =$	140
$890^{851} \bmod 1643 =$	513
$1453^{851} \bmod 1643 =$	151
$374^{851} \bmod 1643 =$	405

M = 001 140 513 151 405

M= ANEMONE

Lors du déchiffrement, sachant qu'il faut obtenir des blocs de 2 éléments (grâce au codage particulier de l'exemple). Finalement, Alice prend sa table de correspondance alphabétique pour restituer le message M, elle aura :

01 14 05 13 15 14 05

A N E M O N E

II.3.3.4. Résumé :

Le RSA est un algorithme de chiffrement asymétrique fait appel aux notions suivantes :

1. Génération de 2 nombres premiers p et q

2. Calcul de n = p × q

3. Déterminer e tel que 3 < e < _(n) et (e,_(n)) = 1

4. Calculer d tel que e × d ≡ 1 mod phi(n)

5. Clé publique : (e, n)

6. Clé privée : (d, n)

7. p et q doivent rester secrets, voire supprimés

8. C = M^emod n et M = C^dmod n

Les deux figures suivantes représentent respectivement la manière de chiffrement un Un bloc de texte, et un exemple qui explique bien le fonctionnement de l'algorithme RSA en blocs.

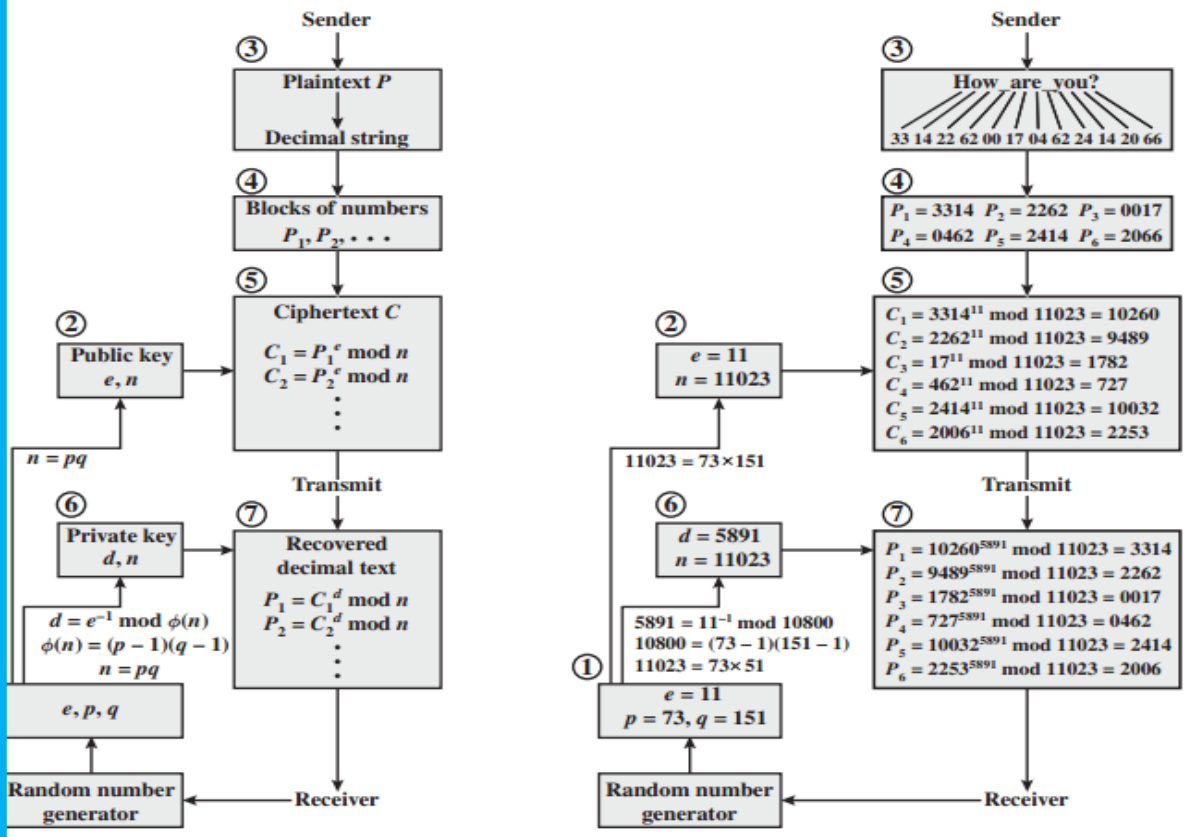


Figure II.13 Cryptage à plusieurs blocs

II.4. Discussion

Nous avons présenté dans ce chapitre, la structure détaillée des algorithmes AES et RSA. Nous avons aussi expliqué comment obtenir la clef étendue à partir d'une clef de longueur 128 bits pour l'algorithme AES. Et nous avons défini les étapes à suivre pour générer la paire de clefs privée-publique avec l'algorithme RSA. L'utilisation de ces deux algorithmes est avantageuse, car ils consomment peu de mémoire, et leurs complexité est moindre et sont plus faciles à implémenter. Ils sont très rapides et n'ont pas été cassés jusqu'à nos jours.

Le seul inconvénient de l'AES réside dans sa structure algébrique simple, et pour l'RSA, son utilisation avec des clefs de chiffrement 1024 bits est peu sure. De se fait, Il est très important de comprendre le fonctionnement de ces deux derniers algorithmes, dans le prochain chapitre nous les utiliserons pour la transmission sécurisée des images Météosat du satellite MSG. L'algorithme AES sera utilisée pour assurer la confidentialité des images, et l'algorithme RSA assurera l'échange des clefs d'une façon sécurisée, il assure aussi la fonction d'authentification.

Chapitre III

Transmission sécurisée des images MSG

III.1. Préambule

Nous avons expliqué dans le chapitre précédent que l'algorithme AES est définit par le NIST pour chiffrer les données de taille 128 bits. Les images MSG (Meteosat Second Generation) sont des images de grande quantité qui dépasse 128 bits, le NIST a défini cinq modes de chiffrement pour les algorithmes par bloc, ces modes permettront de chiffrer nos images MSG. Dans ce chapitre, nous détaillons les différentes étapes de notre système de transmission sécurisée des images Météosat. Nous décrivons la base d'images Météosat utilisée. Ensuite, nous donnons les cinq modes de chiffrement définis par le NIST qui seront appliqués pour chiffrer la base d'images Météosat.

III.2. Les images Météosat [1]

Les images satellitaires sont des mesures de rayonnement solaires réfléchis ou émis par la surface terrestre et les nuages, ou absorbées par l'atmosphère, dans différentes bandes spectrales. Ces images sont générées par différents satellites, nous citons parmi eux le satellite Météosat MSG (Meteosat Second Generation).

III.2.1. Caractéristique des images satellitaire [1]

Il est important, en télédétection, de distinguer les termes "image" et "photographie". Une image est une représentation graphique, quelle que soit la longueur d'onde ou le dispositif de télédétection qui a été utilisée pour capter et enregistrer l'énergie électromagnétique. Une photographie désigne spécifiquement toute image captée et enregistrée sur une pellicule photographique. Les photographies enregistrent habituellement les longueurs d'onde entre 0,3 et 0,9 mm (les portions visibles et infrarouges réfléchis).

Une photographie peut être présentée et affichée au format numérique en divisant l'image en petits morceaux de taille et de forme égale, que nous nommons pixels. La luminosité de chaque pixel est représentée par une valeur numérique. C'est exactement ce qui a été fait à la photographie au dessous. En effet, en appliquant les définitions présentées plus haut, nous déduisons que l'image est vraiment une image numérique de la photographie originale ! Cette photographie a été numérisée et subdivisée en pixels.

Comme montre la figure III.1 chaque pixel a été doté d'une valeur représentant les différents niveaux de luminosité. L'ordinateur affiche chaque valeur numérique comme un niveau de luminosité. Les capteurs enregistrent alors électroniquement l'énergie en format numérique (en rangées de chiffres). Ces deux différentes façons de représenter et d'afficher les données de télédétection, par des moyens photographiques ou numériques, sont interchangeables car elles représentent la même information (mais chaque conversion peut engendrer une perte de précision).

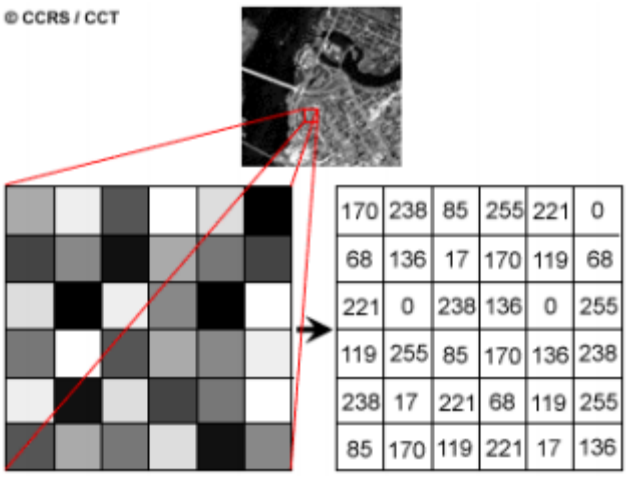


Figure III.1 Représentation d'une image numérique

III.2.2. Le satellite Météosat Second Génération (MSG) [23]

Près de vingt-cinq ans après le lancement du tout premier satellite Meteosat, en novembre 1977, le premier exemplaire de la seconde génération de satellites européens de veille météorologique MSG-1 a été mis en orbite le 28 août 2002 par le lanceur européen Ariane 5 qui a décollé du Centre Spatial de Kourou, en Guyane française. Ce satellite devenu opérationnel le 9 janvier 2004 et prend alors le nom de Meteosat8.

Puis c'est au tour de MSG-2 (Figure III.2) d'être lancé le 22 décembre 2005, avant d'être déclaré en service en juillet 2006 sous le nom deMeteosat9. Le lancement des deux derniers satellites du programme Météosat Seconde Génération est prévu en 2012 pour MSG-3 et en 2014 pour MSG-4.

Construits par Alcatel Space Industries avec la participation d'une équipe de plus de 50 industriels européens, ces satellites sont développés par l'Agence Spatiale Européenne (ESA) pour le compte d'EUMETSAT, organisation européenne d'exploitation des satellites météorologiques. Deux fois et demi plus gros que leurs prédécesseurs, ce sont des satellites cylindriques de 3,22 mètres de diamètre pour 3,74 mètres de haut. Au décollage, leur masse atteint 2 tonnes, dont près de la moitié constituée par les ergols nécessaires à leur mise et leur maintien à poste pendant les 7 années que dure en théorie leur mission. Les satellites MSG, équipés de nouveaux capteurs plus puissants et plus précis pour l'observation en continu de l'atmosphère terrestre, vont engranger jusqu'à l'horizon 2015-2020 une multitude de données indispensables à la compréhension et à la modélisation des activités climatiques de notre planète. La veille météorologique menée avec succès par les satellites Météosat depuis plus de 30 ans se poursuit.

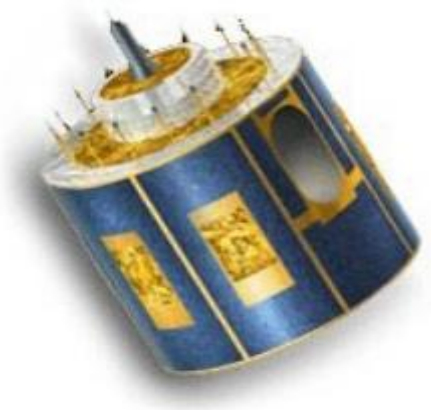


Figure III.2 Le satellite MSG

III.2.2.1. Le radiomètre SEVIRI

L'imageur SEVIRI (Spinning Enhanced Visible & InfraRed Imager) est capable de fournir tous les quarts d'heure (au lieu d'1/2 heure avec Météosat) une image observée par le satellite dans 12 bandes de fréquence différentes du spectre visible et infrarouge, soit 4 fois plus que Meteosat.

De plus, en réduisant de 30 à 15 minutes le rafraîchissement des données, les satellites MSG permettent aux prévisionnistes de déceler plus facilement le déclenchement des phénomènes météorologiques à évolution rapide, comme les orages ou les tempêtes de neige. Enfin, la résolution des canaux infrarouges passe de 5 km à 3 km tandis que l'un des nouveaux canaux visibles fournit des images de 1 km de résolution au lieu des 2,5 km de la première génération. Ces satellites emportent également une charge utile pour la collecte et la retransmission, quasiment en temps réel, d'observations recueillies par des stations automatiques au sol.

III.2.2.2. Les 12 canaux de SEVIRI [24] [25]

Pour chaque pixel, le radiomètre SEVIRI mesure l'énergie radiative dans différentes bandes spectrales. Ces bandes sont la bande visible, vapeur d'eau et l'infrarouge thermique. Le positionnement des 12 canaux de SEVIRI dans ces trois bandes est résumé par la figure suivante :

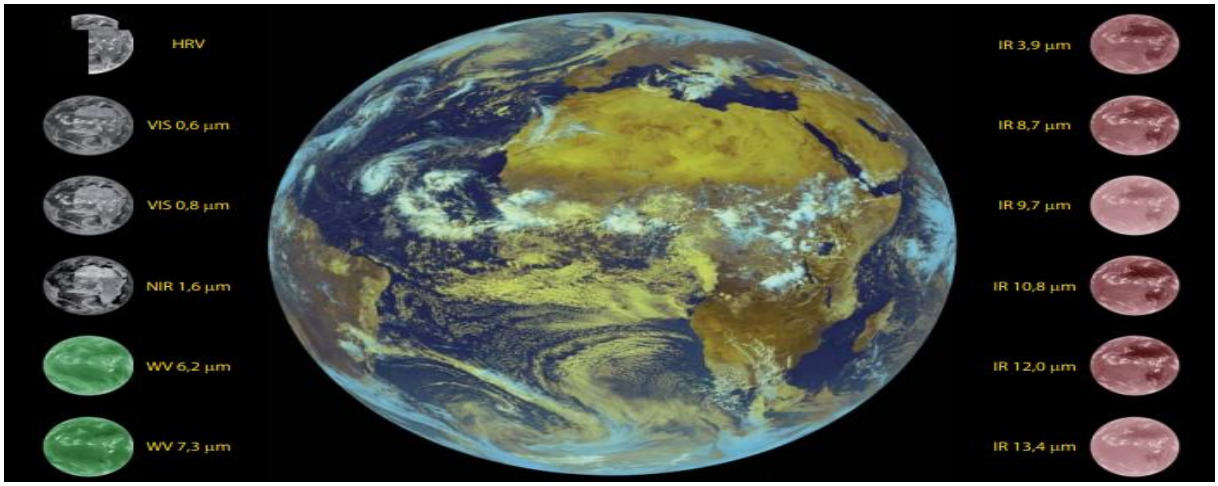


Figure III.3 Les 12 canaux de satellite MSG

canal	utilisation
HRV, 0,6 et 0,8 µm	Détection, identification et évolution des nuages, observation des aérosols, suivis de la végétation.
1,6 µm	Différenciation entre neige et nuage, nuages de glace et d'eau liquide, information sur les aérosols.
3,9 µm	Détection des nuages bas de nuit, des feux de jour.
6,2 et 7,3 µm	Vapeur d'eau de la moyenne et haute troposphère, suivi de la dynamique atmosphérique, hauteur des nuages semi transparents.
8,7 µm	Informations quantitatives sur les cirrus fins, distinction entre les nuages de glace et d'eau liquide.
9,7 µm	Radiance de l'ozone pour assimilation en prévision numérique, évolution du champ total d'ozone.
10,8 et 12 µm	Mesure de la température de surface de la terre et de la mer, détection des cirrus et déduction des quantités d'eau précipitable au-dessus de la mer.
13,4 µm	Amélioration de la détermination du facteur de transmission des cirrus, information sur la température de la basse troposphère dépourvue de nuages pour les évaluations d'instabilité.

Tableau III.1 Les 12 canaux de satellite MSG

III.2.2.2.1. Les canaux visibles (VIS 0.6, VIS 0.8 et HRV)

Les images visibles représentent la quantité de lumière visible rétrodiffusée par les nuages ou la surface de la terre. Les nuages et la neige apparaissent en blanc et les zones sans nuages en noir. Les nuages épais sont plus brillants que les nuages fins. Il est difficile de distinguer les nuages bas des nuages élevés. Pour cela, il faut utiliser les images infrarouges. Les images visibles sont complètements noires la nuit et ne peuvent ainsi être utilisées.

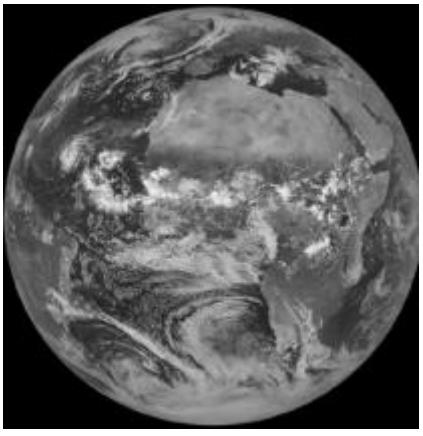


Figure III.4 image prise dans le canal visible HRV à droite et 0,6 µm à gauche

III.2.2.2.2. Les canaux infrarouges (3.9, 8.7, 10.8, 12.0 et 13.4)

Les images infrarouges (Figure III.5) représentent une mesure de rayonnement infrarouge émis par le sol ou les nuages. Ce rayonnement dépend de la température. En mode inversé, plus l'objet est chaud, plus il est noir et plus l'objet est froid, plus il est blanc comme montre la figure. Les nuages élevés apparaissent plus blanc que les nuages bas car ils sont plus froids. Dans les zones sans nuages, plus le sol est chaud, plus il est sombre.

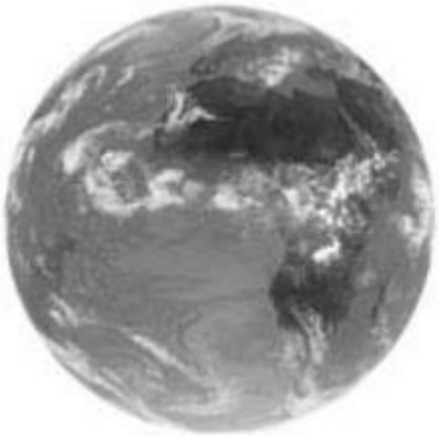


Figure III.5 image prise dans l'IR 10,8 µm

III.2.2.2.3. Les canaux vapeur d'eau (WV 6.2 et 7.3 um)

Les images vapeur d'eau (Figure III.6) représentent une mesure du rayonnement infrarouge influencée par la vapeur d'eau dans l'atmosphère. Cela permet de déterminer les zones sèches et les zones humides. Lorsque l'atmosphère est pauvre en vapeur d'eau, les rayons infrarouge la traversent et parviennent au capteur WV de Météosat Seconde Génération. Au contraire plus l'atmosphère est chargée de vapeur d'eau moins ils la traversent.

Figure III.6 image prise dans le canal vapeur d'eau 6,2 µm à droite et 7,3 µm à gauche

III.2.2.3. Collecte et transmission des images MSG [26]

Le radiomètre imageur SEVIRI balaie la surface terrestre ligne par ligne d'est en ouest d'une manière à produire une nouvelle image multi-spectrale toutes les 15 mn. L'acquisition des images est assurée en combinant la rotation de satellite auteur de son axe et celle du

miroir de balayage correspondant à 18 degré. Il est décrit en 30 ms. Pendant les 570 ms suivantes, le télescope vise l'espace et cette durée est mise à profit pour modifier l'orientation du miroir de façon qu'au tour suivant, il balaie au sol une bande contiguë à la précédente mais plus au nord de 3 Kms (voir Figure III.7). Le balayage d'une ligne d'est en ouest est assuré par la rotation du satellite. Le balayage du sud au nord est réalisé par un mouvement pas à pas d'un miroir de balayage couvrant le disque terrestre d'environ 1250 tours ; ceci fournit 3750 lignes d'image pour les 11 canaux disposant 3 détecteurs chacun. Pour le HRV, 9 détecteurs sont utilisés pour le balayage d'une ligne. Pour fournir une image plein disque, le télescope balaie la totalité de la surface de la terre en 12 minutes 30 s. les 2 mn 30s suivantes sont consacrées au retour du miroir à sa position initiale, et il reprend le balayage de l'image suivante. La phase de non acquisition des images est consacrée au calibrage des canaux infrarouges par référence au rayonnement du corps noir inséré dans le chemin optique de télescope. Chaque 15 mn, le satellite produit une image de 3712 lignes et de 3712 pixels codée sur 10 bits dans les 11 canaux, et une image de 11136×5568 pixels pour le HRV.

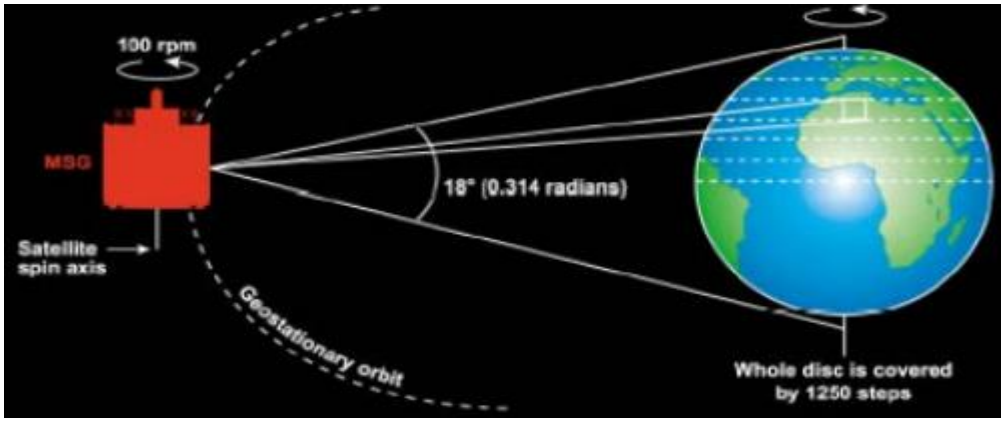


Figure III.7 Principe de prise l'mage par le radiomètre SEVIRI

Les images brutes collectées par le satellite MSG sont des images de niveau -1. À qui se transforment en images de niveau 1.5, après traitement et correction géométrique de toutes les perturbations introduites par le satellite.

Le satellite MSG reçoit les données envoyées par la plateforme de collection de données (DCP). Puis, il les transmet à la station au sol (PGS) situé à Using en Allemagne. Ces données sont traitées au centre de contrôle spécifique (MCC) au siège d'Eumetsat à Darmstad en Allemagne, puis retransmises aux utilisateurs, soit via le système mondial de télécommunication (GTS) de l'organisation mondiale de la météorologie (OMM), soit via Eumetcast, le système de distribution de données d'EUMETSAT, ou par le service LRIT (Low Rate Information Transmission). Le service LRIT transmet des données de faible débit (128kb/s) aux stations de réception de faible débit LRUS (Low Rate User Station) qui reçoivent trois à cinq images dans les canaux de SERIVI, par cycle de 30 minutes dans les 15 minutes qui suivent l'observation.

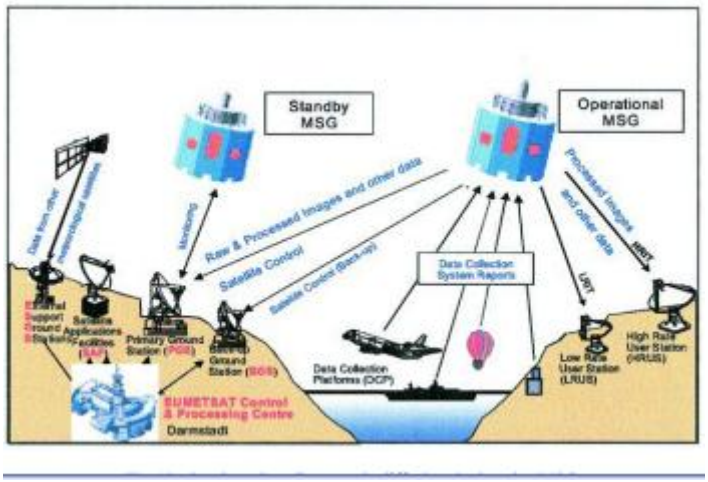


Figure III.8 Système de collecte et de diffusion de données MSG

III.2.2.4. Le segment sol de MSG [23] [27]

Le segment sol est nécessaire au contrôle et à la commande des satellites et à la réception des données. Il est constitué d'un centre de contrôle spécifique (MCC), d'une station sol principale(PGS), d'une station de secours et d'orbitographie à Maspalomas en Espagne et d'une deuxième station de réserve à Cheia, en Roumanie, d'un centre d'extraction de produits météorologique (MPEF), d'un centre d'archivage et de consultation de produits météorologiques (U-MARF) installés à Darmstadt et d'un réseau décentralisé de centres d'applications satellitaires (SAF) (voir Figure III.9)

Le réseau SAF a été développé par EUMETSAT à travers l'Europe afin de fournir des outils logiciels, produits et données spécialisés à diverses communautés d'utilisateurs. Il existe huit SAFs qui sont :

- SAF OSI (SAF on OceanabdSeaIce): il produit et dissémine des produits caractérisant la surface des océans et les flux d'énergie à travers la surface de la mer.
- SAF O_3M (SAF on Ozone Monitoring) : il est développé pour le traitement des données de l'Ozone, des aérosols et le rayonnement ultra-violet estimé par des observations satellitaires.
- SAF NWP (SAF on Numerical Weather Prediction) : ce SAF vise à augmenter les avantages aux divers centres météorologiques européens de NWP en développant des technique avancées pour l'usage efficace des données satellites.
- SAF CLM (SAF on Climate Monitoring): il produit et archive des ensembles de données de haute qualité pour des domaines d'application spécifiques du climat. Actuellement, il se concentre sur des paramètres de nuage et des produits de l'humidité atmosphérique.
- SAF LSA (SAF on Land Surface Analysis): ce SAF est consacré à l'analyse de la surface terresyre.
- SAF GRAS (SAF on Ground positioning system Receiver for Atmospheric Sounding): ce SAF est consacré aux mesures par radio occultation du satellite.
- SAF-H : est consacré pour l'hydrologie opérationnelle et la gestion de l'eau.

- SAF NWC : la prévision immédiate et à court terme.

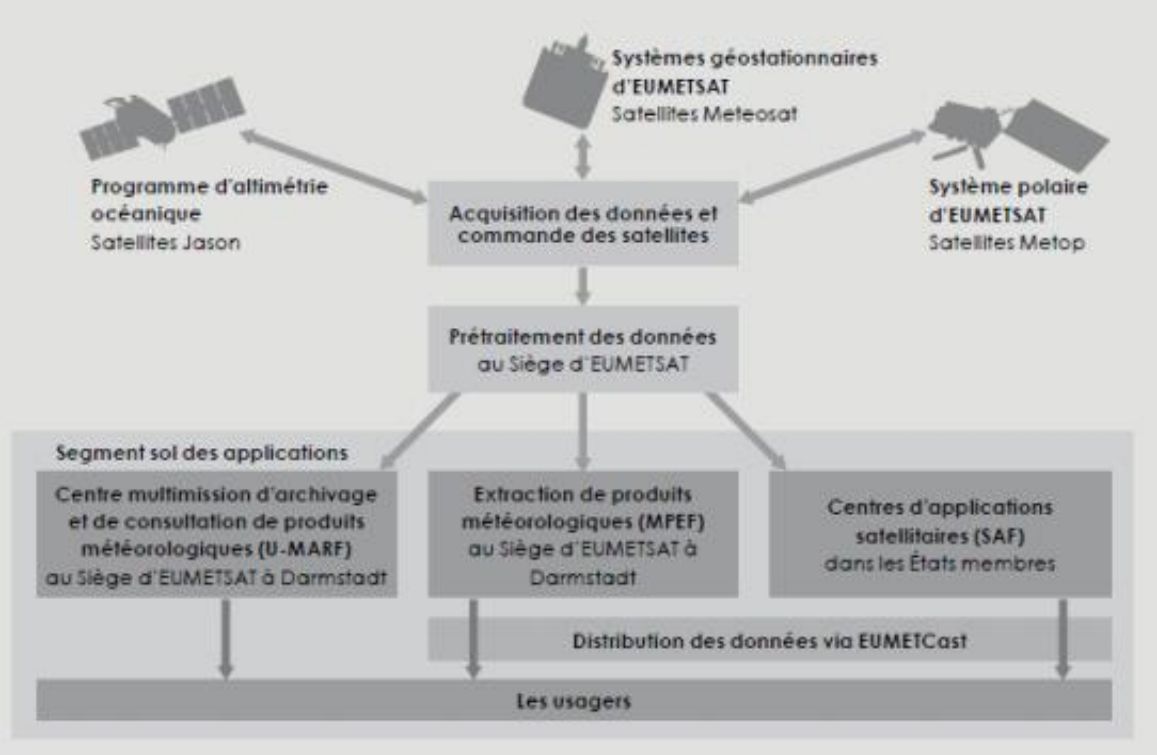


Figure III.9 Collecte et transmission des données MSG

III.3. Le cryptosystème de transmission

Le cryptosystème de transmission (Figure III.10) utilise un chiffrement hybride et présente deux blocs principaux. Un bloc d'émission et un autre bloc de réception. Notre cryptosystème a pour but de protéger l'image transmise entre les deux blocs à travers un canal de transmission de toute sorte d'attaque.

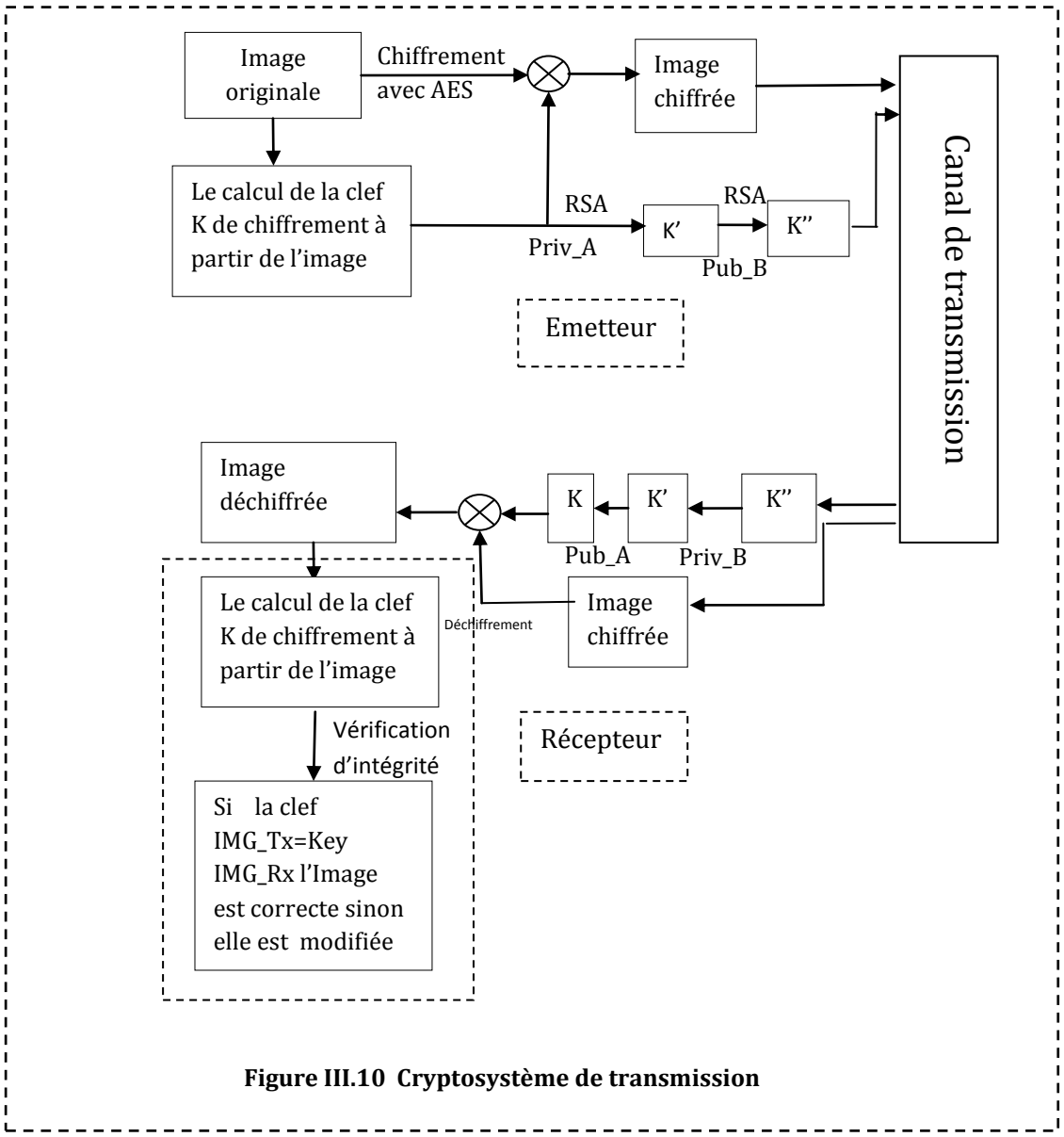


Figure III.10 Cryptosystème de transmission

III.3.1. Bloc de transmission

Dans le bloc d'émission trois opérations principales sont déroulé dans l'ordre suivant :

III.3.1.1. Extraire la clef de session K

La clef de session est calculé à partir de l'image originale MSG tout les 15 minute, le calcul de la clef est basé sur la fonction de corrélation en utilisant l'équation ci-dessous :

$$R_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \quad \text{III.7}$$

Où x et y sont les valeurs du niveau de gris des pixels des images. La covariance et la variance sont données par les équations suivantes :

$$E(x) = \frac{1}{L}\sum_{i=1}^{l} x_i \quad \text{III.8}$$

$$D(x) = \frac{1}{L}\sum_{i=1}^{l}(x_i - E(x))^2 \quad \text{III.9}$$

$$cov(x,y) = \frac{1}{L}\sum_{i=1}^{l} E(x_i - E(x))(y_i - E(y)) \quad \text{III.10}$$

Avec L est le nombre de pixels utilisés.

Nous pouvons résumer les étapes de calcul de la clef de session par l'algorithme suivant :

*** la clef de session ***

debut

$x \longleftarrow$ image originale MSG

$x_1 \longleftarrow x(:,1:end-1)$

$x_2 \longleftarrow x(:,2:end)$

$clef \longleftarrow R_{x1x2}$

$k \longleftarrow clef$

fin

III.3.1.2. Chiffrement des images MSG

Une fois la première étape terminée et l'extraction de la clef de session fini, nous passons à l'étape de chiffrement de l'image MSG avec l'algorithme AES, nous choisissons un mode parmi les cinq mode pour la transmission de l'image MSG, nous représentons ci-dessous les cinq algorithmes de mode de chiffrement :

*** mode ECB ***

debut

$x \longleftarrow$ *image originale MSG*
$[n, m] \longleftarrow$ *la taille de* x
$blocks \longleftarrow (n \times m) \div 16$
$k \longleftarrow$ *lecteur la clef de session*
$y \longleftarrow zeros(n, m)$
$idx \longleftarrow 1{:}16$
pour $i = 1 : blocks$

$in \longleftarrow x(idx)$
$y(idx) \longleftarrow AESencryption(aesinit(k), in)$
$idx = idx + 16$
fin

afficher l'image chiffrée

*** mode 0FB ***

debut

$x \longleftarrow$ *image originale MSG*

$[n, m] \longleftarrow$ *la taille de* x

$blocks \longleftarrow (n \times m) \div 16$

$k \longleftarrow$ *entrer la clef de session*

$IV \longleftarrow$ *entrer la vecteur d'initiation*

$y \longleftarrow zeros(n, m)$

$idx \longleftarrow 1{:}16$

pour $i = 1 : blocks$

$VI \longleftarrow AESencryption(aesinit(k), VI)$

$in \longleftarrow x(idx)$

$y(idx) \longleftarrow in \oplus VI$

$idx = idx + 16$

fin

afficher l'image chiffrée

*** mode CBC ***

debut

$x \longleftarrow$ *image originale MSG*
$[n, m] \longleftarrow$ *la taille de* x
$blocks \longleftarrow (n \times m) \div 16$
$k \longleftarrow$ *entrer la clef de session*
$IV \longleftarrow$ *entrer la vecteur d'initiation*
$y \longleftarrow zeros(n, m)$
$idx \longleftarrow 1{:}16$
pour $i = 1 : blocks$
$in \longleftarrow x(idx)$
$VI \longleftarrow in \oplus VI$
$VI \longleftarrow AESencryption(aesinit(k), VI)$
$y(idx) \longleftarrow VI$
$idx = idx + 16$
fin
afficher l'image chiffrée
fin

*** mode CTR ***

debut

$x \longleftarrow$ *image originale MSG*
$[n, m] \longleftarrow$ *la taille de* x
$blocks \longleftarrow (n \times m) \div 16$
$k \longleftarrow$ *entrer la clef de session*
$ctr \longleftarrow$ *entrer la valeur de compteur*
$y \longleftarrow zeros(n, m)$
$idx \longleftarrow 1{:}16$
pour $i = 1 : blocks$
$ctr \longleftarrow AESencryption(aesinit(k), ctr)$
$in \longleftarrow x(idx)$
$y(idx) \longleftarrow in \oplus VI$
$ctr(16) \quad mod(i, 255)$
$idx = idx + 16$
fin
afficher l'image chiffrée
fin

```
*** mode CFB ***

debut

x ⟵ image originale MSG
[n, m] ⟵ la taille de x
blocks ⟵ (n × m) ÷ 16
k ⟵ entrer la clef de session
IV ⟵ entrer la vecteur d'initiation
y ⟵ zeros(n, m)
idx ⟵ 1:16
pour i = 1 : blocks

                    VI ⟵ AESencryption(aesinit(k), VI)
VI ⟵ VI(S_bits)
                    in ⟵ x(idx)
                    in ⟵ in(S_bits)
                    VI ⟵ in ⊕ VI
y(idx) ⟵ VI
                    y ⟵ décaler y vers la gauche avec 128 − S_bits
VI ⟵ y
                  idx = idx + 16
fin

afficher l'image chiffrée
```

III.3.1.3. Chiffrement de la clef de session K

Afin de transmettre la clef K d'une manière sécurisée, et pour garantir l'authentification, l'émetteur peut chiffrer cette clef en utilisant l'algorithme asymétrique RSA.

L'émetteur à ces clefs publiques et privées, $Pub_E(b_x,n_x)$, $Priv_E(u_x,n_x)$, et le récepteur à ces clefs publiques et privées, $Pub_R(b_y,n_y)$, $Priv_R(u_y,n_y)$.

L'émetteur signe la clef K avec l'algorithme RSA en utilisant la clef privée de l'émetteur $Priv_E$ afin d'obtenir une clef signé K' telle que :

$$K' = K^{Ux} \bmod (n_x) \qquad \text{III.11}$$

Cette clef K' est cryptée une second fois avec RSA en utilisant la clef publique de récepteur Pub afin de générer la clef K'' :

$$K'' = K'^{by} \bmod (n_y) \qquad \text{III.12}$$

Cette procédure de chiffrement de la clé K avec la clef privée de l'émetteur $Priv_E$ assure l'authenticité, et seul le récepteur peut déchiffrer l'image envoyée.

III.3.2. Bloc de réception

Des fonctions inverses sont utilisées pour reconstituer la même image envoyée dans l'ordre suivant :

- ✓ Déchiffrement de la clef de session K avec l'algorithme RSA.
- ✓ Déchiffrement de l'image avec la clef K.
- ✓ Vérification de l'intégrité.

La fonction de corrélation entre les pixels voisins est utilisée pour vérifier l'intégrité.

III.3.2.1. Vérification de l'intégrité

Avec les cinq modes, nous ne pouvons pas assurer l'intégrité des images envoyées.

Prenons l'exemple des images de la figure III.11. A partir de l'image originale, figure (a), nous avons appliqué l'algorithme AES par bloc (mode ECB) avec une clef de 128 bits afin d'obtenir l'image cryptée figure (b). Si l'image cryptée est modifiée durant le transfert il n'est pas forcément possible de détecter la modification. Par exemple, dans la figure (c) nous avons permuté les quatre régions (modulo 128 bits) de l'image et dans la figure (d) on a copié une petite région de l'image cryptée et nous avons collé cette région sur une autre zone de l'image. Après décryptage, il est possible de visualiser les images mais il n'est pas possible de garantir l'intégrité comme illustrée dans la figure (f).

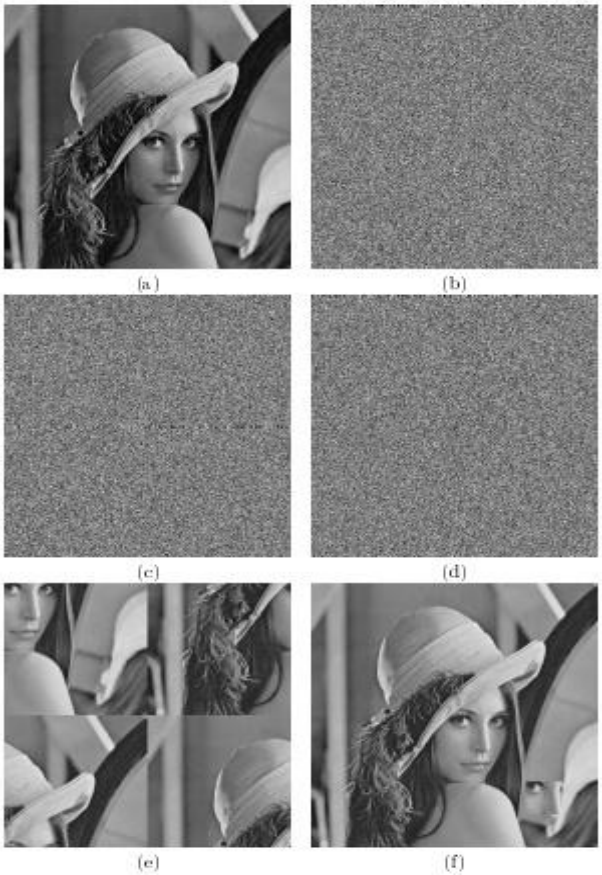


Figure III.11 Problème d'intégrité

Dans notre cryptosystème pour résoudre ce problème, nous proposons d'utiliser la fonction d'autocréation, Nous avons remarqué que chaque image produit une empreinte cryptographique différente. Pour cette raison, nous avons décidé d'exploiter cette propriété pour générer et vérifier l'intégrité de l'image à la réception.

III.4. Modes de chiffrement [4] [12]

Un algorithme de chiffrement par bloc prend des blocs de longueur fixe de bits b et une clef a pour produire des textes chiffrés de longueur b. Lorsque plusieurs blocs de texte en clair sont cryptés en utilisant la même clef, un certain nombre de problèmes de sécurité se pose. Pour appliquer un chiffrement par bloc dans une variété d'applications, cinq modes de fonctionnement ont été définis par le NIST (SP 800-38A). Un mode de fonctionnement est une technique visant a accroitre l'effet d'un algorithme de cryptage ou l'adapte pour une application, dans notre cas c'est la transmission des images MSG, Ces modes sont destinés à être utilisés avec n'importe quel algorithme de chiffrement symétrique, y compris triple DES et AES.

Les cinq modes sont résumés dans le tableau ci-dessous et décrites dans les sections suivantes.

Mode	Description
Le mode « ElectronicCodeBook » (ECB)	Il revient à chiffrer un bloc indépendamment des autres, cela permet entre autre de chiffrer suivant un ordre aléatoire (bases de données, etc.)
Le mode « Cipher Block Chaining» (CBC)	Il permet d'introduire une complexité supplémentaire dans le processus de chiffrement en créant une dépendance entre des blocs successifs. Il effectue un XOR entre un bloc de données en clair et un bloc de données cryptées. Quand au premier bloc il est XORé avec un vecteur appelé vecteur d'initialisation (IV) qui peut être un mot de passe par exemple, ce vecteur change à chaque session, et doit être transmis au destinataire.
Le mode « CipherFeedBack » (CFB)	Le message est ajouté par un XOR à la sortie du bloc chiffré. Le résultat sert d'entrée pour l'étape suivante. Il est utilisé pour le chiffrement par flux.
Le mode « Output Feedback » (OFB)	Une variante du CFB. La différence ici, c'est que le flux entrant vers les étapes ultérieures est indépendant du message clair.
Le Mode « Compteur » (CTR)	Chaque bloc est XORé avec compteur chiffré qui incrémente chaque séquence de chiffrement.

Tableau III.2 les cinq modes de chiffrement

III.4.1. Le mode « ElectronicCodeBook (ECB) »

Le mode le plus simple est le mode Electronic Code Book, dans lequel chaque bloc est chiffré avec la même clef. A la réception les blocs chiffrés sont déchiffrés avec la même clef comme la figure le montre. Le texte clair est devisé en blocs :

$$P_1, P_2 \ \ldots\ldots\ldots\ldots P_N$$

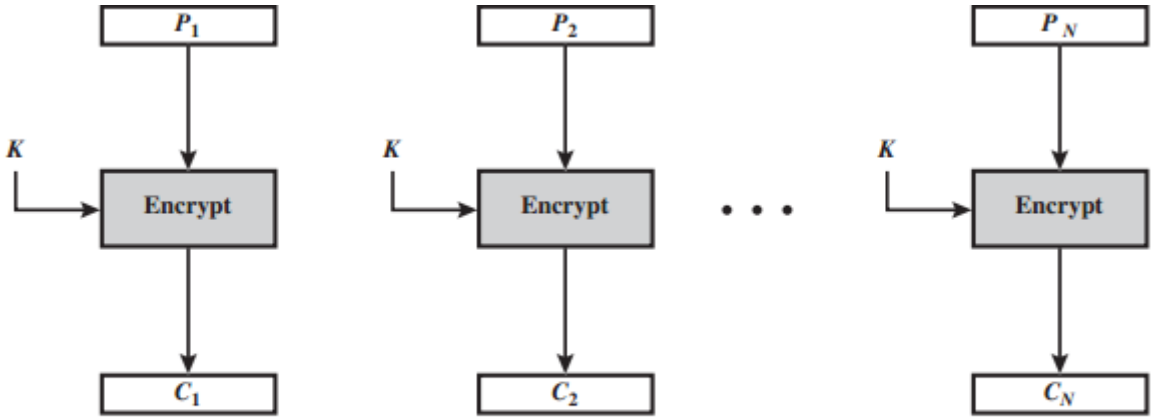
Les blocs chiffrés correspondants sont :

$$C_1, C_2 \ \ldots\ldots\ldots\ldots C_N$$

Nous pouvons définir le mode ECB comme suit :

ECB	$C_j = E(K, P_j)$	$j = 1, \ldots\ldots, N$	$P_j = D(K, C_j)$ $j = 1, \ldots\ldots, N$

III.1



(a) Encryption

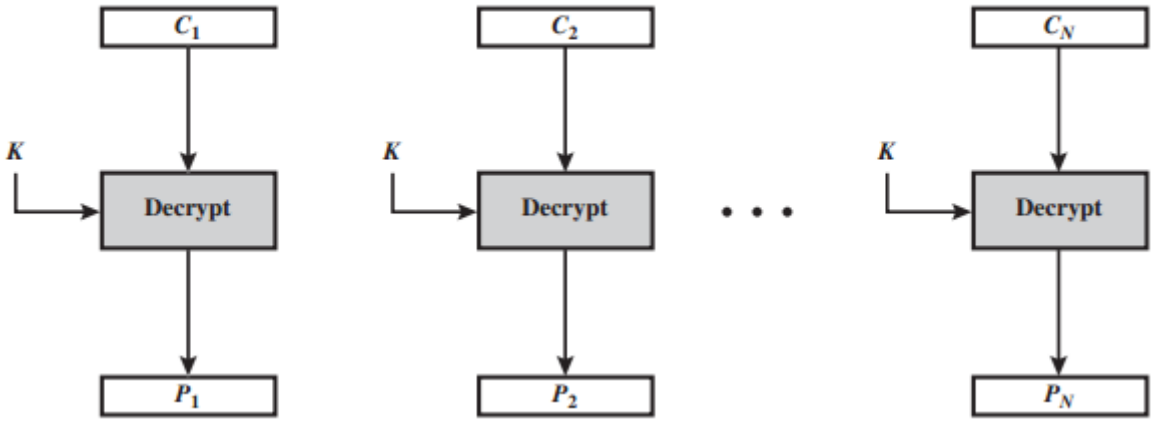


Figure III.12 Mode ECB

Le mode ECB est idéal pour la transmission des données courtes, tel qu'un chiffrement clef. Ainsi, si vous souhaitez transmettre une clef DES ou AES en toute sécurité, ECB est le mode appropriés à utiliser.

La caractéristique la plus importante de ECB est que le même bloc de bits de texte en clair apparaît plus d'une fois dans le message, il produit toujours le même texte chiffré.

Pour de longs messages, le mode ECB est peu sécurisé. il peut être possible pour un cryptographe à exploiter la propriété de mode ECB que les mêmes blocs sont chiffrés de la même manière.

III.4.2. Le mode Cipher Block Chaining (CBC)

Pour résoudre le problème de chiffrement des blocs qui se chiffrent de la même manière avec le mode ECB, nous passons au mode CBC (Figure III.13). Dans ce cryptosystème, l'entrée à l'algorithme de chiffrement est le « OU exclusif » du bloc actuel de texte en clair et le bloc de texte chiffré précédent; la même clé est utilisée pour chaque bloc.

Pour le déchiffrement, chaque bloc de chiffrement est déchiffré par l'algorithme de déchiffrement.

Le résultat est XORed avec le bloc de texte chiffré précédent pour produire le bloc de texte en clair.
Pour bien illustrer le fonctionnement, nous pouvons écrire :

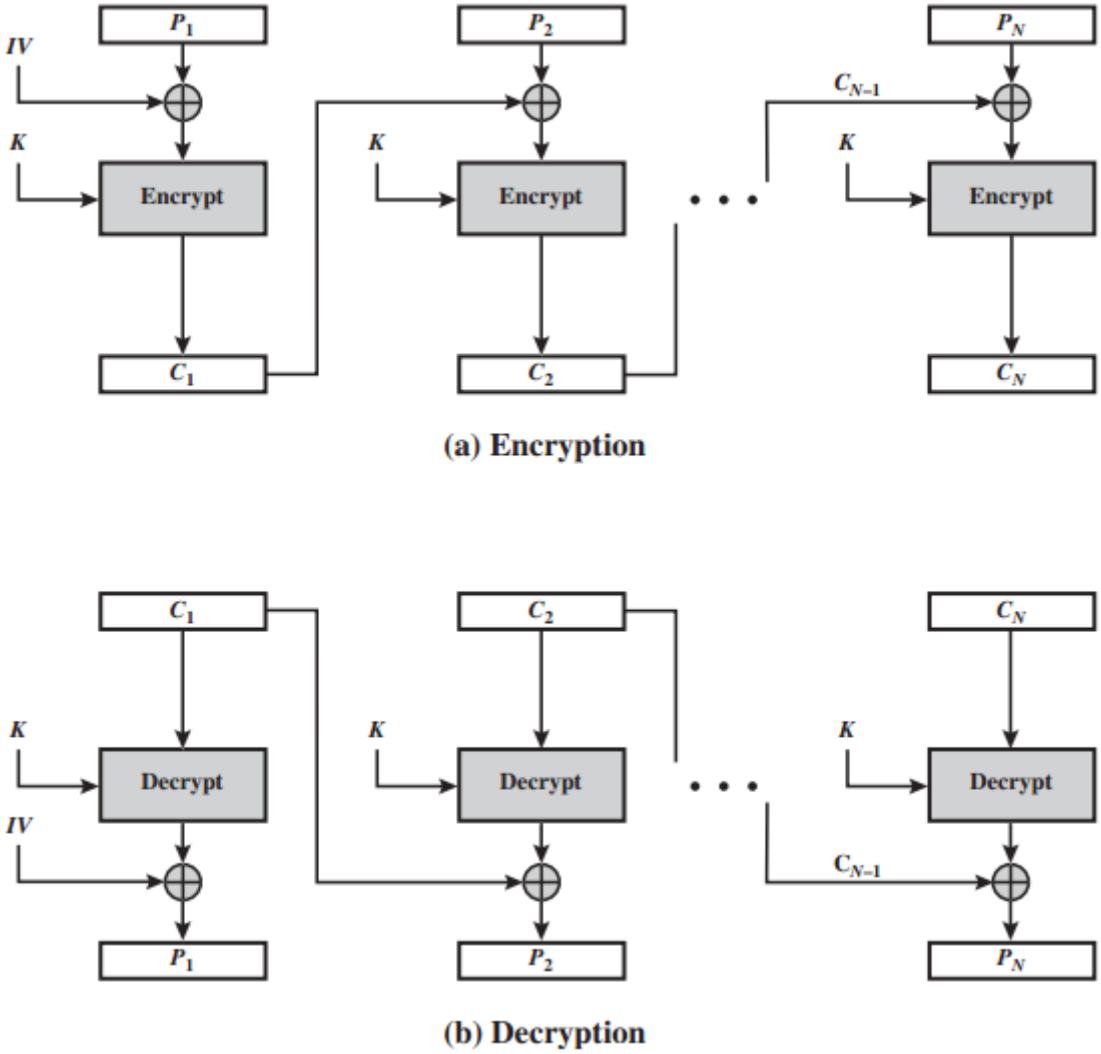


Figure III.13 mode CBC

Après nous écrivons :

$$C_j = E(K, [C_{j-1} \oplus P_j])$$

$$D(K, C_j) = D\left(K, E(K, [C_{j-1} \oplus P_j])\right)$$

$$D(K, C_j) = C_{j-1} \oplus P_j$$

$$C_{j-1} \oplus D(K, C_j) = C_{j-1} \oplus C_{j-1} \oplus P_j = P_j \quad \text{III.2}$$

Pour produire le premier bloc de texte chiffré, un vecteur d'initialisation (IV) est XORed

Avec le premier bloc de texte en clair. pour le déchiffrement, la IV est XORed avec la sortie de l'algorithme de déchiffrement pour récupérer le premier bloc de texte en clair, L'IV est un bloc de données qui est de la même taille que le bloc de chiffrement. Nous pouvons définir le mode CBC comme de suit :

CBC	$C_1 = E(K, [P_1 \oplus IV])$ $C_j = E(K, [P_1 \oplus C_{j-1}])j = 2, \dots, N$	$P_1 = D(K, C_1) \oplus IV$ $P_j = D(K, C_j) \oplus C_{j-1} j = 2, \dots, N$

III.3

III.4.3. Le mode CipherFeedBack (CFB)

Il est possible de convertir un chiffrement par bloc à un chiffrement de flux, en utilisant l'un des trois modes discutés dans la section précédente et les deux sections suivantes :

CFB, OFB et CTR. Un flux de chiffrement élimine la nécessité de deviser un message à un nombre entier de blocs. Il peut aussi fonctionner en temps réel. Ainsi, si un flux de caractères est transmis, chaque caractère peut être chiffré et immédiatement transmis.

La figure III.14 décrit le cryptosystème de la CFB. Sur la figure, on suppose que l'unité de transmission est s bits; une valeur de S est 8. Ici le texte en clair est divisé en segments de S bits.

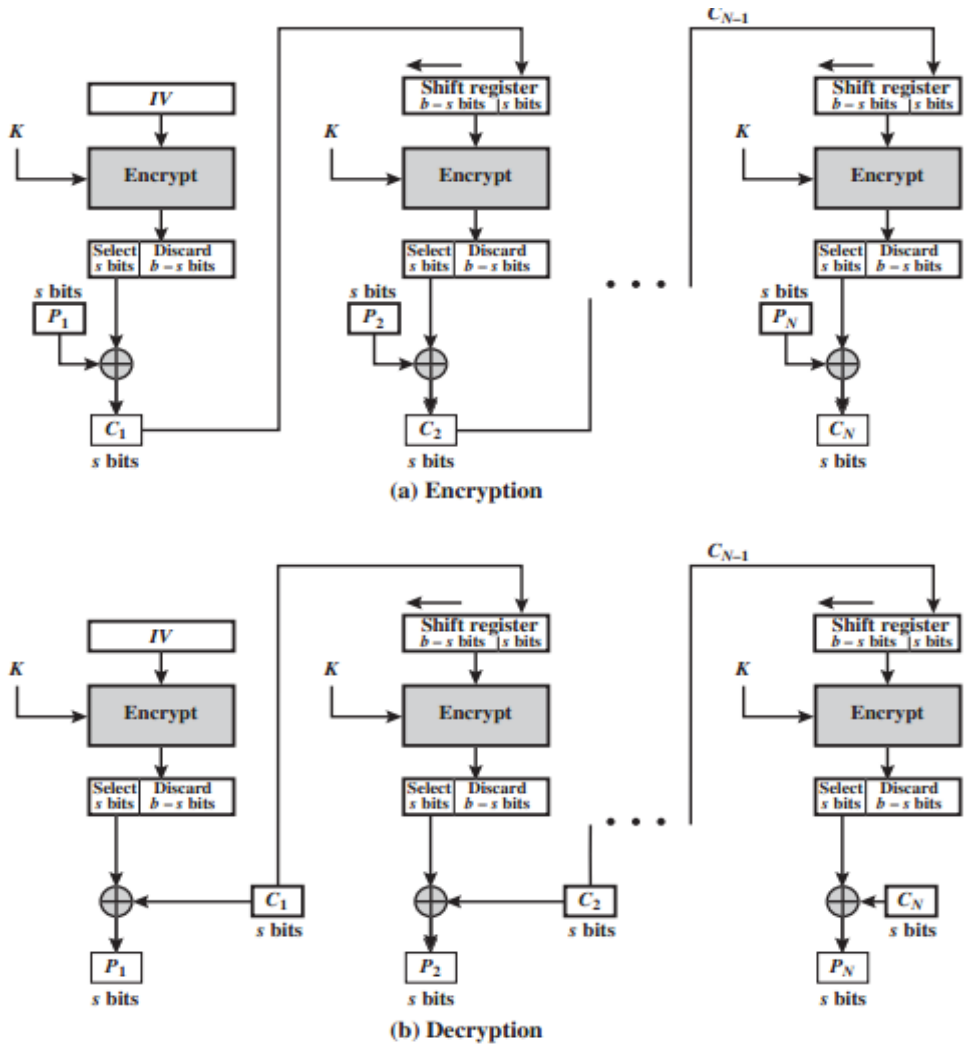


Figure III.14 Mode CFB

A partir de la figure nous pouvons résumer le fonctionnement du mode CFB comme suit :

CFB	$I_1 = IV$ $I_j = LSB_{b-s}(I_{j-1})//C_{j-1} j = 2, \ldots, N$ $I_j = E(K, I_j) j = 1, \ldots, N$ $C_j = P_j \oplus MSB_s(O_j) j = 1, \ldots, N$	$I_1 = IV$ $I_j = LSB_{b-s}(I_{j-1})//C_{j-1} j = 2, \ldots, N$ $O_j = E(K, I_j) j = 1, \ldots, N$ $P_j = C_j \oplus MSB_s(O_j) j = 1, \ldots, N$	III.4

III.4.4. Le mode « Output FeedbackOutput Feedback » (OFB)

La structure du mode OFB est similaire à celle de l'CFB, comme nous pouvons le constater à partir de la figure. C'est la sortie de la fonction de chiffrement qui est réinjecté au registre à décalage dans le mode OFB, alors que dans le mode OFB, le texte chiffré est celui qui est renvoyé au registre à décalage. L'autre différence est que le mode OFB opère sur des blocs complets de texte en clair et chiffré, et ne pas sur s-bit. Le chiffrement peut s'exprimer comme suit :

$$C_j = P_j \oplus E(K, [C_{j-1} \oplus P_{j-1}])$$

Et pour le déchiffrement on aura :

$$C_j = C_j \oplus E(K, [C_{j-1} \oplus P_{j-1}])$$

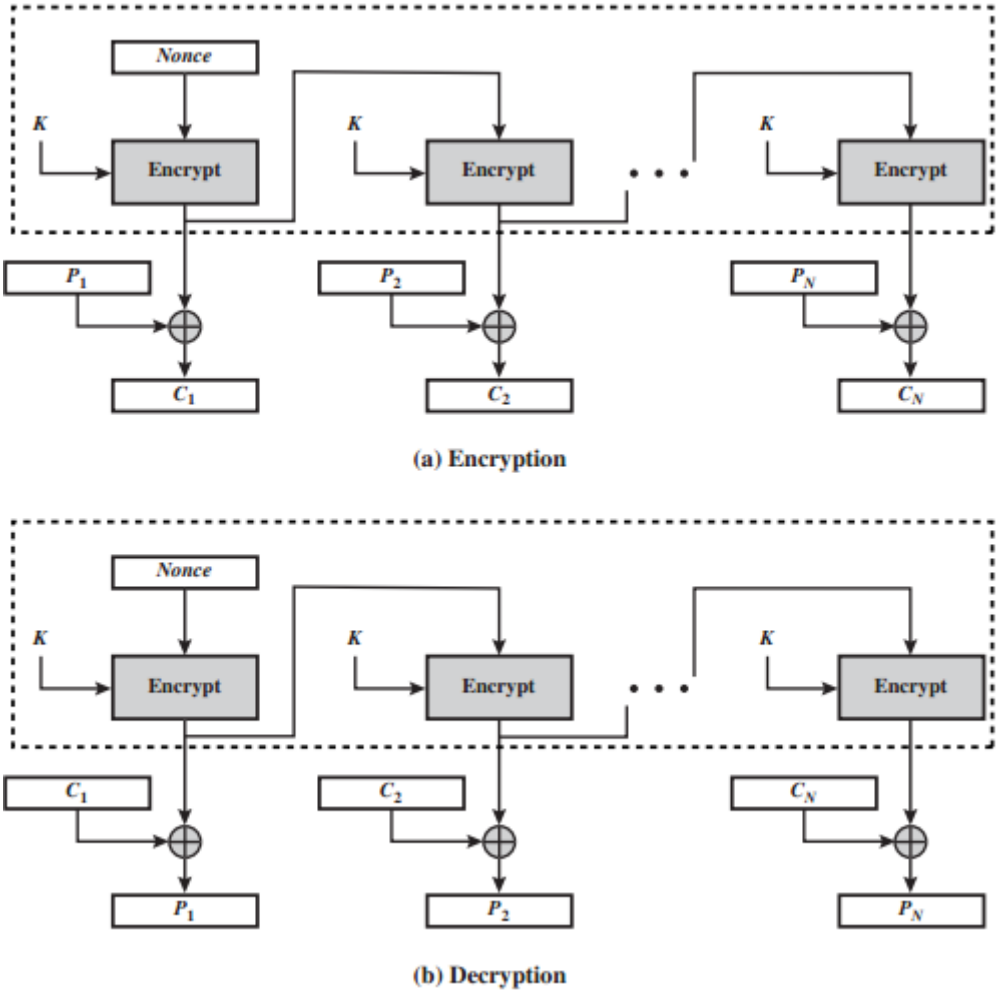


Figure III.15 mode OFB

Nous pouvons résumer le fonctionnement du mode CFB dans le tableau suivant :

OFB	$I_1 = Nonce$ $I_j = O_{j-1} j = 2, \dots, N$ $O_j = C_j \oplus E(K, I_j) \quad j = 1, \dots, N$ $O_j = E(K, I_j) \quad j = 1, \dots, N$ $C_j = P_j \oplus E(K, I_j) \quad j = 1, \dots, N-1$	$I_1 = Nonce$ $I_j = LSB_{b-s}(I_{j-1}) // C_{j-1} j = 2, \dots, N$ $O_j = C_j \oplus E(K, I_j) \quad j = 1, \dots, N$ $P_j = C_j \oplus O_j j = 1, \dots, N-1$ $P_N^* = C_N^* \oplus MSB_u(O_N)$	III.5

Comme avec les modes CBC et CFB, le mode OFB nécessite un vecteur d'initialisation. Dans le cas du mode OFB, la IV doit être un nonce; autrement dit, la IV doit être unique pour chaque exécution de l'opération de chiffrement. Un avantage du mode OFB est que l'erreur ne se propage pas, par exemple, si une erreur de bit se produit dans le bloc C_1 , seule la valeur récupérée P_1 après le déchiffrement est affecté.

III.4.5. Le mode compteur CTR

La figure suivante illustre le mode CTR. La taille du compteur est égal à la taille du bloc de texte en clair. Typiquement, le compteur est initialisé à une certaine valeur, puis incrémenté de 1 pour chaque bloc suivant. Pour le cryptage, le compteur est chiffré puis XORed avec le bloc de texte en clair pour produire le bloc de texte chiffré; il n'y a pas de chaînage. Pour le déchiffrement, la même séquence de valeurs de compteur est utilisée, chaque compteur chiffré XORed avec un bloc de texte chiffré pour récupérer le bloc de texte clair correspondant. Pour la séquence de compteurs $T_1, T_2, T_3, \dots \dots, T_N$, nous pouvons définir le mode CTR comme suit :

CTR	$C_j = P_j \oplus E(K, T_j) \quad j = 1, \dots, N-1$ $C_N^* = P_N^* \oplus MSB_u[E(K, T_N))]$	$P_j = C_j \oplus E(K, T_j) \quad j = 1, \dots, N-1$ $P_N^* = C_N^* \oplus MSB_u[E(K, T_N))]$	III.6

Comme dans le mode OFB, la valeur initiale du compteur T_1 doit être différente de l'ensemble des messages chiffrés en utilisant la même clef.

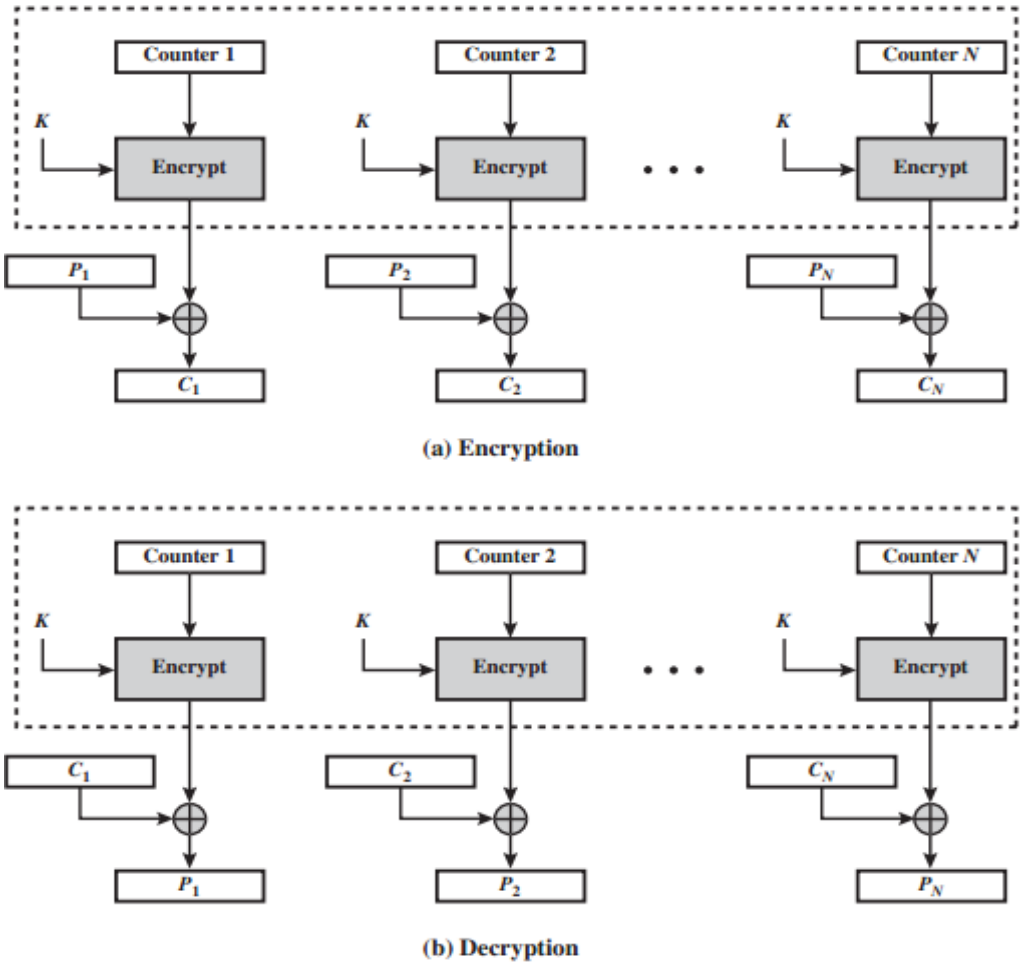


Figure III.16 Mode CTR

III.5. Discussion

Dans ce chapitre, nous avons décrits les caractéristiques du satellite MSG, puis nous avons détaillé les cinq modes de chiffrement, et dans la dernière partie nous avons présenté notre cryptosystème de transmission. Ce dernier est divisé en deux blocs : un bloc d'émission et un autre de réception. Ces deux derniers sont reliés à travers un canal de transmission. Dans le premier bloc, nous avons chiffré les images MSG avec les cinq modes, et généré, ensuite, une clef de session toutes les quinze minutes à partir de l'image MSG elle-même avec la fonction de corrélation, puis nous avons chiffré cette clef avec l'algorithme RSA deux fois successives pour assurer une transmission sécurisée de cette dernière, et garantir l'authenticité de l'émetteur.

Dans le bloc de réception des fonctions inverses, sont appliquées à l'image chiffrée pour la déchiffrer. Après le déchiffrement, nous avons pu vérifier l'intégrité et assuré que l'image n'a pas changé dans le canal de transmission.

Le chapitre suivant sera consacré à l'analyse de la sécurité de notre cryptosystème de transmission.

Chapitre IV

Résultats et discussion

IV.1. Préambule

Pour analyser la robustesse de la sécurité de notre cryptosystème de transmission, la simple inspection visuelle reste insuffisante pour juger le chiffrement d'une image. Pour notre analyse nous avons utilisé les métriques d'évaluation du degré de chiffrement suivantes :

- Analyse d'histogramme,
- La corrélation entre l'image originale et l'image chiffrée,
- La corrélation entre les pixels adjacents,
- Analyse de la sensibilité de la clef secrète,
- Analyse de la fonction de corrélation pour contrôler l'intégrité.

Dans nos différents tests, le nombre d'image que nous avons chiffré est de 12 images issues du satellite MSG dans différents canaux de type PNG de résolution$512 \times 512 \times 8\ bit$, prises le 06 Avril 2011 à 16 H.

IV.2. Analyse d'histogramme [36-38]

Sur la figure VI.1, nous représentons un exemple de l'image Météosat prise dans le canal Visible 0.6 le 06-04-2011à 16 H ; ainsi que les images chiffrées par les cinq modes de fonctionnement. Nous donnons sur la figure VI.2 l'histogramme de l'image Météosat ainsi que celles des images chiffrées. Nous remarquons que l'histogramme de l'image originale n'est pas uniforme et que nous pouvons facilement en tirer plusieurs remarques :

- Avec l'image de la figure IV.1 b chiffrée en mode ECB, nous remarquons clairement que les blocs homogènes dans l'image originale sont chiffrés avec la même manière.
- L'histogramme de la figure IV.2 b confirme cette remarque. Cet inconvénient majeur du mode ECB représente une faille que les attaquants peuvent exploiter pour tirer des informations de l'image originale.
- L'histogramme des autres modes dans la figure IV.2 c-d-e-f sont uniforme. Ce qui permet de résoudre le problème du mode ECB.

Même analyse pour le reste des figures (figures VI.3, 4, 5, 6,7)

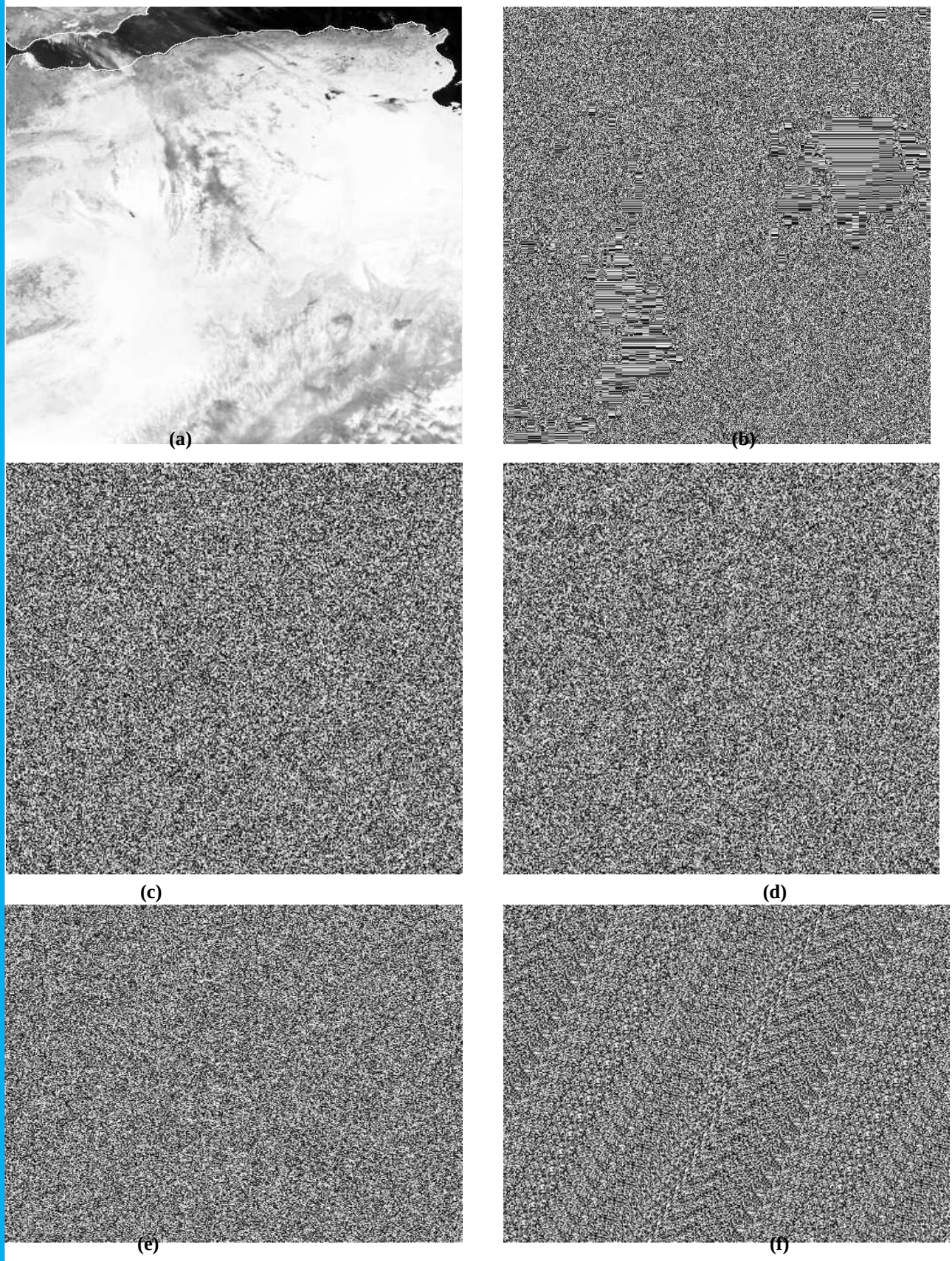


Figure IV.1 a) image originale, b) image chiffrée en mode EBC, c) image chiffrée en mode CBC, d) image chiffrée en mode CFB, e) image chiffrée en mode OFB (f) image chiffrée en mode CTR

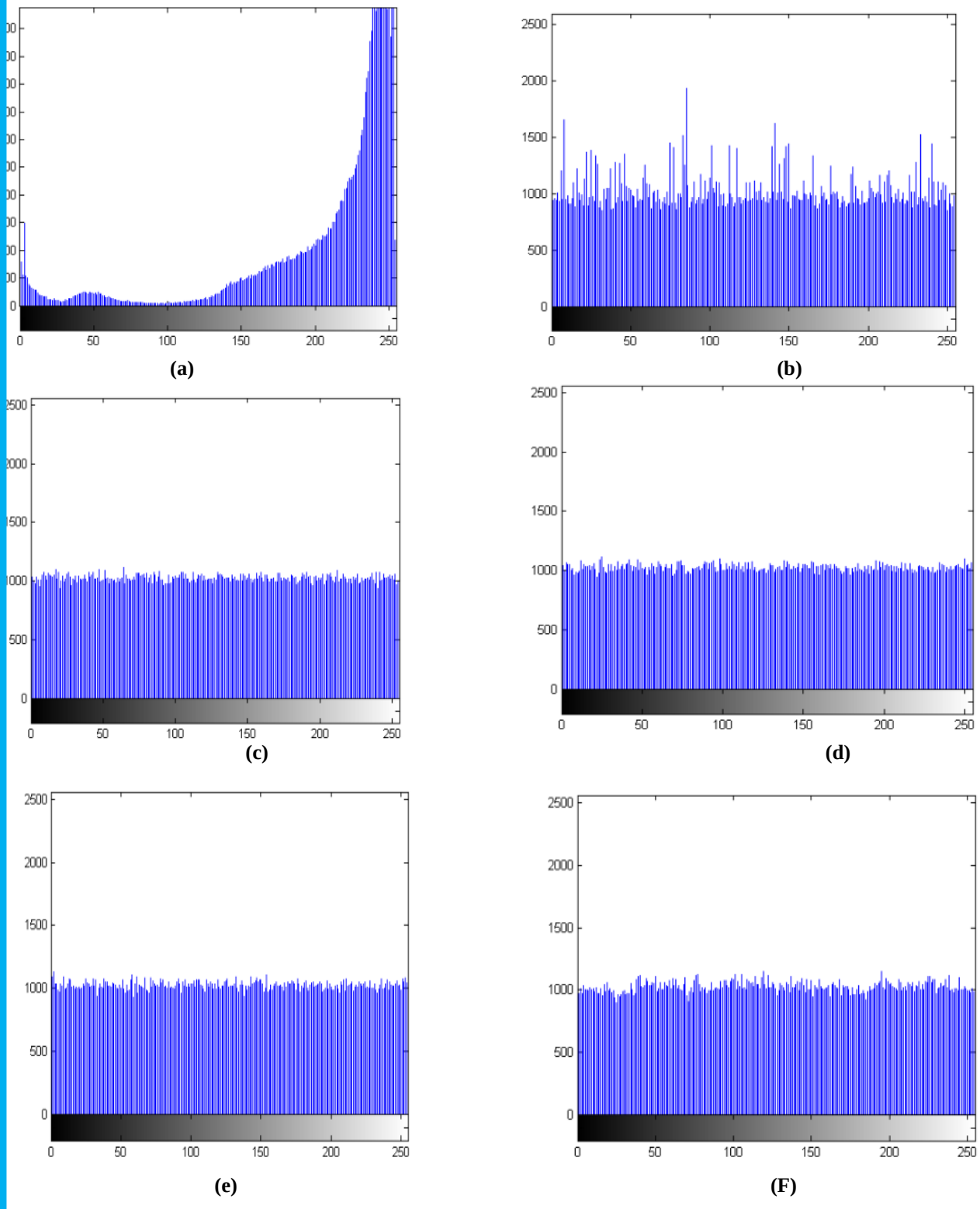
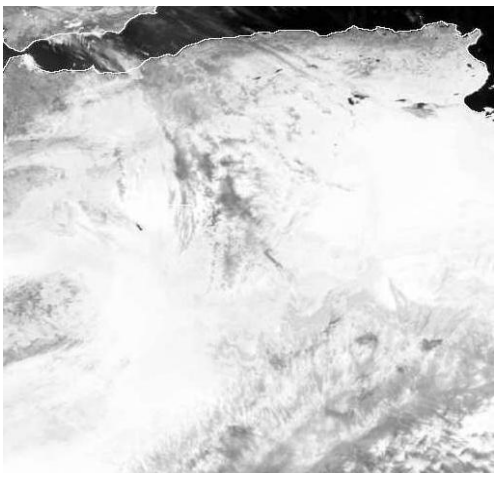
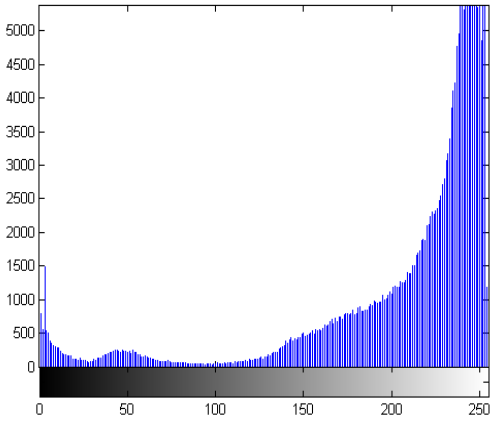


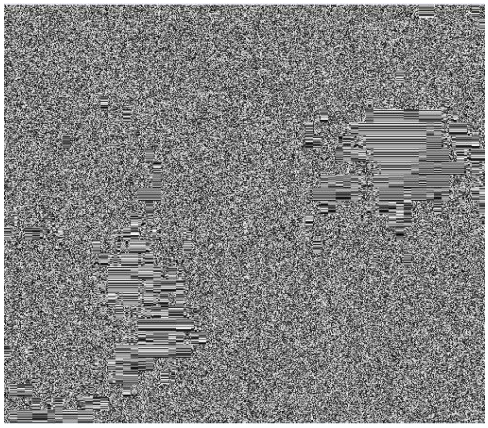
Figure IV.2 Histogrammes des images
a) Image originale, b) Image chiffrée en mode EBC, c) Image chiffrée en mode CBC, d) Image chiffrée en mode CFB, e) Image chiffrée en mode OFB, f) Image chiffrée en mode CTR



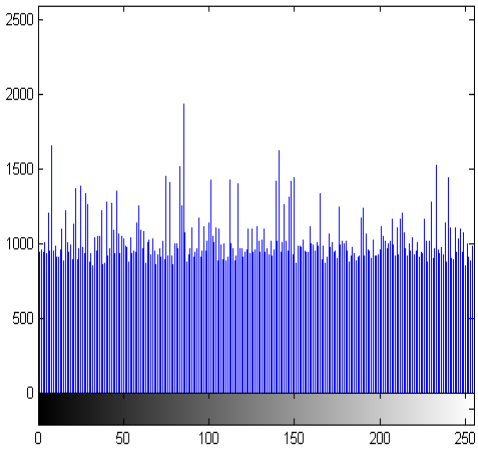
(a)



(b)



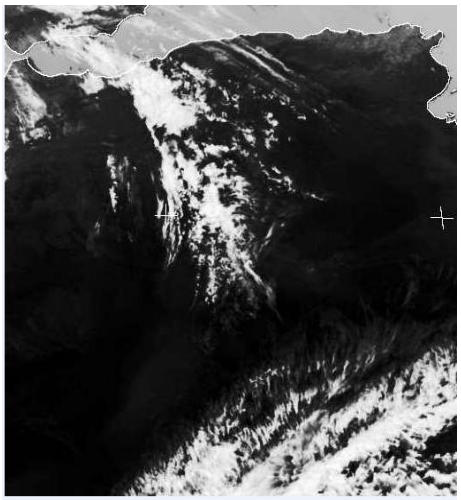
(c)



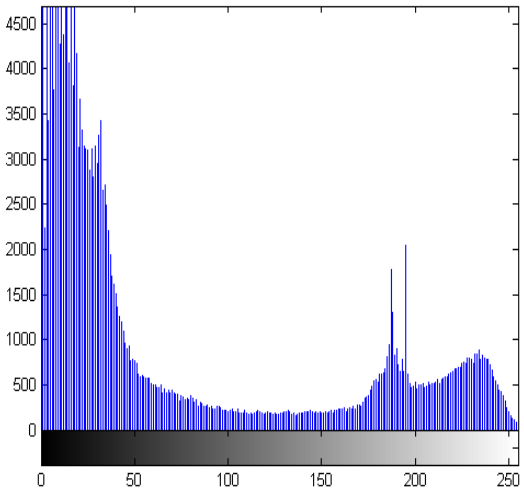
(d)

Figure IV.3 a) image originale, b) histogramme image originale , c) image chiffrée en mode ECB

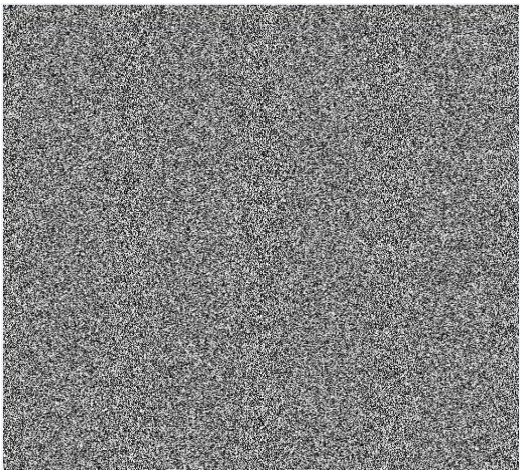
d) histogramme image chiffrée en mode ECB

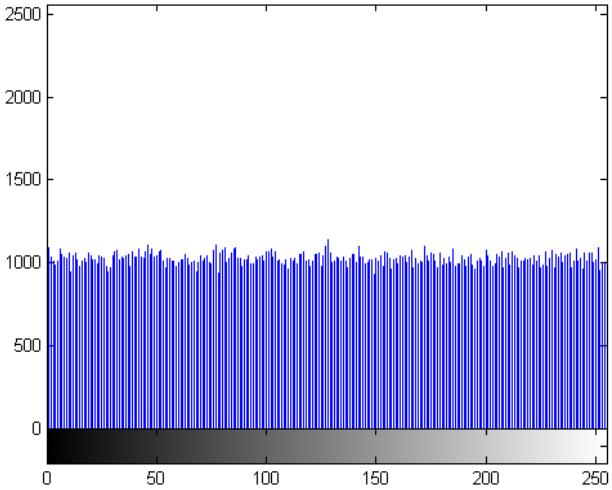


(a)



(b)

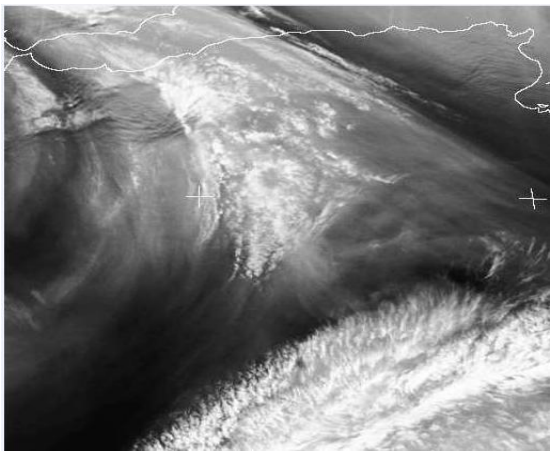


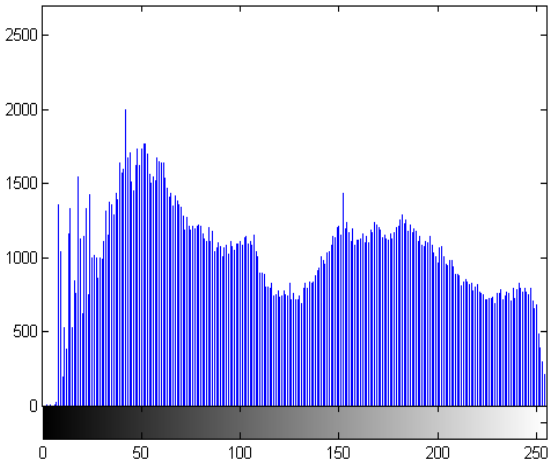


(c) **(d)**

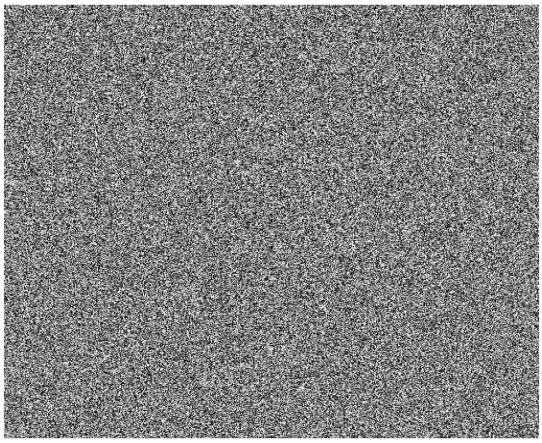
Figure IV.4 a) image originale, b) histogramme image originale , c) image chiffrée en mode CBC

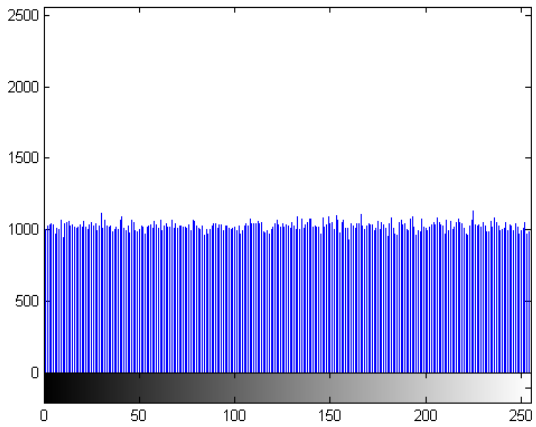
d) histogramme image chiffrée en mode CBC





(a) **(b)**

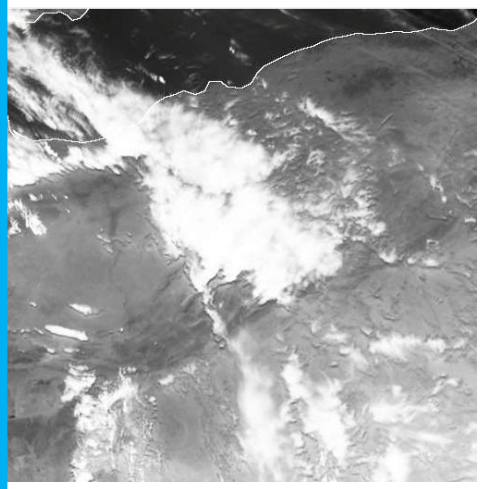




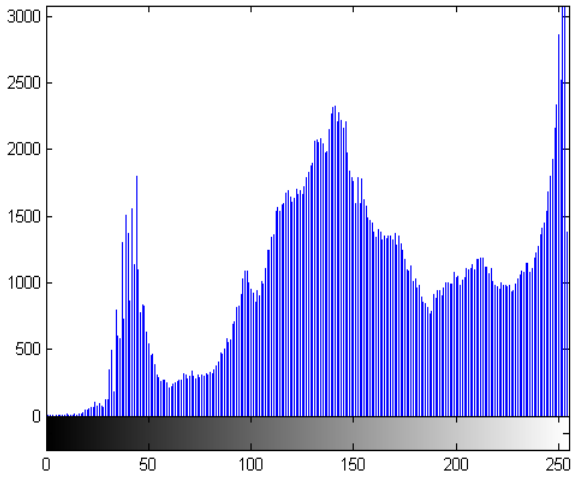
(c) **(d)**

Figure IV.5 a) image originale, b) histogramme image originale , c) image chiffrée en mode CFB

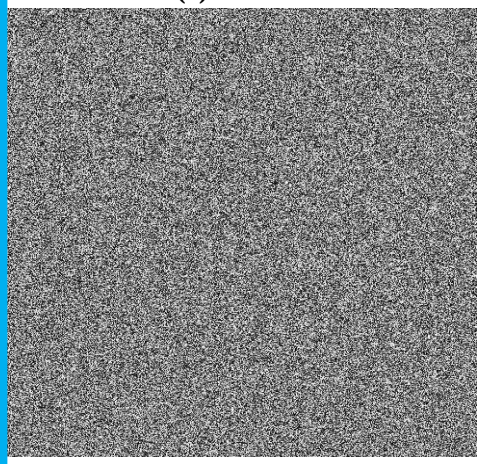
d) histogramme image chiffrée en mode CFB



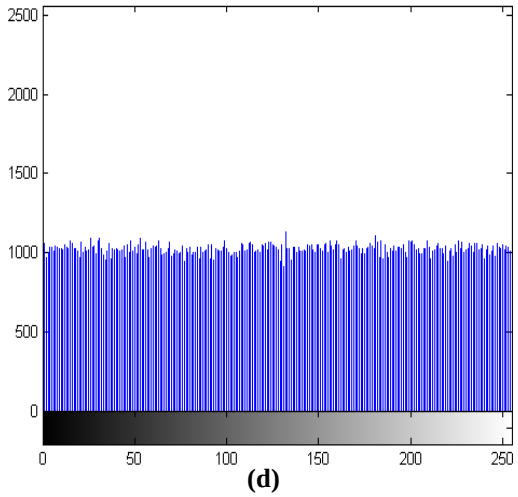
(a)



(b)



(c)

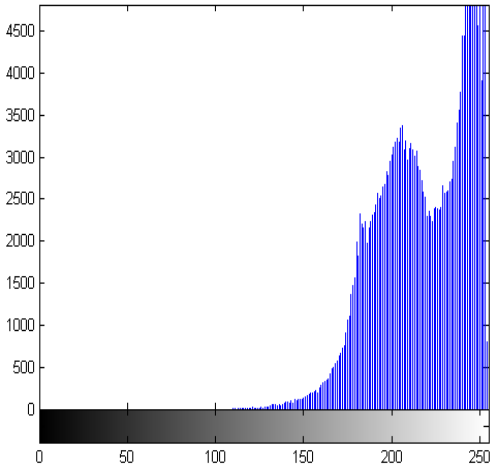


(d)

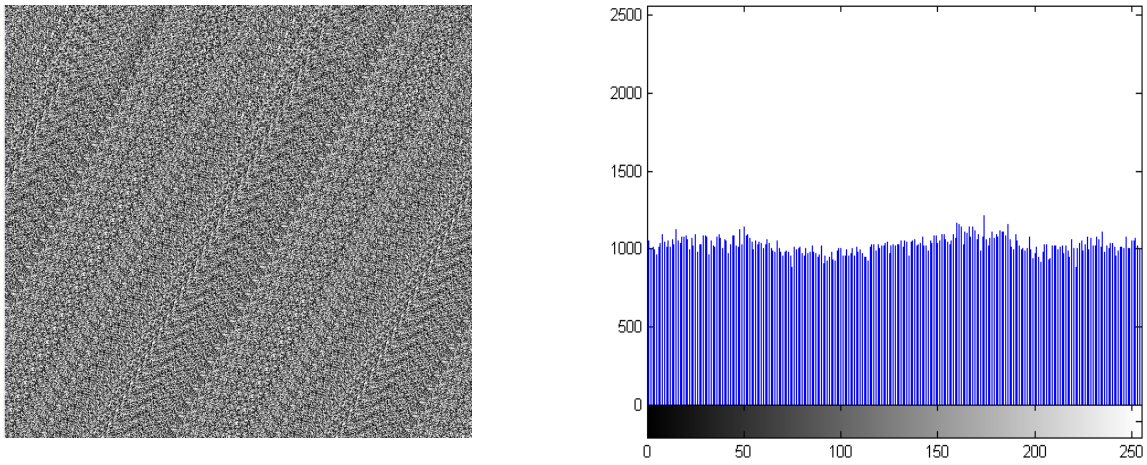
Figure IV.6 a) image originale, b) histogramme image originale , c) image chiffrée en mode OFB

d) histogramme image chiffrée en mode OFB

(a)



(b)



(c) **(d)**

Figure IV.7 a) image originale, b) histogramme image originale, c) image chiffrée en mode CTR

d) histogramme image chiffrée en mode CTR

IV.3. Corrélation entre l'image originale et l'image chiffrée [36-38]

La figure IV.8 montre la corrélation entre l'image originale et l'image chiffrée par les cinq modes de chiffrement. Nous remarquons que les deux modes OFB et CFB ont des performances qui excèdent les trois autres modes car ils ont un coefficient de corrélation inférieur. Le mode CBC a le coefficient le plus élevé.

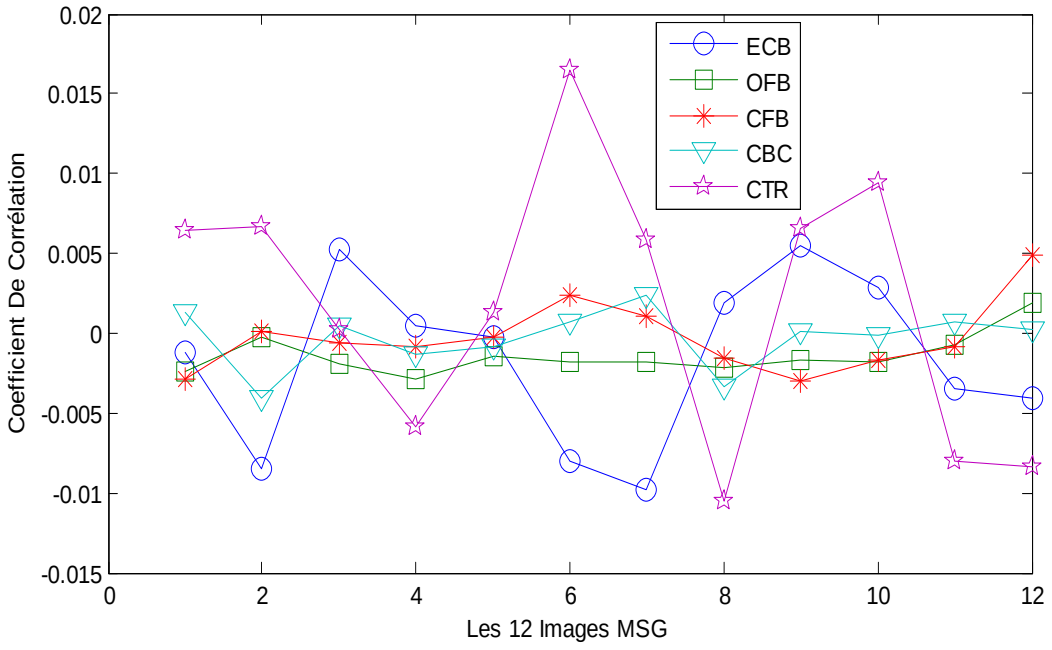


Figure VI.8 Corrélation entre l'image originale et l'image chiffrée

IV.4. La corrélation entre les pixels adjacents

Pour cette métrique, nous allons étudier la corrélation entre les pixels adjacents verticaux et horizontaux.

IV.4.1.La corrélation entre les pixels adjacents verticaux

Nous remarquons dans la figure VI. 9, qu'avec les quatre modes ECB, OFB, CFB, CBC, La corrélation entre les pixels a pratiquement le même coefficient zéro. Le mode CTR quand donne une forte corrélation par rapport aux autres modes.

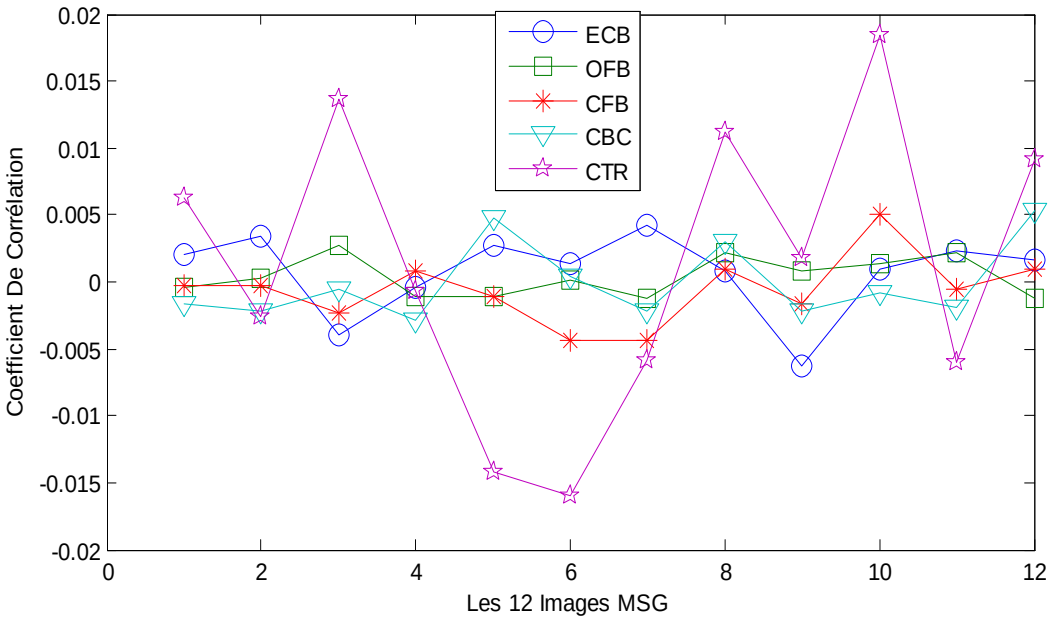


Figure VI.9 Corrélation entre les pixels adjacents verticaux

IV.4.2.La corrélation entre les pixels adjacents horizontaux

De la figure VI.10, nous remarquons que le chiffrement avec les trois modes OFB, CFB, CBC donne une forte décorrélation entre les pixels, par contre avec les modes ECB et CTR les pixels sont moins décorrélés.

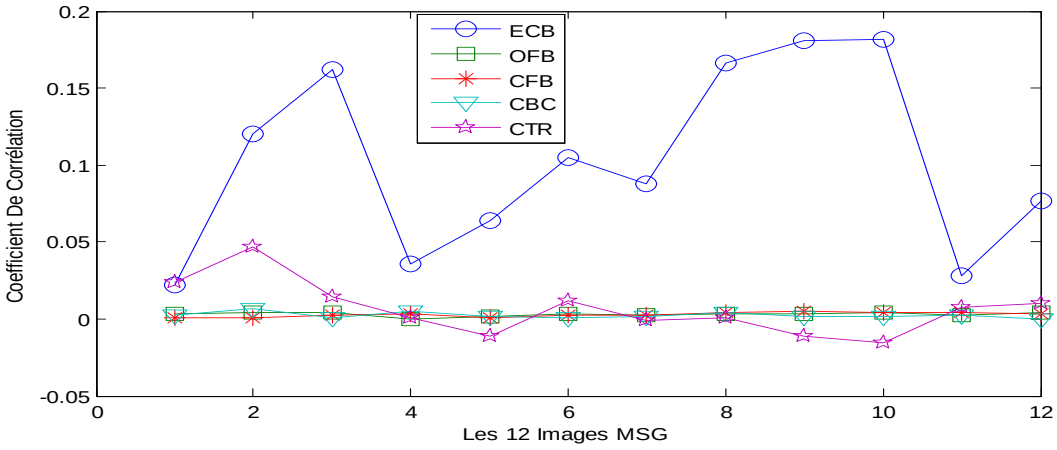


Figure VI.10 Corrélation entre les pixels adjacents horizontaux

IV.5. La sensibilité de clef [36-38]

Dans cette partie nous avons chiffré les images avec deux clefs légèrement différentes (un bit différent), ensuite, nous avons calculé la corrélation entre les images pour les cinq modes. Nous constatons à partir de la figure VI.11, que les deux images obtenues sont complètement indépendantes l'une par rapport à l'autre pour les cinq modes (faible corrélation).

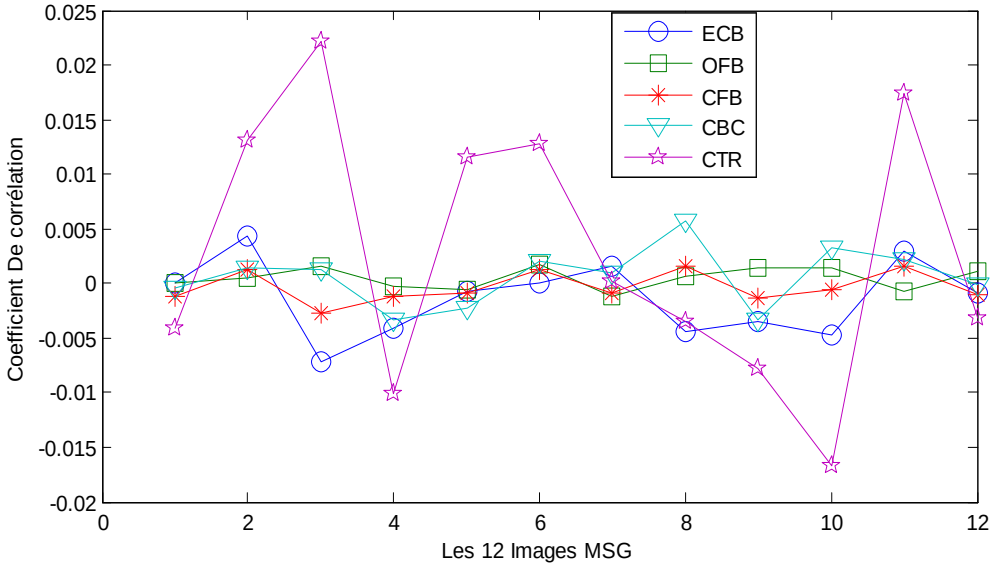


Figure VI. 11 sensibilités de clef

IV.6. Contrôle d'intégrité [36-38]

Pour ce test nous avons chiffré les douze images en mode CTR, après nous avons remplacé un bloc de 64×64 pour les 12 images chiffrées avec des nouveaux blocs, à la réception nous avons calculé l'empreinte cryptographique pour les douze images déchiffrées, après nous avons comparé avec l'empreinte cryptographique déjà calculée à l'émission. Et nous avons obtenu les résultats suivants :

Images	Empreinte émission	Empreinte réception
1	2366066258077760	7037531707906123
2	4700807931588939	8955723006089274
3	1480613745413110	6735069093392235
4	9470400119164090	4321858620076709
5	1120109572998128	2959747088018309
6	1206980435821704	6161052925191214
7	1036336179776336	3308328482669586
8	5982384706154786	3555002447183723
9	1144896521516351	3314035514716729

10	1508371361990782	3282495943667020
11	7703120859811909	4249934748266025
12	9835147943326845	5371308422246467

Tableau VI.1 Comparaison entre les empreintes

Depuis le Tableau VI.1, nous constatons que si nous modifions des images dans le canal de transmissions, les empreintes cryptographiques de ces images après déchiffrement seront complètement différentes par rapport aux empreintes des images originales, alors nous concluons que le problème d'intégrité est vérifié.

IV.7. Comparaison avec d'autre algorithmes

Pour évaluer notre cryptosystème hybride, nous la comparons avec d'autres algorithmes utilisés dans la littérature. Pour cette comparaison, nous utilisons l'image LENA qui est une image de référence très utilisée en traitement d'images. Elle est au format PNG de taille 80×80 codée sur 8 bits.

IV.7.1. Comparaison de la corrélation entre les pixels adjacents pour différents algorithmes

Afin de comparer les performances de notre cryptosystème avec celles des algorithmes existants dans la littérature, nous avons pris comme référence l'algorithme AES et nous avons calculé la différence entre les coefficients de corrélation des pixels adjacents des images cryptées. Nous donnons dans le tableau IV.2, les résultats de notre comparaison avec les algorithmes ECKBA(Enhanced Chaotic Key-Based Algorithm)[28] qui a été conçu à l'origine par D. Socek et S.Li en 2005 [34] et basé sur le travail de CKBA (Chaotic Key-Based Algorithm) proposé par Yen et Guo [35] ainsi que l'algorithme développé par Mansour et al [29] qui a proposé et évalué une nouvelle architecture dynamique de sécurité pour les RCSF (réseaux de capteurs sans fil). Il permet de garantir et de maintenir la sécurité des communications durant toute la durée de vie du réseau. Pour la direction verticale, AES a des performances meilleurs que les autres algorithmes étudiés en ayant un coefficient de corrélation des pixels adjacents beaucoup plus inférieur. Les pixels adjacents diagonalement cryptés par AES ont un coefficient de corrélation largement inférieur comparé aux autres algorithmes. L'AES possède le plus faible coefficient de corrélation horizontal. Nous pouvons conclure que l'algorithme AES donne de bonnes propriétés en diminuant la corrélation des pixels adjacents et surpasse les performances des algorithmes étudiés dans la littérature.

Direction des pixels adjacents / Images cryptées	Horizontale	Verticale	Diagonale
Image non cryptée	0,8426	0,9317	0,8012
Image cryptée selon ECKBA [28]	0,0760	0,0227	-0,0012
Image cryptée selon Mansour et al. [29]	0,0479	-0,0414	-0,0416
Image cryptée selon AES	0,0237	-0,0139	-0,0162

Tableau VI.2 Coefficient de corrélation des pixels adjacents

IV.7.2.Temps d'exécution

Les résultats du tableau VI.3 ont été évalués par MATLAB (R2011a) [31-32]. Les expériences ont été menées à l'aide d'un processeur Intel® Core i3 avec une fréquence de 2.4 GHz. Dans ce tableau, le temps d'exécution nécessaire pour chaque algorithme pour crypter une image Lena (80×80) est fourni. L'algorithme Mansour et al est plus rapide avec un temps ne dépassant pas 5 secondes. En effet, le temps d'exécution de l'algorithme ECKBA est de 269.242 secondes, il est alors 12 fois plus lent que AES. L'algorithme AES s'exécute en 11.94 secondes, ce qui est un temps d'exécution proche de celui de Mansour et al.

Algorithmes	Temps d'exécution(s)
ECKBA [25]	269.242
Mansour et al. [25]	4.1496
AES	11.94

Tableau VI.3 Mesure du temps d'exécution en seconde des différents algorithmes testés

IV.8. Discussion

Nous avons présenté dans ce chapitre, les résultats obtenus pour l'application de notre cryptosystème hybride pour la transmission des images MSG ; Pour le plan de confidentialité, les cinq modes avec AES ont garanti cette fonction sauf avec le mode ECB qui a présenté un inconvénient pour les zones homogènes et les histogrammes ont confirmé cet inconvénient.
Sur le plan d'intégrité, les résultats que nous avons obtenus lorsque nous avons modifié un bloc de 64×64 pour chaque image sont encourageants et l'empreinte cryptographique à la

réception a complètement changé, et nous avons facilement détecté les modifications qui ont touché les images dans le canal de transmission.
Nous sommes aussi satisfaits pour les résultats obtenus pour la sensibilité des clefs, les images sont complètement décorrélées malgré que nous n'ayons changé qu'un seul bit de la clef de session !

Laboratoire d'Analyse et Modélisation des Phénomènes Aléatoires, UMMTO, BP 17 RP, 15000, Tizi-Ouzou, Algérie

Au cours de ce mémoire, nous avons étudié trois problématiques liées à la protection des images MSG (Météosat Second Génération). Le premier problème, concerne la transmission sécurisée d'images MSG. Le deuxième problème, est d'assurer la fonction d'authentification. Le dernier problème est de garantir l'intégrité des images. Nous avons développé un cryptosystème hybride pour faire face à ces problèmes à la base des deux algorithmes AES (Advanced Encryption Standard), et RSA et la fonction d'autocorrélation.

Dans un premier temps, nous avons présenté plusieurs techniques et quelques théories de la cryptographie, qui nous permettrons de comprendre cet axe de recherche. Nous avons aussi développé, en détails, les deux algorithmes AES et RSA, nécessaires dans notre cryptosystème de transmissions en donnant leurs avantages et inconvénients.

Dans la troisième partie, nous avons donné en détail notre cryptosystème hybride de transmission. Les caractéristiques de la base d'images MSG est décrites afin de l'adapter pour une transmission en mode chiffré. Notre cryptosystème est composéde deux étages : étage d'émission et celui de réception, reliés entre eux par un canal de transmission. A l'émission la clef de session est extraite. Ensuite nous avons chiffré cette dernière, par l'algorithme RSA, deux fois pour assurer l'authentification et transmettre la clef de façon sécurisée. Nous avons aussi chiffré les images Météosat avec la clef de session en utilisant l'AES afin de garantir la confidentialité. A la réception de ces images, des fonctions réversibles sont élaborés pour les déchiffrer et de garantir que l'image ne soit pas modifiée dans le canal de transmission.

Pour tester notre cryptosystème et analyser ces résultats, nous avons exploité plusieurs métriques d'évaluation du degré de chiffrement (analyse d'histogramme, calcul de la corrélation entre l'image originale et celle chiffrée, calcul de la corrélation entre les pixels adjacents, analyse de la sensibilité de la clef secrète, analyse de la fonction de corrélation pour contrôler l'intégrité). Le nombre d'images Météosat que nous avons chiffré est de 12 images issues du satellite MSG dans différents canaux de type PNG de taille$512 \times 512 \times 8\ bit$.

A partir des résultats obtenus, nous avons remarqués que tous les histogrammes des images chiffrées sont uniformes pour les modes d'opérations (OFB, CFB, CBC, CTR). Avec le mode de chiffrement ECB, l'histogramme présente des pics dû aux zones homogènes.

Nous avons aussi constatés que les pixels sont fortement décorrélés pour les cinq modes de chiffrement. Cela donne une distribution aléatoire d'information. En utilisant notre cryptosystème de transmission, les images originales et celles chiffrées sont fortement décorrélés. Le temps d'exécution de l'algorithme AES est très acceptable en comparaison avec d'autres algorithmes de cryptages.

Nous avons aussi constaté que l'application de la fonction d'autocorrélation pour vérifier l'intégritéde notre cryptosystème donne de bons résultats.

Nous avons remarqué que le changement de la taille d'un bloc dans le canal de transmission, conduit au changement total de l'empreinte digitale à la réception.

Nous pouvons conclure à partir de l'ensemble des résultats obtenus, qu'il est très conseillé de commencer à chiffrer les images MSG avec l'algorithme AES à la place de l'algorithme 3-DES utilisé par les stations météorologiques.
Comme perspectives, nous donnons les améliorations suivantes :

- Utilisation du chaos [39] dans l'étape de génération de la table S-box de l'algorithme AES.
- Implémentation matérielle de l'AES pour chiffrer les images MSG dans le but d'optimiser le temps de transmission.
- Amélioration du cryptosystème en introduisant une étape de codage et de compression de données pour optimiser le système de cryptage et de transmission [40-41].

Laboratoire d'Analyse et Modélisation des Phénomènes Aléatoires, UMMTO, BP 17 RP, 15000, Tizi-Ouzou, Algérie

[1] un cours tutoriel du centre canadien de télédétection, "notions fondamentales de télédétection, " ressources naturelles canada, 2011.

[2] Suyintan, "meteorological satellite systems," springer, new york, 2014.

[3] J. daemen, and V. rijmen, "AES proposal: the rijndael block cipher," technical report, proton world int. l, katholieke universiteit leuven, esatcosic, belgium, 2002.

[4] William stallings, "cryptography and network security principles and practice," pearson, united states of america, 2011, p. 900.

[6] NIST: national institute of standards and technology, web site: http://www.nist.gov/aes/

[7] Han wen, "AES encryption algorithm analysis and security study," computer applications of petroleum 2008,vol 16 no.2

[8] Abdulkarim amer shtewi, m. hasan, and Abd el fatah, A. hegazy, "an efficient modified advanced encryption standard (MAES) adapted for image cryptosystems," IJCSNS international journal of computer science and network security, vol.10 no.2,pp.226-232 february 2010.

[9] R. rivest, A. shamir, and l. adleman, "a method for obtaining digital signatures and public-key cryptosystems," communications of the ACM, 21:120-126, 1978.

[10] R. stinson, "cryptography: theory and practice, (discrete mathematics and its applications)," chapman & hall/ crc press, new york, november 2005.

[11] Bruce schneier, "applied cryptography," CRC press, united states of America, 1996, p. 780.

[12] Morris dworkin, "recommendation for block cipher modes of operation," NIST special publication 800-38 ,2001 edition.

[13] Robert pre, "system DVB MSG," fontana roberto software, EUMET cast, p. 22, 2008.

[14] Ralph merkle, martin hellman,"on the security of multiple encryption, communications of the acm," vol 24, no 7, pp 465–467, july 1981.

[15] J. daemen, V . rijmen, "the design of rijndael.," springer new york, inc. secaucus, nj, usa, 2002.

[16] W. diffie, M. hellman, "new directions in cryptography," IEEE transactions on information theory. 644-654, 1976.

[17] Peter w. shor, "polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM journal of computing, 26(5):1484-1509, 1997.

[18] S. goldwasser and S. micali, "probabilistic encryption," journal of computer and system sciences 28 :270-299, 1984.

[19] G. pafford, S. garnkel, and A. schwartz, "practical unix & internet security," o'reilly & associates , 1004 pages. second edition, april 1996, usa.

[20] A. lenstra, E. tromer, A. shami, W. kortsmit "factoring estimates for a 1024-bit RSA modulus," lecture notes in computer science 2894 :55-74, january 2003.

[21] J. menezes, C. van oorschot, and scott A. vanstone, " hand-book of applied cryptography (5th edition)," crc press llc, florida,eua, august 2001.

[22] W. diffie and M. hellman, "exhaustive cryptanalysis of the data encryption standard, " IEEE computer 10(6):74-84, 1977.

[23] D. renaut, "les satellites météorologiques," météo-france - direction commerciale et de la communication, la météorologie - n° 45, pp. 33-37, france, 2004.

[24] M. fariza,et Z. ameur "détermination du gisement solaire par images msg, " mémoire de magister en teledetection, ummto, 2011.

[25] Johannes schmetz, marianne könig, paolo pili, sergio rota, alain ratier and stephen tjemkes, "meteosat second generation (msg): status after launch," EUMET sat, Germany.

[26] Johannes schmetz , paolopili , stephen tjemkes, dieter just , jochenker kmann ,sergio rota, and alain ratier, "an introduction to meteosat second generation (MSG)," article, American meteorological society, pp. 977-992, 2002.

[27] Eumetsat," MSG, meteosat data collection service," accurate support for meteorology and weather prediction, Germany, 2010.

[28] Ghada zaïbi, " sécurisation par dynamiques chaotiques des réseaux locaux sans fil au niveau de la couche MAC, " thèse de doctorat en systèmes embraques, Toulouse, 2012.

[29] Ismail mansour, Gerard chalhoub, and Bassem bakhache, "evaluation of a fast symmetric cryptographic algorithm based on the chaos theory for wireless sensor networks", MWNS (international symposium on mobile wireless network security), liverpool, uk, 25-27 june, 2012.

[30] Goumidi djamal eddine, "fonction logistique et standard chaotique pour le chiffrement des images satellitaires," mémoire de magister en télécommunications spatiales, université constantine, 2010.

[31] Rafael gonzalez, Richard woods, and steven eddins, "digital image processing using matlab," prentice hall, 2004.

[32] Rafael gonzalez, Richard woods, "digital image processing," 21e, prentice hall, 2001.

[33] Jean philippe gaulier, "analyse des algorithmes finalistes concourant pour le futur standard AES," conservatoire national des arts et métiers centre régional associé de limoges, france.

[34] Daniel socek, Shujun li, Spyros magliveras, borkofurht. "enhanced 1-d chaotic

key-based algorithm for image encryption," in securecomm , 2005, p406-p407.

[35] j. yen and j. guo. ‘’a new chaotic key-based design for image encryption and decryption’’, in proceeding of 2000 ieee international conference on circuits and systems (ISACS 2000), vol.4, pages49-52, 2000.

[36] Boukhatem mohammed belkaid et lahdir mourad, “chiffrementhybride pour la transmission sécurisée des images médicales,” biomeic’14 (biomedical engineering international conference), tlemcenalgeria, october 15-16, 2014

[37] Boukhatem mohammed belkaid and lahdir mourad, “security enhancement of digital msg images transmission with the five modes using hybrid aes-rsa algorithm,” iceeb’14 (first international conference on electrical engineering), biskra-algeria, december 07-08, 2014.

[38] Boukhatem mohammed belkaid and lahdir mourad, “secure transfer of medical images using hybrid encryption,” iccvia’15 (computer vision and images analysis), sousse-tunisia, january 18-20, 2015.

[39] H. hamiche, M. lahdir , M. tahanout and S. djennoune , “masking digital image using a novel technique based on a transmission chaotic system and spiht coding algorithm” international journal of advanced computer science and applications (IJACSA), 3(12), 2012.

[40] A. jose marconi and M. rodrigues, "transfert sécurisé d'images par combinaison de techniques de compression et cryptage" thèse de doctorat de l'université de Montpellier ii, oct. 2006.

[41] c. oatrieux, H. maitre, “images médicales, sécurité et tatouage,” annales des télécommunications, numéro spécial santé, vol. 58, pp. 782-800, 2003.

Définitions et théorèmes

Réseau de Feistel

Le réseau de Feistel inventé par Horst Feistel est un réseau appliqué en premier lieu à l'algorithme Lucifer puis au DES et d'autres algorithmes de cryptages tel que : RC5 Twofish, Blowfish…

Il est simple et le cryptage et décryptage sont similaires. Il est basé sur des opérations de substitutions et de permutations avec une fonction principale changeant de clé à chaque tour.

La figure A représente un exemple de cryptage et décryptage selon la structure de Feistel.

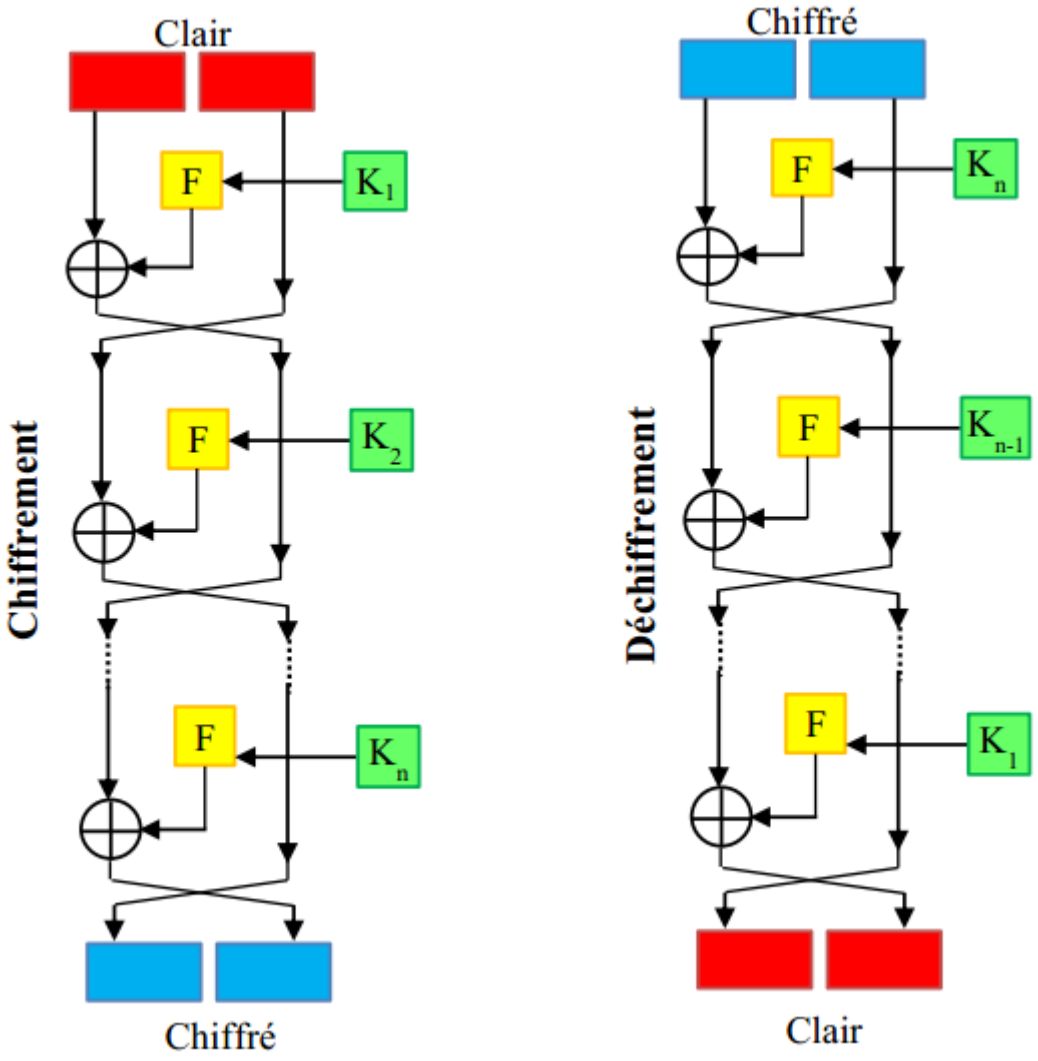


Figure A. Réseau de Feistel à n tours utilisant l'opérateur XOR.

Masque jetable (One time Pad)

Un masque jetable ou (one time Pad) est appelé aussi le chiffrement de Vernam est un chiffrement disant incassable. Il consiste à appliquer à l'aide d'un XOR à une donnée de longueur n octets, une clé aléatoire de même longueur. La clé ne doit être appliquée qu'une seule fois. La difficulté dans ce type de cryptage est la condition de non réutilisabilité de la clé et l'aspect aléatoire de la clé (et non pas pseudoaléatoire), ainsi que la taille de la clé si la donnée est de taille importante.

L'algorithme DES

Le Data Encryption Standard (DES) est un algorithme de chiffrement symétrique (chiffrement par bloc) utilisant des clés de 56bits. Son emploi n'est plus recommandé aujourd'hui, du fait de sa lenteur à l'exécution et de son espace de clés trop petit permettant une attaque systématique en un temps raisonnable. Quand il est encore utilisé c'est généralement en Triple DES, ce qui ne fait rien pour améliorer ses performances. DES a notamment été utilisé dans le système de mots de passe UNIX.

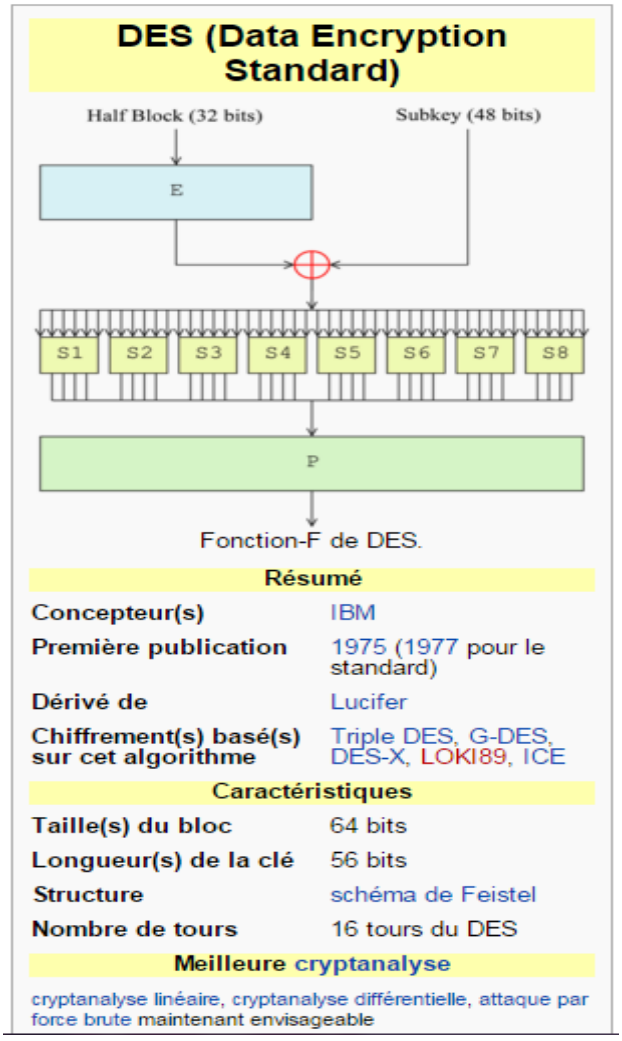


Figure B. l'algorithme DES

L'algorithme triple DES

Le Triple DES (aussi appelé 3DES) est un algorithme de chiffrement symétrique par bloc, enchaînant 3 applications successives de l'algorithme DES sur le même bloc de données de 64 bits, avec 2 ou 3 clés DES différentes.

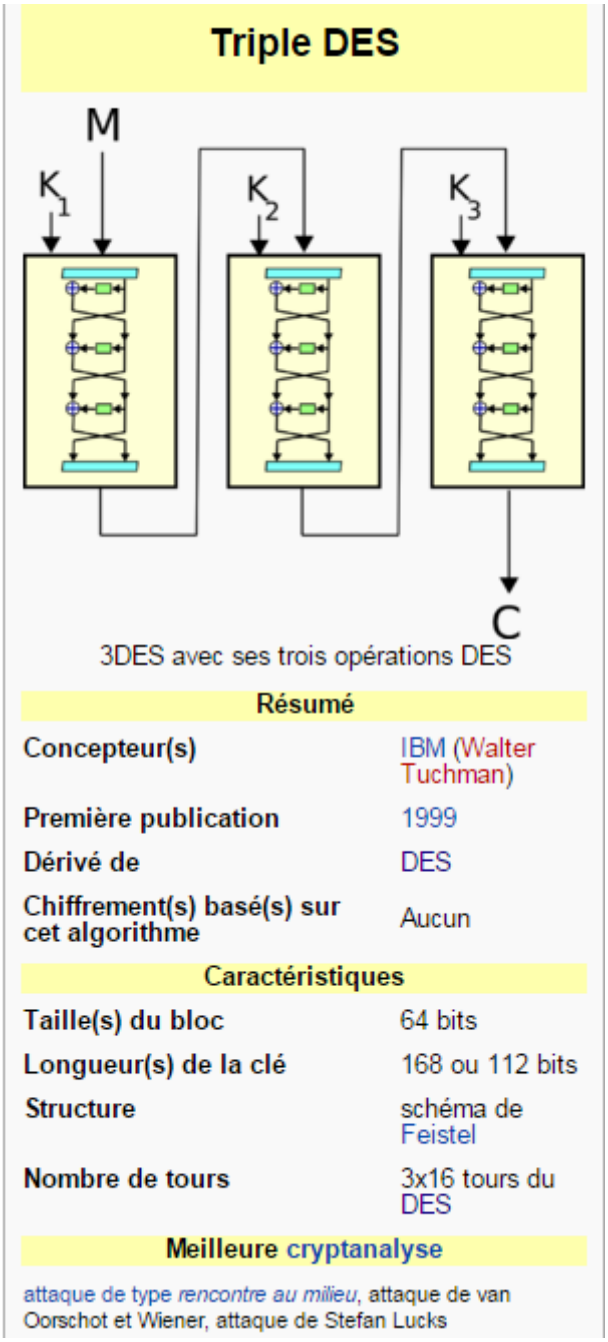


Figure C. l'algorithme 3-DES

CHAOS

Le terme "chaos" définit un état particulier d'un système dont le comportement ne se répète jamais, est très sensible aux conditions initiales, est imprédictible à long terme. Des chercheurs d'horizons divers ont alors commencé à s'intéresser à des problèmes non

linéaires jusqu'alors sans solution parce qu'imprédictibles et regroupés sous la dénomination de chaos. Ils ont cherché à répondre à des questions telles que : Les arythmies cardiaques ou les variations d'une population animale obéissent-elles à des règles? Les mouvements commerciaux ou les marchés financiers peuvent-ils s'expliquer? Le modèle du biologiste Robert May décrit l'évolution de la population d'une espèce en fonction des contraintes du milieu (famines, épidémies, ...) et obéit à une dynamique chaotique (équation logistique). Richard Cohen, physicien et cardiologue, a montré lors de simulations que le caractère chaotique du rythme cardiaque pourrait expliquer l'apparition de crise cardiaque. William Baumol et Jess Benhabib, économistes, se sont intéressés à la théorie du chaos et à ses applications à l'économie. Le chaos a ainsi trouvé de nombreuses applications dans les domaines tant physique que biologique, chimique ou économique, par exemple.

CORPS FINI

En mathématiques et plus précisément en algèbre, un corps fini est un corps commutatif qui est par ailleurs fini. À isomorphisme près, un corps fini est entièrement déterminé par son cardinal, qui est toujours une puissance d'un nombre premier, ce nombre premier étant sa caractéristique. Pour tout nombre premier p et tout entier non nul n, il existe un corps de cardinal p^n, qui se présente comme l'uniqueextension de degré n du corps premier Z/pZ.

Les corps finis sont utilisés en théorie algébrique des nombres, où ils apparaissent comme une structure essentielle à la géométrie arithmétique. Cette branche a permis, entre autres, de démontrer le dernier théorème de Fermat.

Les corps finis ont trouvé de nouvelles applications avec le développement de l'informatique. En théorie des codes, ils permettent par exemple de déterminer des codes correcteurs efficaces. Ils interviennent également en cryptographie, dans la conception deschiffrements à clé secrète comme le standard AES, ainsi que dans celle des chiffrement à clé publique, à travers, entre autres, le problème du logarithme discret.