République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de Tizi-Ouzou.

Faculté de Génie Electrique et d'Informatique.

Département d'Electronique.

Mémoire de fin d'études

En vue d'obtention d'un diplôme Master en électronique

Option : Télécommunication et réseaux

Thème

Cryptographie : Approche Quantique

Dirigé par :

M[r] Ziani Rezki

Présenté par

M[r] Bouizeri Brahim

Devant le jury :

Président : M[r] Oualouche Fethi

Examinateur : M[r] Hameg Slimane

Promotion 2016/2017

Introduction générale : [Sou-08]

Nous vivons actuellement dans un monde, où l'homme a ressent le besoin de transı informations de manière sûre. Avec le progrès de l'informatique et l'apparition de l'Internet, la sécurisation des données s'est imposée comme un passage incontournable dans le transit des informations sensibles. Ainsi, on la retrouve dans des domaines variés tels que : la protection des données confidentielles (bases de données, Email, ...), la sécurisation des communications, le paiement sécurisé (cartes bancaires) ...

Il s'avère alors que la sécurité de l'information est l'un des piliers majeurs de tout système d'information quelque soit sa nature. Pour protéger une information, rien de plus simple que de l'écrire de sorte que seul le destinataire légitime la comprenne, c'est exactement ce que fait la cryptographie.

Malgré ses diverses formes, sans cesse en progrès, le fonctionnement de la cryptographie reste peu connu par le grand public mise à part quelques formes primitives englobant des algorithmes rudimentaires dans leur ensemble du fait qu'ils consistaient notamment au remplacement de caractères par d'autres. Quand aux applications modernes, avec l'usage des mathématiques, de la physique quantique, et des technologies les plus avancées, elles sont encore beaucoup moins connues.

Ces dernières permettent de protéger le contenu d'un message en s'aidant le plus souvent de clés, et suivant que ces clés de chiffrement sont gardées secrètes ou pas, les algorithmes de chiffrement se scindent en deux principales catégories: les algorithmes à clé publique (RSA, ElGamal,…) et les algorithmes à clé privée (DES, 3DES, AES, IDEA,…).

Cependant, ces deux principes soufrent de quelques vulnérabilités qui pourraient remettre en cause la sécurité des systèmes qui les adoptent. D'autant plus, qu'avec le développement technologique, et la force de calcul offerte par les nouveaux ordinateurs, précisément les ordinateurs quantiques, la sûreté des systèmes de cryptage actuels sera inévitablement remise en cause.

Cela dit, l'ordinateur quantique n'est pas la seule application de l'informatique quantique, la cryptographie quantique constitue également l'une des applications, les plus en vogue, de l'informatique quantique et qui permet de surmonter plus d'un obstacle en cryptographie classique. De part les principes sur lesquelles se basent la cryptographie quantique, et en plus de la sûreté des clés qu'elle offre, la cryptographie quantique résout l'une des problématiques majeures de la cryptographie, en l'occurrence la distribution des clés. En effet, la cryptographie quantique se base sur l'usage des états quantiques des photons, sur les principes d'incertitudes d'Heisenberg, et celui du non-clonage, qui permettent de détecter toute tentative d'intrusion.

L'un des points motivant notre recherche est donc l'étude de la cryptographie sous ces deux aspects (symétrique et asymétrique) dans un premier temps, puis dans un second temps, son étude sous la perspective quantique, et nous avons choisi le logiciel Qucrypt pour la simulateur du protocole BB84,vu l'apport escompté sur le plan sécuritaire.

Le présent mémoire est organisé en quatre chapitres :

Le premier chapitre décrit des notions de bases liées au domaine de la cryptographie. Dans le deuxième chapitre, nous traitons les différents algorithmes cryptographiques.

Le troisième chapitre, traite quelques propriétés de l'information quantique, les principes de la cryptographie quantique, puis ceux du protocole BB84.

Le quatrième chapitre est consacré à la simulation du protocole quantique BB84. Nous terminerons le présent mémoire par une conclusion et une présentation de quelques perspectives qui pourraient enrichir ce travail ultérieurement.

I-1- Introduction :

La cryptographie, reconnue comme étant la science du secret, doit assurer différentes fonctions qui devront permettre une communication sécurisée et sans faille entre destinataire et destinateur, cela se fera grâce au mécanisme de chiffrement. Ceci dit, le mécanisme de chiffrement semble ne pas être utilisé seulement pour la transformation du message à transmettre en un autre inintelligible, autrement dit pour en assurer la confidentialité. Un processus de chiffrement peut être utilisé de diverses façons pour assurer d'autres objectifs, authentification, intégrité,…, créant ainsi de nouveaux mécanismes qui, de là même, seront combinés entre eux pour assurer une communication secrète.

Avec l'avènement des réseaux, et tout particulièrement Internet, la cryptographie prend maintenant une nouvelle dimension, économique cette fois. C'est en effet toute la sécurité du commerce électronique qui dépend maintenant de l'inviolabilité des codes cryptés.

Dans ce qui suit, nous décrivons le principe fondateur de la cryptographie, les types de menaces et les objectifs de la sécurité de l'information. Nous mettrons l'accent sur les objectifs considérés comme étant les piliers même de cette sécurité, avant de faire une description des mécanismes assurant ces objectifs, entre autres le chiffrement qui lui, sera détaillé par la suite.

I-2- Principe fondateur de la Cryptographie :

I-2-1- Notion cryptologie : [Sou-08]

La cryptographie constitue en réalité une branche d'une science plus élargie, la cryptologie.
L'origine du mot cryptologie est dérivée du Grecque **kryptos** (caché) et **logos** (mot)
La cryptologie renferme à la fois la cryptographie et la cryptanalyse.

I-2-2- Cryptographie : (du grec kruptos et graphein) est la discipline incluant les principes, les moyens et les méthodes de transformation des données, dans le but de masquer leur contenu, empêcher leur modification ou leur utilisation illégale, ainsi que les opérations inverses, pour rendre le document à nouveau intelligible.

I-2-3- Cryptanalyse : (du grec kruptos et analusis)
« Résolution, dissolution », est l'art de décoder un message chiffré en mêlant une intéressante combinaison de raisonnement analytique, d'application d'outils mathématiques, de découverte de redondances, de patience, de détermination, et de chance.

Les deux disciplines de cryptographie et de cryptanalyse s'alimentent l'une l'autre. On ne peut pas évaluer la sécurité d'un mécanisme sans le soumettre à des attaques qui, à leur tour, conduisent à des critères de conception pour rendre les procédés plus sûrs. Ces derniers seront à nouveau passés au crible du cryptanalyste.

I-3- Objectifs de la sécurité de l'information: [Dje-09]

Définition de l'information : L'information désigne à la fois le message à communiquer et les symboles utilisés pour l'écrire ; elle utilise un code de signes porteurs de sens tels qu'un alphabet de lettres, une base de chiffres

En informatique et en télécommunication, l'information est un élément de connaissance (voix, donnée, image) susceptible d'être conservé, traité ou transmis à l'aide d'un support et d'un mode de codification normalisé.

Ceci étant, quelque soit le type de l'information, les communicants voudraient s'assurer que des objectifs soient atteints à côté de la sécurité. Ces objectifs peuvent être résumés dans le tableau I-1 suivant :

Objectif	Description
Authentification	Vérification de la source de l'information.
Identification	Confirmation de l'identité d'une entité (e.g : une personne,un ordinateur, ou une carte de crédit)
Confidentialité	L'information n'est accessible que pour le/les destinataire(s) légitime(s)
Intégrité	L'information n'a pas été altérée lors de son transfert
Non-répudiation	Empêchant le démenti des engagements ou des actions précédents.
Signature	Moyen par lequel une information est liée à une entité.
Certification	Approbation d'information par une entité de confiance.
Accusé de réception	Confirmation que l'information a été reçue.
Anonymat	Cacher l'identité d'une entité impliquée dans un certain processus.
Propriété	Un moyen de fournir à une entité le droit légal d'employer ou transférer une ressource à d'autres.
Révocation	Rétraction de certification ou d'autorisation.
Témoignage	Vérifiant la création ou l'existence d'information par une entité autre que le créateur.
Times tamping	Enregistrant la période de la création ou l'existence d'information.
Validation	Un moyen de fournir l'opportunité de l'autorisation d'employer ou manœuvrer l'information ou des ressources.
Autorisation	Délivrer, à une autre entité, la permission officielle à faire ou être quelque chose
Confirmation	Reconnaissance que le service a été fourni.
Contrôles d'accès	Restriction de l'accès à des ressources, à des entités privilégiées.

Tableau I-1 : Les objectifs de la sécurité.

Ce tableau présente différents objectifs de la sécurité de l'information, mais les quatre qui constitue les piliers de cette sécurité sont les suivants :

- Confidentialité.
- Authentification et identification.
- Intégrité.
- Non – répudiation.

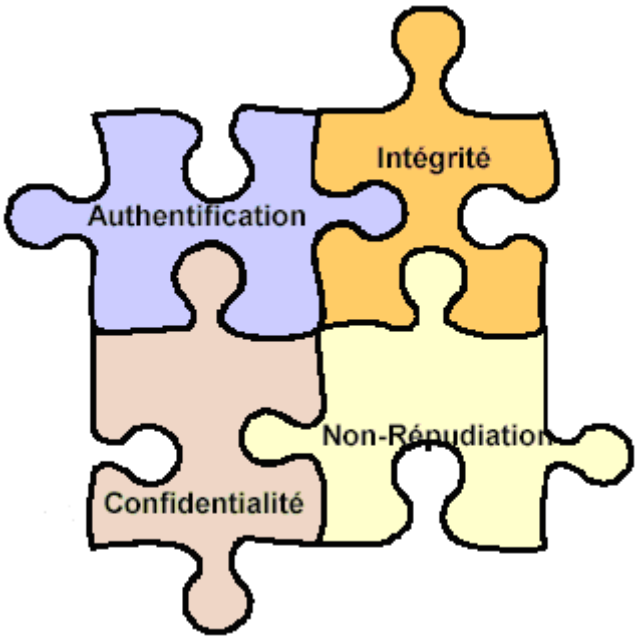


Figure I-1 : Principaux objectifs de la sécurité.

Confidentialité : les données reste inintelligibles à toute personne non autorisé.

Authentification : l'authentification consiste à vérifier l'identité des différents éléments impliqués dans un dialogue. Il peut s'agir d'identifier

- Une personne (par exemple l'expéditeur, destinataire,…).
- Une machine (dans le cadre d'une relation d'un client avec un serveur à travers un réseau).
- Un document (son auteur).

Intégrité : on doit éviter que les données transmises soient modifiées par un adversaire. Plus précisément, l'intégrité est la prévention d'une modification non autorisée de l'information. L'intégrité du système et de l'information garantit que ceux –ci ne sont modifiés que par une action volontaire et légitime.

Non-répudiation : c'est un mécanisme qui empêche de nier un contrat. L'auteur d'un message ne peut nier l'avoir écrit ou transmis.

I-4- Les menaces : pour construire un bon système cryptographique, il faut étudier les différents types d'attaques. On distingue les menaces de type passif, où l'attaquant est limité à l'écoute et l'analyse du trafic échangé, des menaces de type actif. Dans ce dernier mode, l'attaquant se donnera les moyens d'agir sur la gestion, la configuration et l'exploitation du réseau. Il peut injecter son propre trafic, modifier le fonctionnement d'un nœud, saisir l'identité d'un élément valide, rejouer/modifier des messages transitant sur le réseau ou provoquer un déni de service. L'attaque passive prive le réseau de la

confidentialité des messages échangés. Eventuellement, l'analyse du trafic représente un risque pour l'anonymat des participants et le respect de leur vie privée.

Avec la popularité grandissante des réseaux, des échanges de données, et donc des transmissions entre individus, de nombreuses menaces émergèrent. Parmi celles-ci, on trouve diverses catégories :

- Les menaces accidentelles
- Les menaces intentionnelles :
  - Passives.
  - Actives.

Les menaces accidentelles ne supposent aucune préméditation. Dans cette catégorie, sont repris les bugs logiciels, les pannes matérielles, et autres défaillances "incontrôlables".

Les menaces intentionnelles quant à elles, reposent sur l'action d'un tiers désirant s'introduire et relever des informations. Dans le cas d'une attaque passive, l'intrus va tenter de dérober les informations par audit, ce qui rend sa détection relativement difficile. En effet, cet audit ne modifie pas les fichiers, ni n'altère les systèmes.

Dans le cas d'une attaque active, la détection est facilitée, mais il peut être déjà trop tard lorsque celle-ci a lieu. Ici, l'intrus aura volontairement modifié les fichiers ou le système en place pour s'en emparer. Les menaces appartiennent principalement à quatre catégories (illustrées à la figure 1.2) :

– Interruption = problème lié à la disponibilité des données

– Interception = problème lié à la confidentialité des données

– Modification = problème lié à l'intégrité des données

– Fabrication = problème lié à l'authenticité des données

- **Types de menaces actives**

Les auteurs de ces attaques sont notamment les hackers (agissant souvent par défi personnel), les concurrents industriels (vol d'informations concernant la stratégie de l'entreprise ou la conception de projets), les espions, la presse ou encore les agences nationales.

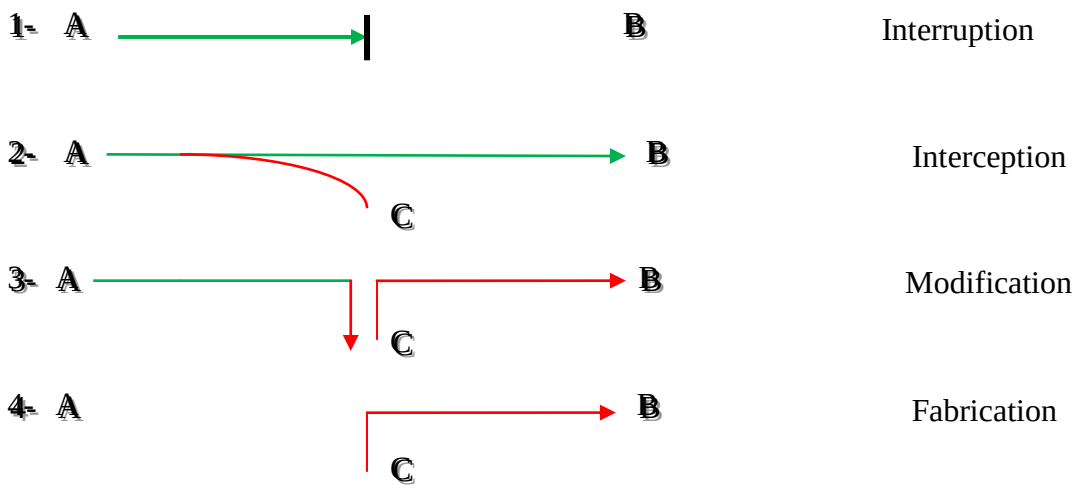


Fig I-2 : Menaces actives

I-5- Les types de chiffrement :

On peut classer ces méthodes en trois grandes classes, comme nous le montre le schéma qui suit :

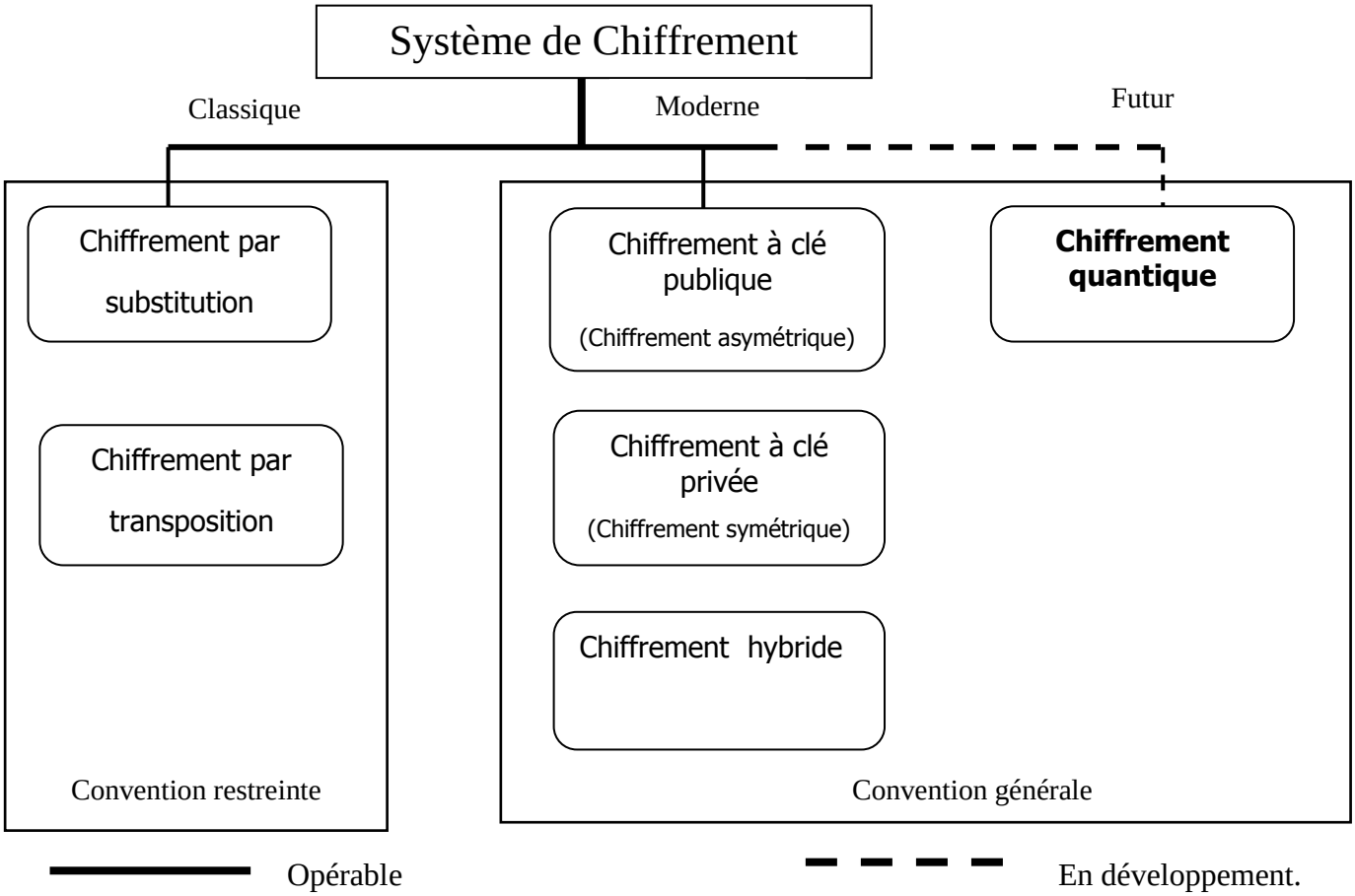
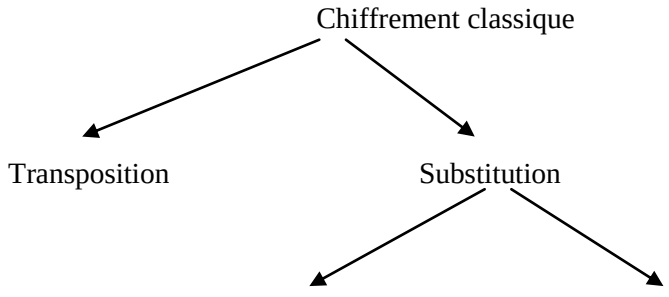


Fig. I-3 : Types de chiffrement.

I-5-1-Chiffrement classique : [Web 01]

Le chiffrement classique décrit la période avant les ordinateurs. Elle traite des systèmes reposant sur les lettres et les caractères d'une langue naturelle (allemand, anglais, français, etc...). Les principaux outils utilisés sont la substitution et la transposition. Les meilleurs systèmes répètent les deux opérations de base plusieurs fois, tout en gardant les opérations réalisées secrètes. Ici on va présenter quelques méthodes, mais plutôt les concepts mathématiques (connus depuis très longtemps) qui sont à la source de celles-ci. On appelle généralement cette classe de méthodes : le chiffrement à usage **restreint**.



Code Alphabet

Fig. I-4 : Types de chiffrement classique.

a) Substitution :

La substitution consiste à effectuer des dérivations pour que chaque caractère du message chiffré soit différent des caractères du message en clair. Le destinataire légitime du message applique la dérivée inverse au texte chiffré pour recouvrer le message initial. La complexité des systèmes à substitutions dépend de trois facteurs :

- la composition spécifique de l'alphabet utilisé pour chiffrer ou pour communiquer,
- le nombre d'alphabets utilisés dans le cryptogramme,
- la manière spécifique dont ils sont utilisés.

On distingue couramment quatre types de substitutions différentes :

a-1- Substitution simple ou substitution monoalphabétique :

Chaque caractère du texte en clair est remplacé par un caractère correspondant dans le texte chiffré. Les exemples les plus célèbres sont les algorithmes de César, Rot13, et bien évidemment le code morse. Ils sont encore utilisés aujourd'hui pour cacher le sens de certains messages (par exemple la solution de certains jeux dans des journaux), mais bien sûr elles sont très peu sûres. En effet avec ce principe, les lettres les plus fréquentes dans le texte en clair restent les plus fréquentes dans le texte chiffré, il ne cache donc pas les fréquences d'apparition des caractères. C'est une faiblesse importante puisque des techniques statistiques peuvent être utilisées pour associer aux lettres les plus fréquentes, une lettre probable et en appliquant une technique sémantique récursive, les algorithmes à base de substitutions monoalphabétiques sont facilement cassés par les spécialistes.

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

Tableau I-3 : Dictionnaire ROT13

Exemple : Texte en clair = « NON JE NE SUIS PAS FOU »

Texte chiffré (avec 5 divisions) = « ABAWR ARFHV FCNFS BH »

a-2- Substitution homophonique :

Pour échapper à l'analyse de fréquences, une solution consiste à remplacer une lettre non pas par un symbole unique, mais par un symbole choisi au hasard parmi plusieurs. Dans sa version la plus sophistiquée, on choisira un nombre des symboles proportionnel à la fréquence d'apparition de la lettre; on parle alors de renversement des fréquences. Ce type de substitution est appelé substitution homophonique (on dit aussi substitution à représentations multiples). On peut situer l'âge d'or de la substitution homophonique entre 1500 et 1750. [Web 02]

a-3- Substitution polyalphabétique :

Le principe consiste à remplacer chaque lettre du message en clair par une nouvelle lettre prise dans ou plusieurs alphabets aléatoires associés. Par exemple, on pourra utiliser n substitutions monoalphabétiques ; celle qui est utilisée dépend de la position du caractère à chiffrer dans le texte en clair. On choisit une clé qui sert d'entrée dans la grille polyalphabétique incluant autant de symboles qu'il y a de lettres différentes à chiffrer. Chaque caractère de la clé désigne une lettre particulière dans la grille de codage. Pour coder un caractère, on doit lire le caractère correspondant du texte en clair en utilisant la grille polyalphabétique et le mot clé associé dans l'ordre séquentiel (on répète la clé si la longueur de celle-ci est inférieure à celle du texte de départ). L'exemple le plus célèbre est le chiffre de Vigenère.

Exemple : On veut coder le texte "CRYPTOGRAPHIE" avec la clé "MATHWEB". On commence par écrire la clef sous le texte à coder :

Message clair	C	R	Y	P	T	O	G	R	A	P	H	I	E
Clé de chiffrement	M	A	T	H	W	E	B	M	A	T	H	W	E
Message chiffré	O	R	R	W	P	S	H	D	A	I	O	E	I

Tableau I-4 : Substitution polyalphabétique.

Pour coder la lettre C, la clé est donnée par la lettre M. On regarde dans le tableau l'intersection de la ligne donnée par le C, et de la colonne donnée par le M On trouve O. (Voir Tableau Vigenère)

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

Tableau I-5 : Tableau de vigenère.

Puis on continue. Sous le R on trouve le A. La lettre dans le tableau correspondant au R dans la colonne A est R. On remplace donc le R par lui-même. Pour le Y, la lettre de la clé est T. Dans la colonne T, le Y est remplacé par R. On continue ainsi et on trouve finalement le message chiffré : ORRWPSHDAIOEI.

a-4- Substitution par polygrammes :

Même principe que la substitution simple, sauf que l'on travaille ici par blocs de lettres auquel l'on fait correspondre un autre bloc. Les exemples les plus célèbres sont les chiffrements de Playfair et de HILL inventés en 1854 et utilisés pendant la première guerre mondiale par les anglais.

Le chiffre Playfair, chiffre polygrammique, a été popularisé par Lyon Playfair, mais il a été inventé par Sir Charles Wheatstone (1854). On dispose les 25 lettres de l'alphabet (W exclu car inutile, on utilise V à la place) dans une grille 5x5, ce qui donne la clef. La variante anglaise consiste à garder le W et à fusionner I et J.

On chiffre le texte par groupes de deux lettres (des bigrammes) en appliquant les règles suivantes:

1. Si les deux lettres sont sur les coins d'un rectangle, alors les lettres chiffrées sont sur les deux autres coins. Exemple **OK** devient **VA**, **BI** devient **DC**, **GO** devient **YV**. La première des deux lettres chiffrées est sur la même ligne que la première lettre claire.
2. Si deux lettres sont sur la même ligne, on prend les deux lettres qui les suivent immédiatement à leur droite: **FJ** sera remplacé par **US**, **VE** par **EC**.
3. Si deux lettres sont sur la même colonne, on prend les deux lettres qui les suivent immédiatement en dessous: **BJ** sera remplacé par **JL**, **RM** par **ID**.

4. Si le bigramme est composé de deux fois la même lettre, on insère une nulle (usuellement le X) entre les deux pour éliminer ce doublon.

Pour déchiffrer, on applique les règles ci-dessus à l'envers.

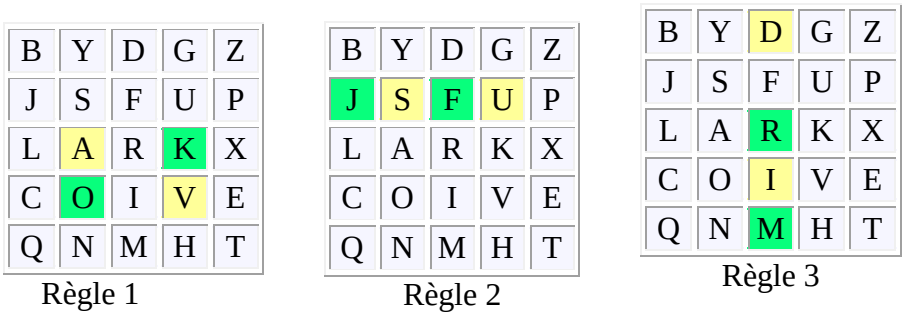


Fig. I-5 : Chiffrement par polygrammes

b- <u>Transposition</u> : Avec le principe de la transposition toutes les lettres du message sont présentes, mais dans un ordre différent. Il utilise le principe mathématique des **permutations**. Plusieurs types de transpositions existent :

b-1- Transposition simple par colonnes : on écrit le message horizontalement dans une matrice prédéfinie, et on trouve le texte à chiffrer en lisant la grille verticalement (Tableau I-5). Le destinataire légal pour décrypter le message réalise le procédé inverse. L'algorithme allemand ADFGFX est fondé sur ce principe et fut utilisé pendant la première guerre mondiale. Il fut cassé par une jeune étudiante française.

Exemple : Texte à chiffré = « IL ETAIT UNE FOIS EN KABYLIE ».

On utilise une matrice [6x4]

I	L	E	T
A	I	T	U
N	E	F	O
I	S	E	N
K	A	B	Y
L	I	E	

Tableau I-6 : Transposition simple par colonnes

Texte chiffré = « IANIKL LIESAI ETFEBE TUONY ».

b-2- **Transposition complexe par colonnes** : un mot clé secret (avec uniquement des caractères différents) est utilisé pour dériver une séquence de chiffres commençant à 1 et finissant au nombre de lettres composant le mot clé. Cette séquence est obtenue en numérotant les lettres du mot clé en partant de la gauche vers la droite et en donnant l'ordre d'apparition dans l'alphabet. Une fois que la séquence de

transposition est obtenue, on chiffre en écrivant d'abord le message par lignes dans un rectangle (comme le dessin ci-dessous le montre), puis on lit le texte par colonnes en suivant l'ordre déterminé par la séquence.

Exemple : Texte à chiffré = « IL ETAIT UNE FOIS EN KABYLIE ».

On utilise un mot clé : « PAUVRE »

3	1	5	6	4	2
P	**A**	**U**	**V**	**R**	**E**
I	L	E	T	A	I
T	U	N	E	F	O
I	S	E	N	K	A
B	Y	L	I	E	

Tableau I-7 : Transposition complexe par colonnes

Texte chiffré = « LUSYI OAITI BAFKE ENELT ENI ».

b-3- **Transposition par carré polybique** :[Web3]

Un mot clé secret est utilisé pour construire un alphabet dans un tableau. Les coordonnées des lignes et des colonnes correspondant aux lettres du texte à chiffrer sont utilisés pour transcrire le message en chiffres. Avec ce procédé chaque lettre du texte en clair est représentée par deux chiffres écris verticalement. Ces deux coordonnées sont ensuite transposées en les recombinants par deux sur la ligne ainsi obtenue.

Par exemple, le mot « bonjour » est ainsi chiffré par le carré de Polybe : 12 34 33 24 34 45 42. Pour déchiffrer un mot, il suffit d'effectuer la méthode inverse

	1	**2**	**3**	**4**	**5**
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Tableau I-8 : Carré de Polybe.

Il est important de faire remarquer que les transpositions sont plus contraignantes que les substitutions, car elles ont besoin de plus de mémoire et ne fonctionnent que sur des messages à chiffrer

d'une longueur limitée ; c'est pourquoi elles sont moins utilisées dans les algorithmes bien que pourtant un peu plus sûres que les substitutions.

I-5-2-Chiffrement moderne : [Pie-12]

Le chiffrement est le processus qui permet la transformation d'un message **m**, en un autre message **C**,

de telle sorte que le message résultant du processus ne puisse être compréhensible que pour le/les participant(s) légitime(s) d'une communication.

I-5-2-1- **Chiffrement symétrique** :(Chiffrement à clé secrète)

Le chiffrement symétrique, consiste en l'utilisation d'une seule et unique clé aussi bien pour le chiffrement que pour le déchiffrement. Ainsi, pour envoyer un message **m**, Alice le chiffre en utilisant la clé **K**, lorsque Bob reçoit le message **C**, il le déchiffre en utilisant la même clé **K** (Alice et Bob étant les émetteur/récepteur traditionnels du processus de chiffrement dans la littérature de la cryptologie) ; ainsi la confidentialité est assurée. D'autre part, si l'on assume que Alice et Bob sont les seuls possesseurs de la clé **K**, en recevant le message **C**, Bob est sûr de sa bonne provenance, puisqu'il est chiffré via la clé **K**, et seule Alice possède cette même clé, on gagne alors en matière d'authentification. Pour ce qui est de l'intégrité, et pour vérifier que le message n'a pas subi de changement durant sa transmission d'Alice vers Bob, il suffit d'appliquer un contrôle d'erreur externe ou interne toujours en se basant sur le chiffrement symétrique pour vérifier que le message n'a pas subi de modification.

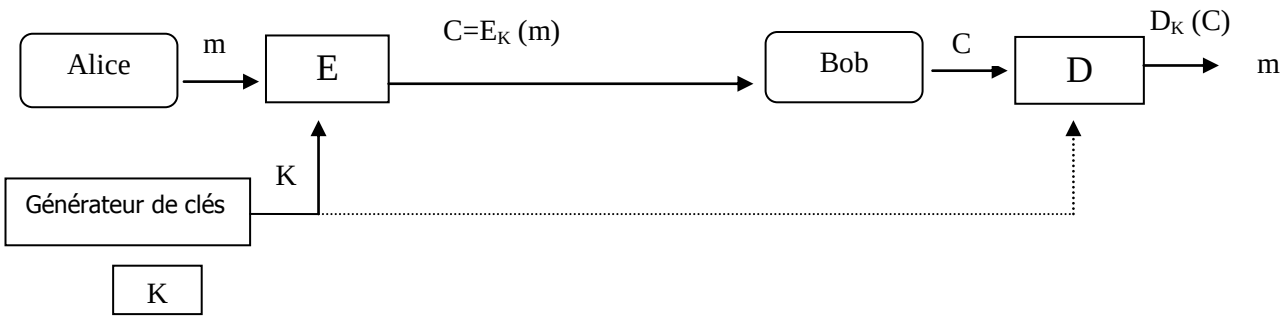


Fig. I-6 : Schéma classique d'un système de chiffrement symétrique.

Dans la figure précédente, on dispose de deux fonctions publiques :

- La fonction de chiffrement E qui à un texte clair m et une clé secrète K fait correspondre un texte chiffré C=E(m,K).
- La fonction de déchiffrement D qui à un texte chiffré C et une clé K fait correspondre le texte clair m=D(C,K).

Dans ce système de chiffrement l'échange de la clé secrète est la principale difficulté. Les cryptosystèmes symétrique se répartissent en deux familles : le chiffrement à flot et le chiffrement par bloc.

a- Chiffrement à flot : c'est un chiffrement continu qui agit sur le texte en clair un bit à la fois. Ce mode de chiffrement est encore appelé **chiffrement en flux** (Stream cipher en anglais)

b- Chiffrement par bloc : (Bloc cipher en anglais). Il opère sur le texte en clair par groupes de bits appelés blocs.

Parmi les algorithmes de chiffrement à clé secrète on peut citer : DES, 3DES, AES, Blowfish, RC (Rivest Cipher), IDEA, etc.

I-5-2-2- **Chiffrement asymétrique** :

Dans le cas du chiffrement asymétrique, il est question d'utiliser deux clés distinctes, l'une publique pour le chiffrement, l'autre secrète pour le déchiffrement. Bob chiffre alors le message m avec la clé $\mathbf{K_{Ap}}$, clé publique de Alice puis lui envoie le message. Alice est la seule à pouvoir déchiffrer le message puisqu'elle est la seule possesseur de la clé secrète $\mathbf{K_{As}}$. La confidentialité est donc assurée. Avec le chiffrement asymétrique, il est évident que l'expéditeur ne peut être identifié vu que tous les participants à la communication disposent de la clé publique du destinateur, on perd en matière d'authentification. L'inconvénient avec une telle manipulation réside dans le fait que rien ne garantie que le message ne sera pas déchiffré par un espion qui détient la clé publique.

Une autre manipulation pourrait être utilisée pour assurer à la fois confidentialité et authentification au prix d'un chiffrement asymétrique double ; Bob chiffre le message **m** via sa clé privée, puis chiffre le résultat via la clé publique d'Alice. A la réception Alice peut retrouver le message **m** par un premier déchiffrement via la clé privée d'Alice, puis un second déchiffrement via la clé publique de Bob. La confidentialité est assurée puisque seul Alice détient la clé de déchiffrement (sa clé privée), alors que l'authentification est assurée par la clé privée de Bob, utilisée lors du premier chiffrement. L'inconvénient avec cette manipulation réside dans la lenteur du procédé.

1-5-2-3- **Chiffrement hybride** (Combinaison Symétrique/Asymétrique) :

En pratique, une combinaison symétrique/asymétrique peut être utilisée ; Alice chiffre le message **m** en utilisant une clé secrète qu'elle partage avec Bob, on obtient ainsi **C**, qui sera chiffré à son tour par la clé publique de Bob. A la réception, Bob déchiffre en premier avec sa clé privée pour avoir **C** puis déchiffre encore une fois le résultat avec la clé privée du cryptosystème pour obtenir **m**. On assurera ainsi confidentialité et authentification tout en surpassant le problème de la lenteur imputé par un double chiffrement asymétrique.

Le procédé est schématisé comme suit :

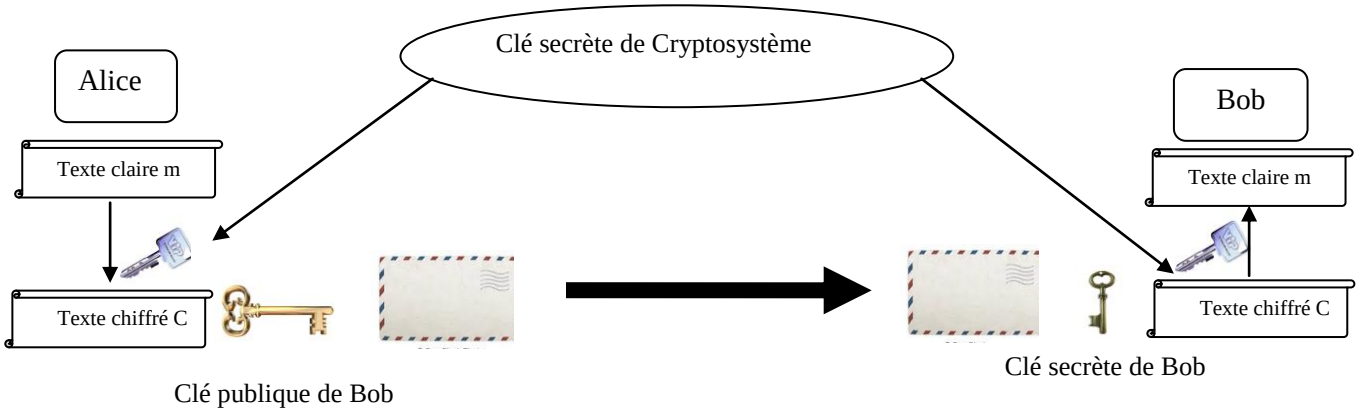


Figure 1-7 : Combinaison Symétrique/Asymétrique.

1-5-3- Chiffrement quantique :

La **cryptographie quantique** consiste à utiliser les propriétés de la physique quantique pour établir des protocoles de cryptographie qui permettent d'atteindre des niveaux de sécurité qui sont prouvés ou conjecturés en utilisant uniquement des phénomènes classiques (c'est-à-dire non-quantiques). Un exemple important de cryptographie quantique est la **distribution quantique de clés**, qui permet de distribuer une clé de chiffrement secrète entre deux interlocuteurs distants, tout en assurant la sécurité de la transmission grâce aux lois de la physique quantique et de la théorie de l'information. Cette clé secrète peut ensuite être utilisée dans un algorithme de chiffrement symétrique, afin de chiffrer et déchiffrer des données confidentielles.

Le principe de fonctionnement de ce système :

La cryptographie quantique repose sur trois domaines distincts :

– La cryptographie, en ce sens qu'elle permet de garantir la confidentialité d'une clé.

– La physique quantique, et plus particulièrement la mécanique quantique.

– La théorie de l'information, car elle fournit un système inconditionnellement sûr.

Le problème majeur en cryptographie est le transfert de la clé entre les deux parties communicantes.

Ici, on n'émet aucune hypothèse sur la sécurité du canal employé. La raison en est que la cryptographie quantique repose sur le principe d'Heisenberg :

- Certaines quantités subatomiques ne peuvent être simultanément mesurées.

La conséquence de ce principe est qu'il est impossible de mesurer ces particules sans les modifier.

Il est donc possible de construire un canal de communication que nul ne peut espionner sans modifier la transmission de manière détectable. Ainsi, il est possible de transmettre une clé secrète entre deux personnes sans qu'elles disposent d'informations secrètes communes préalables.

Dans le cadre du transport d'une clé, la technique qui nous occupera ici consistera en l'envoi de photons. On utilisera la technique dite de Polarisation de photons.

La cryptographie quantique est fondée sur l'utilisation de deux canaux :

- Un canal quantique par lequel transitent des objets régis par les lois de la mécanique quantique (il s'agit en général d'une fibre optique par laquelle on envoie des Impulsions lumineuses) .
- Un canal classique qu'Eve peut écouter sans restriction, mais ne peut pas modifier.

Des protocoles de cryptographie classiques permettent de réaliser un tel canal, authentifie de manière inconditionnellement sure : destinateur et le destinataire sont ainsi certains qu'ils se parlent bien l'un à l'autre. On ne peut pas empêcher Eve d'espionner le canal quantique, mais on peut savoir après la transmission si elle l'a fait. Il ne faut donc pas envoyer de message dans ce canal mais une suite d'éléments aléatoires, qui serviront ensuite a produire une clé s'ils n'ont pas été interceptes. Cette clé, parfaitement secrète, peut ensuite servir à chiffrer classiquement le message.

Conclusion :

Dans ce chapitre nous avons détaillé une importante composante de la cryptographie, à savoir le chiffrement. Les chiffrements sont subdivisés en deux grandes catégories ; les chiffrements symétriques où la clé du chiffrement et de déchiffrement sont identiques, et les chiffrements asymétrique qui, à la différence des premiers, possèdent une paire de clés chacune pour un procédé.

L'utilisation des algorithmes symétriques semble être plus facile, mais nécessite cependant que les deux parties communicantes aient la même clé, donc là c'est le problème de distribution de cette clé qui se pose. Par contre l'utilisation du chiffrement asymétrique ne pose pas une telle problématique,

Cependant, l'utilisation du chiffrement asymétrique est beaucoup plus gourmande en terme de temps.

Cette étude nous a permis de découvrir les avantages et les inconvénients de chaque type de chiffrement. Afin de remédier à leurs inconvénients et particulièrement le problème de distribution des clés, le quantique intervient dans le volet sécurité de l'information que nous aborderons dans le chapitre III.

Introduction :

Les algorithmes de chiffrement sont classés en deux catégories, les algorithmes symétriques et les algorithmes asymétriques.

II-1-Algorithme symétrique.

Les systèmes de cryptage à clé privée, appelés aussi systèmes de cryptage symétrique ou cryptage conventionnel, sont utilisés depuis plusieurs siècles déjà. C'est l'approche la plus authentique du chiffrement de données et mathématiquement la moins problématique.

Dans ce chapitre nous présentons les différents algorithmes utilisés dans le chiffrement.

II-1-1- **D.E.S** (Data Encryption Standard).

Il s'agit d'un système de chiffrement symétrique par blocs de 64 bits, dont 8 bits (un octet) servent de test de parité (pour vérifier l'intégrité de la clé). Chaque bit de parité de la clé (1 tous les 08 bits) sert à tester un des octets de la clé par parité impaire de '1' dans l'octet à qui il appartient .

La clé possède une longueur « utile » de 56 bits, ce qui signifie que seuls 56 bits servent réellement dans l'algorithme.

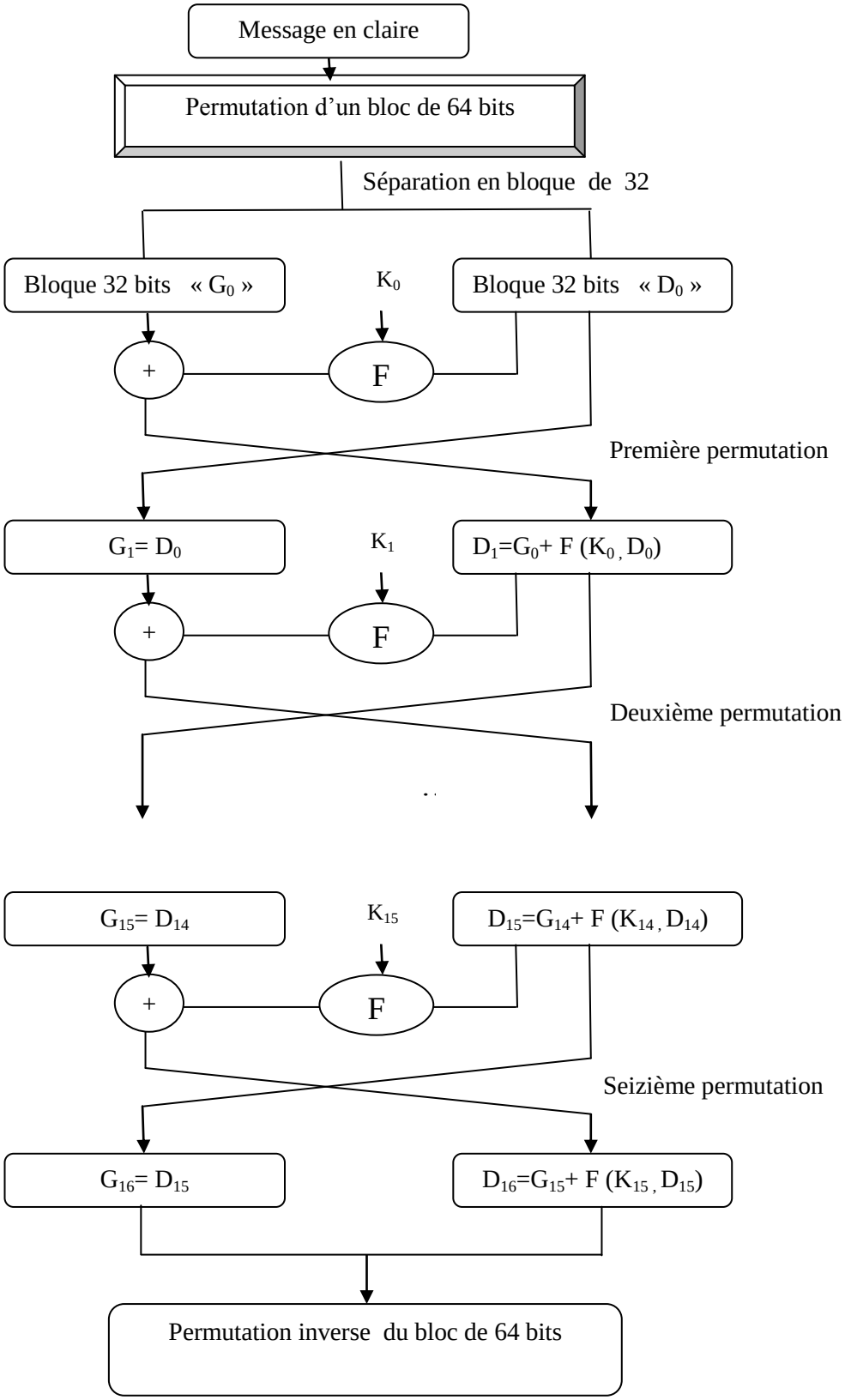
L'algorithme consiste à effectuer des combinaisons, des substitutions et des permutations entre le texte à chiffrer et la clé, en faisant en sorte que les opérations puissent se faire dans les deux sens (pour le déchiffrement). La combinaison entre substitutions et permutations est appelée code produit.

La clé est codée sur 64 bits et formée de 16 blocs de 4 bits, généralement notés k_0 à k_{15}. Etant donné que « seuls » 56 bits servent effectivement à chiffrer, il peut exister 2^{56} (soit $7.2*10^{16}$) clés différentes !

Les grandes lignes de l'algorithme sont les suivantes :

- Fractionnement du texte en blocs de 64 bits (8 octets) ;
- Permutation initiale des blocs ;
- Découpage des blocs en deux parties: gauche et droite, nommées G et D ;
- Etapes de permutation et de substitution répétées 16 fois (appelées rondes) ;

- Recollement des parties gauche et droite puis permutation initiale inverse. [Sou-08]



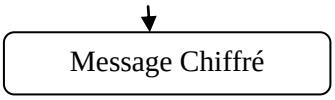


Figure II-1 : Schéma du D.E.S.

1-Transposition initiale

Chaque bit d'un bloc subit une permutation selon l'arrangement du tableau ci-contre c'est-à-dire que le 58ème bit du bloc se retrouve en 1ère position, le 50ème en seconde position, etc…

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Tableau II-1 :64 bits du message à chiffré

Permutation initiale

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	14	7

Tableau II-2 : Permutation initiale « PI »

2- Scindement en bloc de 32 bits

Le bloc de 64 bits est scindé en deux blocs de 32 bits notés G et D. On notera G_0 et D_0 l'état initial de ces deux blocs.

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8

Tableau II-3 : partie gauche du bloc « PI »

G_0 D_0

57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	14	7

Tableau II- 4 : partie droite du bloc « PI »

On remarque que G_0 contient tous les bits pairs du message initial et D_0 tous les bits impairs.

3- Rondes

Les blocs G_i et D_i sont soumis à un ensemble de transformations appelées rondes. Une ronde est elle-même composée de plusieurs étapes, dont le détail est donné à la figure II-2:

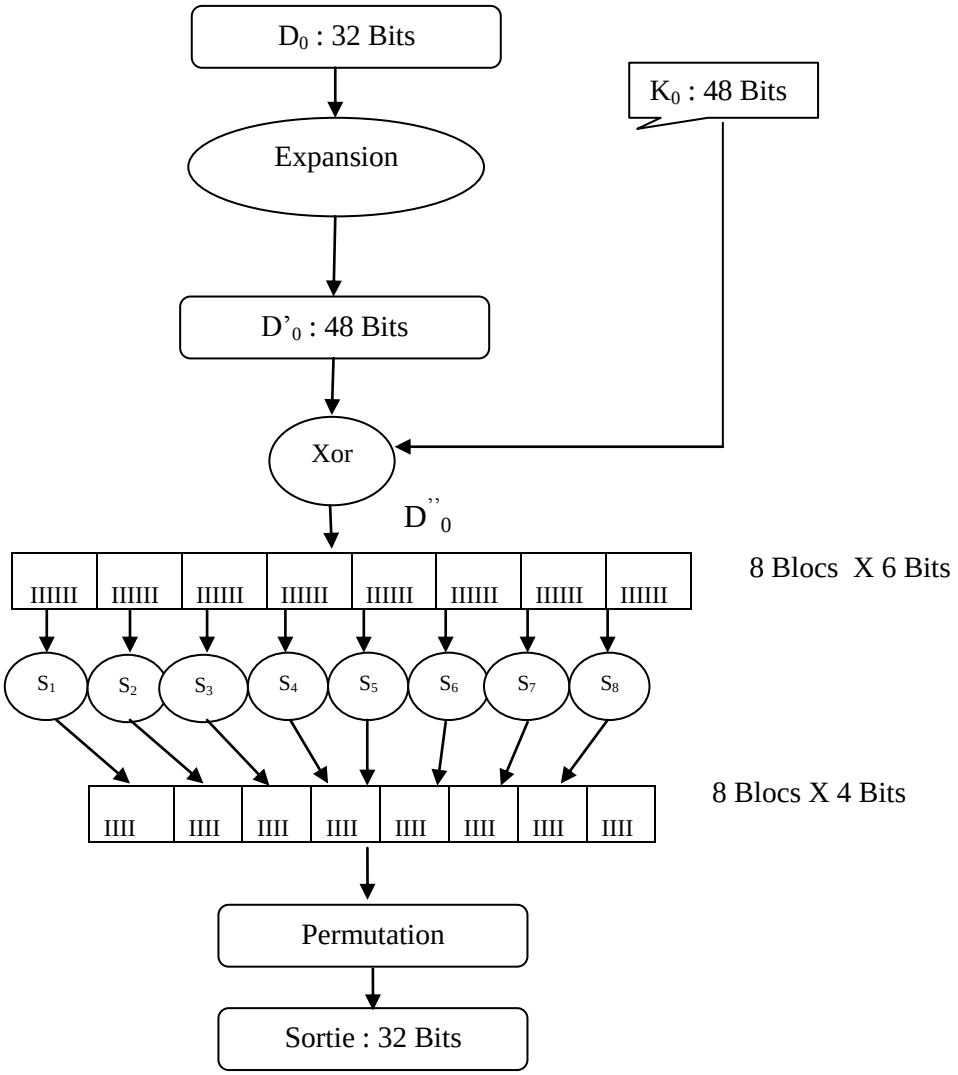
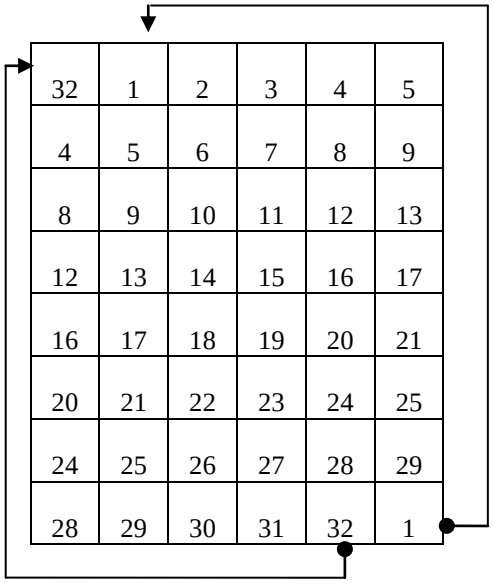


Figure II-2 : Schéma de la fonction « F ».

3-1-**Fonction d'expansion :**

Les 32 bits du bloc D_0 sont étendus à 48 bits grâce à une table d'expansion dans laquelle 32 bits sont mélangés et 16 d'entre eux sont dupliqués. Ainsi, le 32ème bit devient le premier, le premier devient le second… Les bits 1,4,5,8,9,12,13,16,17,22,21,24,25,28,29 et 32 sont dupliqués et disséminés pour former un bloc de 48 bits que l'on nommera D'_0 .



32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Tableau II-5: Représente la fonction d'expansion « E »

3-2- **OU exclusif (XOR) avec la clef :**

DES procède ensuite à un OU exclusif entre D'_0 et la première clef K_0 générée à partir de la clef K (que doivent se partager émetteur et destinataire) par l'algorithme de cadencement des clefs que nous écrirons plus bas. Nous appellerons D''_0 le résultat de cette opération.

Addition de la sous-clé : Le résultat de l'expansion est additionné (par une opération $\oplus$) à la sous-clé Kn correspondant à l'itération selon la formule :

$E(Ri-1) \oplus Ki = B1B2 \ldots B8$

Les B1,B2, ...,B8 sont des blocs de 6 bits :

Bj = b1b2b3b4b5b6.

3-3- Boîtes de substitution :

D''0 est découpée ensuite en 8 blocs de 6 bits, noté D''0i . Chacun de ces blocs passe par des boîtes de substitution(S-boxes), notées généralement Si. Les premier et dernier bits de chaque D0i

déterminent la ligne de la fonction de substitution, les autres bits déterminent la colonne. Grâce à cela la fonction de substitution « choisit » une valeur codée sur 4 bits (de 0 à 15).

Voici la première boîte de substitution :

Transformations par S-Boxes : Chaque bloc Bj constitue ensuite l'entrée de l'opération de substitution réalisée sur base des S-Box.

L'opération de substitution consiste pour chaque S-box à calculer :

– b1 b6 = N° de ligne

– b2 b3 b4 b5 = N° de colonne

Soit D_{0i} égal à 010101, les premiers et derniers bits donnent 01, c'est-à-dire 1 en binaire.

Les bits autres bits donnent 1010, soit 10 en binaire. Le résultat de la fonction de substitution est donc la valeur située à la ligne n°1, dans la colonne n°10. Il s'agit de la valeur 6, soit 0110 en binaire.

Chacun des 8 blocs de 6 bits est passé dans la boîte de substitution correspondante, les autres fonctions S-Boxes (S_2 ,S_3, S_4 , S_5 , S_6 , S_7 , S_8)

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Tableau II- 6 : représente la première fonction S-Boxe «S_1 »

Permutation initiale inverse :

A la fin des itérations, les deux blocs $\mathbf{G}_{16}$ et $\mathbf{D}_{16}$ sont "recollés, puis soumis à la permutation initiale inverse :

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	29	59	27
34	2	42	10	50	28	58	26
33	1	41	9	49	27	57	25

Tableau II-7 : permutation initiale inverse

Algorithme du calcul de la clé G(K, n) :

La clé est constituée de 64 bits dont 56 sont utilisés dans l'algorithme. Les 8 autres peuvent être utilisés pour la détection d'erreurs où chacun de ces bits sera utilisé comme bit de parité des 7 groupes de 8 bits. Ainsi, le nombre total de clés est de 2^{56}.

La clé initiale est de 64 bits. Le calcul a lieu en 4 étapes :

1) Permutation selon la table ci-dessous :

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	29	59	27
34	2	42	10	50	28	58	26
33	1	41	9	49	27	57	25

Tableau II-8 : Permutation pour l'algorithme de cadencement des clés dans le D.E.S.

2) Les 56 bits de la clé seront scindés en deux sous-clés chacune de 28 bits.

3) Les sous-clés vont subir un décalage d'un ou deux bits, suivant le numéro de la ronde, comme le montre le tableau :

Ronde	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Nombre de décalage	1	1	2	2	2	2	2	2	1	1	2	2	2	2	2	1

Tableau II- 9 : Rotation de l'algorithme de cadencement des clés du D.E.S.

4) Suite à cette opération les deux sous-clés seront recombinées pour former un bloc de 56 bits, ce même bloc subira encore une réduction puis une permutation de ses bits (*permutation compressive*) pour avoir la clé K_i de 48 bits comme le montre le tableau ci-dessous. La clé est maintenant de longueur égale à la longueur de D'_i, résultat de la fonction expansive.

14	17	11	24	1	5
3	28	15	6	21	10

23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	39	49	56	34	53
46	42	50	36	29	32

Tableau II-10 : Permutation compressive finale de K_i.

Un schéma récapitulatif du processus de génération des clés est donné dans la Figure II-3.

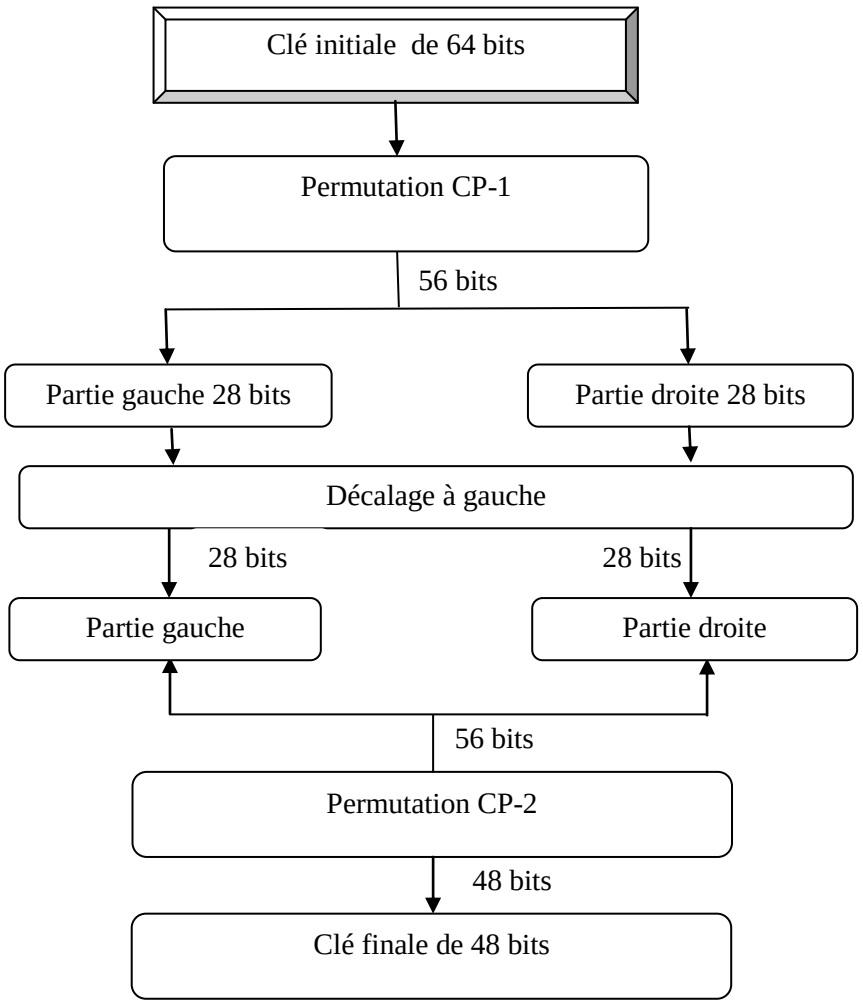


Figure II-3 : Schéma de l'algorithme de génération des clés

II-1-2- Autres Algorithmes symétrique

II-1-2-1- 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 (la longueur de la clé et de 168 bits) , comme le montre la figure suivante :

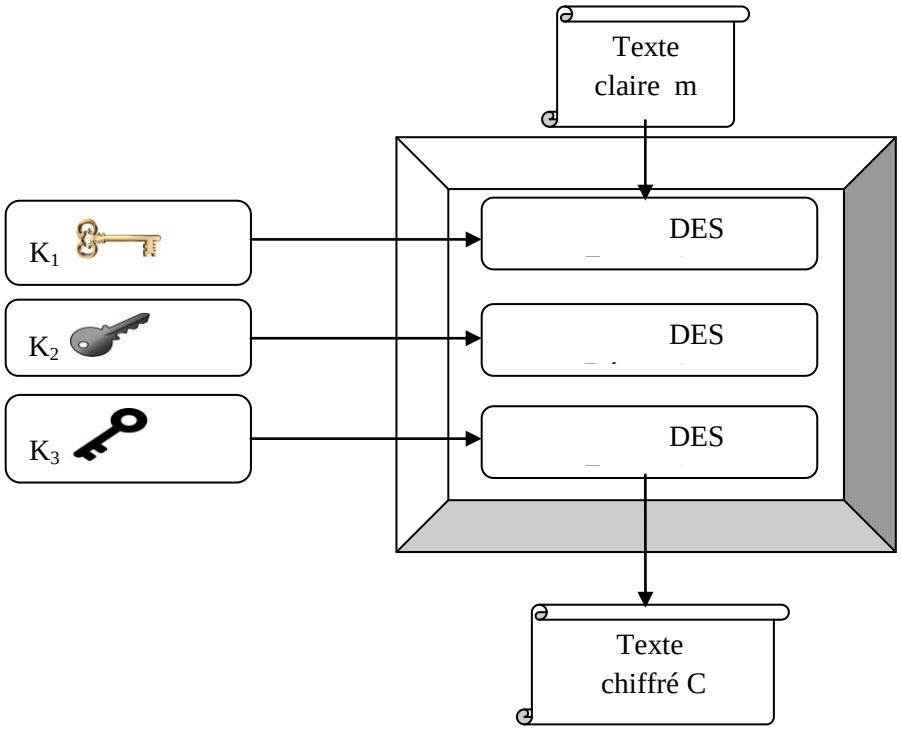


Figure II-4 : Schéma de l'**Algorithme Triple DES** avec ses trois opérations DES

Fonctionnement :

Cette utilisation de trois chiffrements DES a été développée par Walter Tuchman (chef du projet DES chez IBM). Le Triple DES est généralement utilisé avec seulement deux clés différentes. Le mode d'usage standard est de l'utiliser en mode EDE (Encryption, Decryption, Encryption, c'est-à-dire Chiffrement, Déchiffrement, Chiffrement) ce qui le rend compatible avec DES quand on utilise trois fois la même clé. Dans le cas d'une implémentation matérielle cela permet d'utiliser le même composant pour respecter

le standard DES et le standard Triple DES. Dans le mode proposé par Tuchman, 3DES s'écrit plus formellement de cette manière :

$$C = E_{DES}^{K3}\,[\,(E_{DES}^{K2}(\,E_{DES}^{K1}(m))]$$

K1,K2 et K3 : clés du chiffrement. E_{DES} : Fonction d'encryption

Bien que le Triple DES normalisé par le NIST, bien connu, et assez simple à implémenter, il est assez lent, et appelé à être remplacé par des algorithmes plus modernes tels qu'AES, également reconnu via le NIST aux États-Unis comme sûr pour tout échange d'information.

II-1-2-2- Algorithme A.E.S (Advanced Encryption Standard).

AES est le sigle d'**Advanced Encryption Standard** (en français, standard de chiffrement avancé). C'est l'algorithme Rijndael, du nom de leurs concepteurs Belges *Joan Daemen* et *Vincent Rijmen* [Sou 08]. Ce qui fait que, conscient des risques concernant DES, le NIST (National Institute of Standards and Technology) a demandé à la communauté cryptographique de réfléchir au successeur (AES),par un appel d'offre international lancé en janvier 1997. Il a été retenu par le NIST en octobre 2000 pour être l'algorithme AES, le nouveau standard de chiffrement pour les organisations du gouvernement des Etats-Unis, et ce, principalement pour des raisons de sécurité, performance, efficacité, facilité d'implémentation et flexibilité. De plus, son utilisation est très pratique car il consomme peu de mémoire.

L'AES procède par blocs de 128 bits, avec une clé de 128, 192 ou 256 bits. Chaque bloc subit une séquence de transformations que nous résumons à travers les points suivants :

1) Addition de la clé secrète et du bloc en question avec un ou exclusif.

2) Les 128 bits sont répartis en 16 blocs de 8 bits (16 octets), qui sont ensuite placés dans une matrice de 4×4 après leur permutation selon une table définie au préalable. Les lignes de cette matrice sont soumises à une rotation vers la droite où l'incrément pour la rotation varie selon le numéro de la ligne.

3) Chaque colonne est transformée par combinaisons linéaires des différents éléments de la colonne. Cela revient à multiplier la matrice 4×4 par une autre matrice 4×4.

4) Une clé dite de tour est générée à partir de la clé secrète par un sous-algorithme dit de cadencement. Cette clé de tour est ajoutée par un ou exclusif au dernier bloc obtenu. Ces différentes

opérations, définissant un tour, sont répétées plusieurs fois. Pour une clé de 128, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours.

II-1-2-3- Algorithme I.D.E.A :(International Data Encryption Algorithme)

L'International Data Encryption Algorithm (IDEA) , élaboré par Xuejic LAI et James MASSEY de l'Institut Technologique Fédéral Suisse. a été proposé en 1992 pour remplacer le DES. Il utilise une clé de 128 bits (contre seulement 56 bits pour le « DES »), réalise un chiffrement par blocs de 64 bits, en opérant 8 rondes d'une même fonction. Cette fonction utilise seulement 3 opérations :

- Ou exclusif.
- Addition modulo 2^{16}.
- Multiplication modulo $2^{16}+1$.

IDEA est particulièrement adapté aux réalisations logicielles. Cet algorithme est connu essentiellement car il a longtemps constitué la partie "cryptographie à clé secrète" du célèbre logiciel PGP.

Puissance de traitement en parallèle de 4 blocs.

II-1-2-4- Algorithme BlowFish :

Blowfish a été conçu par Bruce Schneier en 1993 comme étant une alternative aux algorithmes existants, en étant rapide et gratuit . Ce cryptosystème est sensiblement plus rapide que le DES. La grandeur de ses blocs est de 64 bits et il peut prendre une longueur de clé variant entre 32 bits et 448 bits Blowfish est basé sur un schéma de Feistel avec 16 tours et utilise des S-Boxes de grande taille qui dépendent de la clé . Ainsi, et depuis sa conception, il a été grandement analysé et est aujourd'hui considéré comme étant un algorithme de chiffrement robuste, mais il n'est pas breveté. Ainsi, son utilisation est libre et gratuite.[Sou-08].

II-1-2-5- Algorithme TwoFish :

Twofish est un algorithme de chiffrement symétrique par bloc inventé et analysé par Bruce Schneier, Niels Ferguson, John Kelsey, Doug Whiting, David Wagner et Chris Hall au sein du Counterpane Labs, pour participer au concours AES, où il a été l'un des cinq finalistes du concours sans pour autant être sélectionné pour le standard. Ce cryptosystème est conçu pour être très sûr et très flexible, en chiffrant des blocs de 128 bits avec une clé de 128,192 ou 256 bits, et en reprenant quelques concepts présents dans le Blowfish du même auteur. Cependant, Twofish est légèrement plus lent que Rijndael mais plus rapide que les autres finalistes de l'AES.

En 2005, Counterpane Labs a passé un long temps en évaluant Twofish, sans pouvoir trouver d'attaques possibles sur la version complète de Twofish, qui semble être plus sûre que la version initialement annoncée durant le concours AES. Malgré ça, il reste relativement peu utilisé.

II-1-2-6- Algorithme Serpent

Serpent, inventé par Ross Anderson, Eli Biham et Lars Knudsen, est un cryptosystème symétrique chiffrant des blocs de 128 bits. Il a été développé en vu d'être un Advanced Encryption Standard. Et malgré que le choix du NIST pour AES s'est porté sur Rijndael, mais ça n'empêche de signaler que Serpent et Rijndael sont similaires, et que la principale différence entre eux, est que Rijndael est plus rapide mais Serpent est plus sûr. De plus, aucune attaque connue n'a réussie à casser cet algorithme.

II-1-2-7- Les Algorithme de la famille RC (Ron's Code) :RC2,RC5 et RC6 [Web 04]

RC2 :

Est un chiffrement de bloc conçu par Ronald Rivest en 1987 ,utilise un bloc de 64 bits avec une clé de taille variable. Ses 18 tours sont arrangés selon un réseau de Feistel avec 16 tours de type Mixing englobés dans deux tours de type Mashing. Un tour Mixing comprend quatre transformations nommées MIX.

RC2 est vulnérable à une attaque par clé apparentée de John Kelsey qui nécessite 2^{34} textes clairs choisis. Elle fut publiée en 1997 peu après la diffusion du code sur Internet.

RC5 :

Est un chiffrement par bloc, fonctionnant grâce à une clé, dont la longueur varie de 40 à 2 040 bits. Il a été créé en 1994 par Ron Rivest pour la RSA Security [archive]. L'acronyme « RC » signifie « Ron's Code » ou « Rivest's Cipher ».

Il existe une variante **RC5P** qui utilise l'addition plutôt que XOR. Il existe une attaque basée sur la cryptanalyse Mod n pour RC5P.

RC6, basé sur RC5 fut candidat au concours pour devenir le standard actuel de chiffrement (AES).

Contrairement à de nombreux algorithmes, RC5 possède une taille variable de bloc 32, 64 ou 128 bits), une clef allant de 0 à 2048 bits et un nombre de tours de 0 à 255. Le chiffrement original suggère un choix de paramètres avec une taille de bloc de 64 bits, une clef de 128-bit et 12 tours.

RC6 :

Est un algorithme de chiffrement par bloc publié en 1998, et conçu au sein de la société RSA Security par Ron Rivest, Matt Robshaw, Ray Sidney et Yiqun Lisa Yin dans le cadre du concours AES, où il parvient à atteindre la finale aux cotés de quatre autres systèmes de chiffrement. Il est basé sur un bloc de 128 bits et supporte des clés de 128, 192 et 256 bits.

RC6 est similaire à RC5 dans sa structure de par la présence de rotations qui dépendent des données, les opérations d'addition modulaire et de XOR. En fait, RC6 pourrait être considéré comme deux chiffrements RC5 entrelacés. Une modification apparaît dans RC6 : il utilise une opération de multiplication absente de RC5. Cet ajout a pour but de rendre la rotation dépendante de chaque bit du mot, au lieu d'une dépendance concernant uniquement quelques bits de poids faible.

II-1-2-8- Algorithme MARS :

MARS est un algorithme de chiffrement symétrique par blocs créé par IBM comme algorithme pour le standard AES. Don Coppersmith était l'un des concepteurs de cet algorithme, qui prend en charge des blocs de 128 bits et des clés de dimensions variables entre 128 et 448 bits par incréments de 32 bits. Cet algorithme est unique, car il associe toutes les techniques de cryptage connues dans un seul produit. Ainsi, il utilise deux algorithmes séparés, de façon que si une partie de MARS est cassée, le reste des chiffres restera sécurisé et les données seront sauvegardées. De plus, MARS offre une meilleure sécurité que le triple DES et il est plus rapide que le DES.

II-2- Algorithmes asymétrique :

Plusieurs systèmes à clé publique ont été proposés. Leur sécurité repose sur divers problèmes calculatoire. Les plus connus sont les suivants :

II-2-1- **R.S.A** : (Rivest,Shamir,Adleman).

Ce cryptosystème tire son nom des noms de ses trois inventeurs : R. Rivest, A.Shamir, et L. Adleman. Ce système inventé en 1977, est le premier protocole de cryptographie à clé publique, présentant la cryptographie asymétrique. Il a été breveté par le MIT en 1983 aux États-Unis d'Amérique, mais le brevet a expiré le 21 septembre 2000.

Ces algorithmes utilisent une paire de clés différentes pour le chiffrement et le déchiffrement. L'une des clés est publique, l'autre doit rester secrète pour identifie son possesseur.

Description

Ce chiffrement est fondé sur la difficulté de factoriser un nombre entier qui est le produit de deux grands nombres premiers. La mise en œuvre du système de chiffrement RSA se fait de la manière suivante :

Lorsque Bob veut envoyer un message à Alice le processus se fait en trois étapes :

1- Alice prépare une clé publique « K_{PA} » et une clé privée « K_{SA} ».

2- Bob utilise la clé publique d' Alice pour crypter son message.

3- Alice reçoit le message crypté et le déchiffre grâce à sa clé privée.

1- Préparation des clés (génération des clés) :

- Choisir deux grands nombres premiers p et q.
- Calculer n le produit de ces deux nombre : n = p * q.
- Calculer le nombre d'Euler de n : Ø(n) = (p-1)(q-1).
- Choisir un nombre aléatoire e < Ø(n) et premier avec Ø(n) : PGCD (e, Ø(n)) = 1.
- Calculer l'inverse d de e modulo Ø(n) par l'algorithme d'Euclide étendu : d * e ≡ 1[mod Ø(n)].
- Les quantités n et e définie la clé publique d'Alice « K_{PA} », les quantités d et Ø(n) définie la clé secrète d'Alice « K_{SA} ».

2- Chiffrement du message :

Soit m le message en clair, le message chiffré est obtenu en calculant :

$C \equiv m^e$ [mod (n)].

3- Déchiffrement du message :

Pour retrouver le message en claire on procède comme suit :

$m = C^d$ [mod (n)].

La description du système est donnée dans la figure II-4

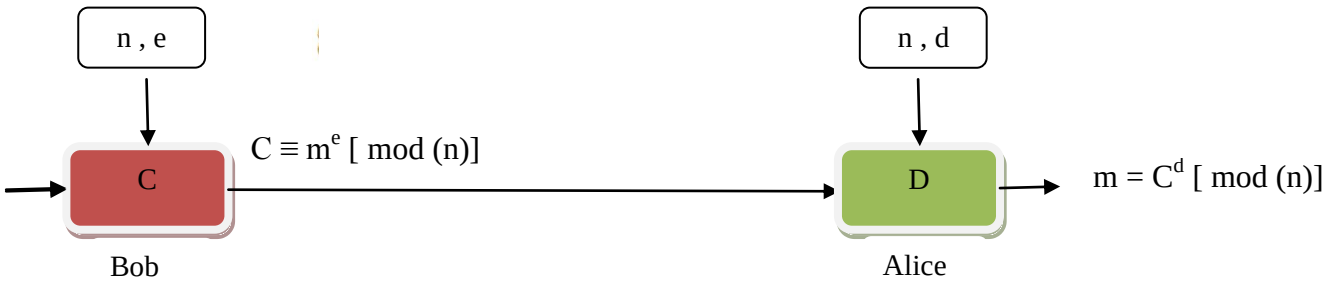


Figure II.4 Le chiffrement RSA

Exemple :

Bob veut transmettre un message secret à Alice, il commence à numériser son message en un (ou plusieurs) entiers m, tel que $0 \leq m < n$.

Choisir :

- p = 5 , q = 17.
- n = p * q = 85.
- Ø(n) = (p-1)(q-1) = 64.
- Choisir e de tel sorte PGCD (e, 64) = 1. Soit e = 5.
- Calculer l'inverse d : d * 5 ≡ 1[mod (64)]. Soit d = 13.

On prend dans notre exemple m = 10.

Le message chiffré s'obtient en calculant :

$C \equiv m^e$ [mod (n)] = 10^5[mod (85)].

- $10^2 = 100 \equiv 15$ [mod (85)].
- $10^4 = (10^2)^2 \equiv 15^2 = 225 \equiv 55$ [mod (85)].
- $10^5 = (10^4) * 10 \equiv 55 * 10 = 550 \equiv$ **40** [mod (85)].

Le message chiffré transmis par Bob : C = 40.

Déchiffrement du message :

Alice reçoit le message chiffré et le décrypte à l'aide de sa clé privée d = 13, grâce au calcul de $m = C^d$ [mod (n)]. Il doit calculer 40^{13} [mod (85)].

- $40^2 = 1600 \equiv 70$ [mod (85)].
- $40^4 = (40^2)^2 = (70)^2 = 4900 \equiv 55$ [mod (85)].
- $40^8 = (40^4)^2 = (55)^2 = 3025 \equiv 50$ [mod (85)].
- $40^{13} = (40)^8 * (40)^4 * (40) = 50 * 55 * 40 \equiv$ **10** [mod (85)].

On retrouve bien le message de Bob m= 10.

Efficacité et robustesse de RSA :

Une attaque évidente à ce système consiste à tenter de factoriser n, alors que, l'intérêt du système RSA repose sur le fait, qu'à l'heure actuelle, il est pratiquement impossible de retrouver dans un temps raisonnable p et q à partir de n si celui-ci est très grand. Donc, Alice est le seul qui peut calculer d dans un temps court, sans que cela nécessite la transmission des entiers p et q, ce qui empêche leur piratage. Mais, si une méthode de factorisation rapide sera développée, ce système serait aussitôt périmé. Ainsi, la sécurité de RSA semble satisfaisante malgré qu'il ne soit pas prouvé mathématiquement qu'on ne puisse pas le casser. En effet, en augmentant constamment la taille des clés, ce système reste très fiable si ses utilisateurs suivent les conseils des spécialistes, qui peuvent porter sur la taille des clés, la forme des nombres employés ou sur les méthodes d'implémentation. De même, le bon choix de p et q est aussi, un point crucial assurant la bonne sécurisation de ce cryptosystème.

II-2-2- **Autres Algorithmes asymétrique** :

II-2-2-1- Algorithme Diffie_Hellman : [Web 05]

Parallèlement à leur découverte du principe de la cryptographie à clé publique, Diffie et Hellman ont proposé en 1976 un protocole d'échange de clés totalement sécurisé. Le problème est le suivant. Alice et Bob veulent s'échanger un message crypté en utilisant un algorithme nécessitant une clé K. Ils veulent s'échanger cette clé K, mais ils ne disposent pas de canal sécurisé pour cela. Le protocole d'échange de clés de Diffie et Hellman répond à ce problème lorsque K est un nombre entier. Il repose sur l'arithmétique modulaire, et sur le postulat suivant :

Étant donnés des entiers p,a,x avec p premier et $1 \leq a \leq (p-1)$:

- Il est facile de calculer l'entier $y=a^x$ [mod (p)].
- si on connait $y=a^x$ [mod (p)], a et p, il est très difficile de retrouver x, pourvu que p soit assez grand.

Retrouver x connaissant a^x [mod (p)], a et p s'appelle résoudre le problème du **logarithme discret**. Comme pour la factorisation d'entiers, c'est un problème pour lequel on ne dispose pas d'algorithme efficace.

Expliquons maintenant comment Alice et Bob peuvent s'échanger une clé secrète par le protocole de Diffie-Hellman. Ils font des actions en parallèle, que l'on décrit dans le tableau suivant :

	Alice	Bob
Étape 1	Alice et Bob choisissent ensemble un grand nombre premier p et un entier $1 \leq a \leq (p-1)$. Cet échange n'a pas besoin d'être sécurisé.	
Étape 2	Alice choisit secrètement x_1.	Bob choisit secrètement x_2.
Étape 3	Alice calcule $y_1=a^{x1}$ [mod (p)] .	Bob calcule $y_2=a^{x2}$ [mod (p)]
Étape 4	Alice et Bob s'échangent les valeurs de y_1 et y_2. Cet échange n'a pas besoin d'être sécurisé.	
Étape 5	Alice calcule $y_2^{x1} = (a^{x2})^{x1} = a^{x1x2}$ [mod (p)] et appelle ce nombre K, la clé secrète à partager avec Bob.	Bob calcule $Y_1^{x2} = (a^{x1})^{x2} = a^{x1x2}$ [mod (p)] et appelle ce nombre K, la clé secrète à partager avec Alice.

À la fin du protocole, Alice et Bob sont donc en possession d'une même clé secrète K, qu'ils ne se sont pas échangés directement. Si quelqu'un a espionné leurs conversations, il connait p,a,y_1 et y_2. Il ne peut pas retrouver K comme le font Alice ou Bob, car il lui manque toujours l'une des informations

nécessaires, à savoir x_1 ou x_2. Et il ne peut pas retrouver x_1 connaissant $y_1=a^{x1}$ [mod (p)] , a et p, puisque la résolution du logarithme discret est un problème difficile.

Cette découverte de Diffie et Hellman est une vraie révolution dans l'histoire de la cryptographie. Le problème de l'échange des clés est en effet résolu. Ce protocole a cependant un défaut : il exige la simultanéité des actions d'Alice et de Bob. Si Alice veut envoyer un e-mail à Bob alors que celui dort ou n'est simplement pas connecté, elle ne pourra pas le faire immédiatement. C'est pourquoi ce protocole fut en réalité très vite supplanté par les méthodes de chiffrement à clé publique de type RSA, pour lesquels on met à la disposition de tout le monde une clé publique. Toutefois, il est utilisé pour les problèmes d'appariement de deux objets dans la technologie Bluetooth.

II-2-2-2- **Algorithme El Ghamal** : .

Le chiffre d'El-Gamal est une méthode de cryptographie à clé publique inventée par Taher ElGamal en 1985. Sa sécurité repose, comme le protocole de Diffie et Hellman, sur la difficulté de calculer le logarithme discret. Le destinataire, Bob, possède deux clés :

- une clé secrète : un entier s.
- une clé publique, qui consiste en un nombre premier p, un entier a compris entre 1 et (p−1), et l'entier $P = a^s \mod (p)$.

Si Bob veut envoyer le message m à Alice, m étant un entier compris entre 1 et (p−1) elle procède de la façon suivante : elle tire au hasard un nombre k, et calcule :

$C_1= a^k \mod (p)$ et $C_2= mp^k \mod (p)$.

Le message chiffré est le couple (C1,C2) qu'elle transmet à Alice. A la réception, celui-ci calcule

$R1=C_1^{\,s} \mod (p) = a^{sk} \mod (p) = P^k \mod (p)$

Puis il retrouve m par la formule

$m = C_2R_1^{\,-1} \mod (p) = m\, P^k\, P^{-k} \mod (p)$

La robustesse de ce système repose sur le fait que, si quelqu'un espionne la conversation d'Alice et Bob, il a en sa possession a, p, $P=a^s$, a^k et m P^k. Pour retrouver m, il doit savoir calculer $P^k \mod (p)$. Ceci impose de trouver k et donc il doit savoir résoudre l'équation suivante (en k) $y=a^k \mod (p)$ pour n'importe quel entier y. On appelle ce problème le calcul du logarithme discret modulo p. C'est un problème pour lequel on ne dispose pas d'algorithme rapide.

Un autre avantage du chiffre d'El-Gamal est qu'il s'agit d'un **chiffrement aléatoire** (on dit aussi **randomisé**). En effet, lorsqu'elle veut chiffrer m, Bob commence par choisir au hasard un entier k. En particulier, si elle envoie deux fois le même message, il n'y a pas de raisons qu'elle choisisse deux fois

au hasard le même entier k. Ainsi, elle enverra deux messages chiffrés différents pour le même message clair, sans que cela n'influe sur le déchiffrement.

Le défaut du système d'ElGamal est que le message chiffré est deux fois plus long que le message original. En revanche, le fait d'utiliser un paramètre aléatoire k est un plus en termes de sécurité : le même message m chiffré à 2 moments différents donnera deux messages codés distincts! Ce système est peu utilisé comme méthode directe de cryptographie. En revanche, il est très utilisé dans les procédés de signature électronique, souvent avec des groupes plus compliqués que $(Z/pZ)^*$, mais dans lesquels il est néanmoins très compliqué de résoudre le problème du logarithme discret.

II-2-3- **Comparaison entre les cryptosystèmes symétriques et asymétriques :**

Le tableau ci-dessous présente une comparaison entre les systèmes de chiffrement symétriques et les systèmes de chiffrement asymétriques, en énumérant les principaux avantages et inconvénients de chaque mode de cryptage

Méthode	Exemples	Avantages	Inconvénients
À clefs Secrètes	DES, AES	- Rapidité de calcul en général (dépend de la taille de la clé). - Adaptée au cryptage de flux de données	- Moins sécurisé (DES). - Problème de communication de clefs entre émetteur et récepteur. - Une clé pour chacun des correspondants : n personnes => n(n-1)/2 clés.
À clefs Publiques	RSA, ElGamal	- Très sécurisée à cause de l'utilisation de deux clés distinctes, l'une ne permettant pas de retrouver l'autre. - Permet la signature électronique. - Un couple de clés publique/privée suffisant pour 'n' correspondants.	- Lente. - Problèmes de gestion de clefs publiques

.

Tableau II-11- Comparaison entre les méthodes de chiffrement symétriques et asymétriques.

II-2-4- **Clé publique ou clé secrète, un compromis**

La question qui se pose à ce niveau est : Dans quels cas on utilise le chiffrement symétrique ? Et dans quels autres cas le chiffrement asymétrique est conseillé ? En effet, et d'après la table comparative présentée juste avant, il est clair que les systèmes de chiffrement à clé publique sont très lents par rapport aux systèmes de chiffrement à clé privée. Alors que, l'algorithme de chiffrement ne doit pas être le facteur limitant à notre époque où la vitesse de transmission de l'information constitue un enjeu crucial.

De plus, et à partir des descriptions des systèmes de chiffrement présentées précédemment, on arrive à constater que la taille des clés nécessaire en cryptographie à clé publique pour assurer une sécurité satisfaisante est plus grande que la taille des clés en cryptographie à clé secrète. En fait, la notion et l'importance de la taille de clé pour assurer la sécurité ne sont légitimes que dans le cas de la clé secrète, puisque ces systèmes reposent sur l'hypothèse que les seules attaques possibles sont les attaques exhaustives. Mais, dans le cas de la clé publique, la taille de clé n'a de pertinence que lorsqu'on considère le même système. Donc, le fait de dire que RSA de 512 bits est bien moins sûr qu'un AES de 128 bits, n'a aucune signification. Cependant, la seule mesure légitime pour évaluer un cryptosystème à clé publique est la complexité de la meilleure attaque connue.

II-2-5- **Algorithmes de chiffrement hybrides :**

II-2-5-1- **PGP :**

Philip Zimmermann, qui est un mathématicien passionné par l'informatique, et en croyant à la philosophie qui dise que tout individu a droit à la confidentialité, a commencé à travailler en 1984 sur un système cryptographique aussi sûr mais plus souple que le RSA. Ainsi, il a développé le **PGP** (**Pretty Good Privacy**) en 1991, puis il l'a mis à disposition gratuitement sur Internet sans se préoccuper des détails juridiques qui concernent son utilisation de RSA sans l'accord de son propriétaire, ou de son vendeur, ViaCrypt. Après quelques négociations commerciales et trois ans de menaces judiciaires par le gouvernement américain, PGP est à nouveau accessible depuis 1993, mais à 150 $ cette fois-ci et vendu par ViaCrypt. Maintenant, les principes et formats de messages utilisés par PGP ont été normalisés à l'IETF sous le nom **OpenPGP**.[Sou 08].

Principe

Lorsqu'un utilisateur chiffre un texte avec le système de chiffrement hybride PGP combinant des fonctionnalités de la cryptographie à clé publique et de la cryptographie symétrique, les données sont d'abord compressées. Cela à pour objectif de réduire le temps de transmission de ces données, et d'économiser l'espace disque et, surtout, le renforcement de la sécurité cryptographique du moment où,

les cryptanalystes exploitent les modèles trouvés dans le texte en clair pour casser le chiffrement ; alors que la compression réduit ces modèles dans le texte en clair. Par conséquent, la résistance à la cryptanalyse sera, considérablement, améliorée. Ensuite, l'opération de chiffrement se fait principalement en deux étapes, qu'on résume comme suit :

- PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ;
- PGP crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire.

L'opération de déchiffrement se fait également en deux étapes, qui sont :

- PGP déchiffre la clé secrète IDEA au moyen de la clé RSA privée.
- PGP déchiffre les données avec la clé secrète IDEA précédemment obtenue.

II-2-5-2- L'**IDEA :** (**International Data Encryption Algorithm**), qui est un cryptosystème symétrique inventé en 1992, effectue des opérations du même genre que celles vues avec l'algorithme DES, et il manipule des blocs de 64 bits avec une clé de 128 bits. En combinant les deux modes de cryptage symétrique et asymétrique, cette méthode de chiffrement profite des avantages de ces deux modes, à savoir la simplicité et la facilité d'utilisation du cryptage asymétrique et la rapidité de calcul du cryptage symétrique. De plus, le cryptage asymétrique résout le problème de la distribution des clés. Ainsi, les performances seront améliorées sans que la sécurité soit affectée.

II-2-5-3- **GPG :**

GPG (Gnu Privacy Guard) est dans le principe un clone de PGP, ou plus exactement une implémentation de l'OpenPGP, mais n'utilise aucun code de PGP . Donc, c'est l'équivalent libre de PGP. Il est entièrement écrit par des développeurs bénévoles, et est complètement libre, sous licence GPL (General Public License). Ainsi, il est remis à jour continuellement, aussi bien au niveau des fonctionnalités, qu'au niveau des éventuels problèmes d'implémentation.

Conclusion :

A l'heure actuelle l'utilisation du DES est simplement déconseillée à cause de la grande puissance de calcul assurée par les ordinateurs les plus récents. Toutefois le triple DES, permet d'apporter un niveau de sécurité acceptable et de résister aux attaques les plus classiques. Le choix de l'AES reste néanmoins le meilleur choix, dans l'attente d'un remplaçant ou d'une méthode d'attaque efficace qui va le remettre en cause.

Les calculs faits en 1995 ont ouvert un vaste horizon devant le chiffre RSA, du fait que le cassage des clés de 130 chiffres, utilisées à l'époque, nécessite 150 ans. Alors que va-t-on dire avec les clés utilisées aujourd'hui, qui comportent plus de 300 chiffres et qui sont donc plusieurs milliards de fois supérieures ? Donc, la méthode est officiellement sûre si l'on respecte certaines contraintes de longueur de clés et d'usage. Toutefois, personne depuis 2500 ans n'a trouvé de solution rapide au problème de la factorisation, alors il est tout à fait clair, que seule une véritable révolution mathématique ou informatique

serait capable de remettre en cause ce cryptosystème. De même, casser l'algorithme ElGamal est dans la plupart des cas au moins aussi difficile que de calculer le logarithme discret.

Cependant, il est possible qu'il existe des moyens de casser l'algorithme sans résoudre le problème du logarithme discret. Signalons aussi que le désavantage d'ElGamal par rapport à RSA réside dans le fait que le message chiffré est deux fois plus gros que le clair.

Introduction :

Suivant ce qui vient d'être introduit dans les chapitres précédents, il est clair que la cryptographie qui se voit attribuer le titre « de science du secret » par excellence, est, et ce depuis son apparition, en évolution perpétuelle afin de satisfaire les différents besoins en matière de sécurité ; confidentialité, authentification,… . Cela dit, les différentes techniques de cryptage actuelles sont basées soit sur le partage d'un secret, soit sur la complexité d'opérations mathématiques, tel que la factorisation d'un grand nombre, et dont l'impossibilité de résolution n'est nullement prouvée jusqu'à ce jour, ce qui pousse à la réflexion ! Si l'on admet qu'un jour les mathématiciens réussissent à trouver une manière plus simple et plus rapide, pour résoudre des problèmes mathématiques tel ceux sur lesquels ce base les algorithmes de cryptage asymétriques, ou qu'un ordinateur quantique serait disponible à de moindres frais, alors toute la sécurité des cryptosystèmes à clé publique serait remise en question !

Une seule technique semble ne pas être menacée par l'évolution en matière de puissance de calcul, c'est la technique du one Time Pad. Elle consiste, comme nous l'avons introduit dans les chapitres précédents, en l'utilisation d'une clé de chiffrage/déchiffrage aussi longue que le message et dont les éléments sont aussi aléatoires que possible, mais qui sera utilisée une et une seule fois, et qui serait connue par tous les participants à la communication ; cela revient donc à partager cette clé au préalable

entre les parties prenantes de la communication. Problématique connue le plus souvent sous la dénomination de distribution de clé. Mais cette distribution même, se doit d'être fiable et sécurisé de sorte à ce que la moindre intrusion ou tentative d'interception de la clé lors de son échange serait facilement décelée.

La cryptographie quantique, du fait qu'elle se base sur l'usage des états quantiques des photons, semble être la discipline, qui pourrait résoudre cette problématique de distribution de clé comme nous le verrons dans la suite de ce chapitre. Nous commençons tout d'abord par introduire quelques notions de bases du monde quantique.

III-1- Quelques propriétés de l'information quantique.

III-1-1- **Le qubit** : En informatique quantique, un **qubit** (quantum + bit), parfois écrit **qbit**, est l'état quantique qui représente la plus petite unité de stockage d'information quantique. C'est l'analogue quantique du bit.

Superposition d'états : Le qubit se compose d'une superposition de deux états de base, par convention nommés $<0|$ et $<1|$. Un état qubit est constitué d'une superposition quantique linéaire de ces deux états. Une mémoire à qubits diffère significativement d'une mémoire classique. Un bit classique se trouve toujours soit dans l'état « 0 », soit dans l'état « 1 ». Dans le cas général, un qubit se trouve dans une superposition de ces deux états, que l'on peut décrire par une combinaison linéaire des deux états : $\alpha <0| + ß <1|$. Les coefficients α et $ß$ étant deux nombres complexes vérifiant la relation de norme $|\alpha|^2 + |ß|^2 = 1$.

En théorie, on peut alors transmettre une infinité d'informations avec un qubit en mettant l'information dans l'angle de polarisation d'un qubit, cet angle étant réel. Cependant on ne peut pas récupérer cette information lors de la lecture.

Mesure :

Lors de la mesure de la valeur du qubit, les seules réponses pouvant être obtenues sont $<0|$ ou $<1|$ (avec les probabilités $|\alpha|^2$ et $|ß|^2$). Après une mesure, le qubit se trouve projeté dans l'état mesuré.

III-1-2- **Intrication :**

L'intrication quantique est un phénomène observé en mécanique quantique dans lequel l'état quantique de deux objets doit être décrit globalement, sans pouvoir séparer un objet de l'autre, bien qu'ils puissent être spatialement séparés. Lorsque des objets quantiques sont placés dans un état intriqué, il existe des corrélations entre les propriétés physiques observées de ces objets qui ne seraient pas présentes si ces propriétés étaient locales. En conséquence, même s'ils sont séparés par de grandes distances spatiales, deux objets intriqués O_1 et O_2 ne sont pas indépendants et il faut considérer $\{O_1+O_2\}$ comme un système unique.

L'intrication quantique a un grand potentiel d'applications dans les domaines de l'information quantique, tels que la cryptographie quantique, la téléportation quantique ou l'ordinateur quantique, elle est au cœur des discussions philosophiques sur l'interprétation de la mécanique quantique. Les corrélations prédites par la mécanique quantique, et observées dans les expériences, montrent que la nature n'obéit pas au principe du « réalisme local » chez Einstein, selon lequel les propriétés observées d'un système, bien définies avant toute mesure, sont attribuables à ce système et ne peuvent changer que par interaction avec un autre système.

III-1-3- Non-clonage, distinguable et perturbation

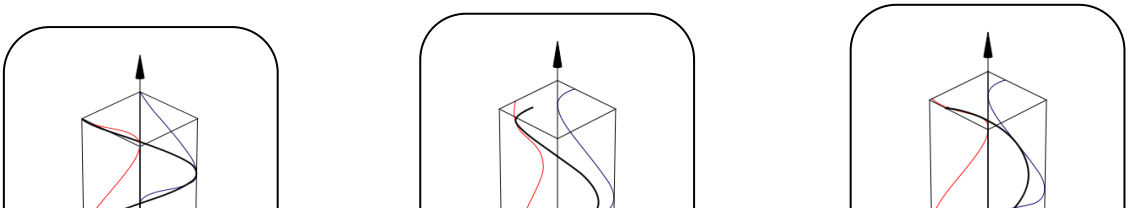
Photon : [Sti-10]

Le photon est la particule élémentaire (quantum) qui constitue le rayonnement. L'état quantique dans ce contexte désigne l'ensemble des caractéristiques qui aident à décrire le photon : sa position, son énergie, sa polarisation, etc.… ici, on ne s'intéressera qu'à sa polarisation. Un photon peut ainsi être considéré comme étant un minuscule champ électrique oscillatoire. La direction de l'oscillation définit alors la polarisation du photon.

Polarisation d'un photon :

Dans une onde électromagnétique, le champ électrique et le champ magnétique oscillent simultanément dans des directions différentes. Par convention, la polarisation de la lumière décrit la vibration du champ électrique E dans le plan orthogonal à la direction de propagation. Ce champ peut osciller dans une seule direction (polarisation rectiligne) ou peut tourner autour de l'axe de propagation de l'onde. On parle dans ce cas de polarisation circulaire ou elliptique (Figure III-1).

Toutefois, la lumière incohérente (lumière naturel) peut aussi présenter, en partie, une polarisation elliptique, circulaire, ou rectiligne. Pour déterminer alors quelle est la polarisation d'une lumière, on utilise des filtres polariseurs



Polarisation rectiligne | Polarisation elliptique | Polarisation circulaire

Figure III-1 : Trois types de lumière polarisée

Dans la suite nous travaillons avec des photons uniques polarisés de façon linéaire (Figure III-2) et selon quatre angles différents avec l'axe (Oy). La polarisation est mesurée par un angle qui varie de

0° à 180°. Habituellement la polarisation peut prendre quatre valeurs (Figure III-3) 0°,45°,90° et 135°.

Pour les photons polarisés de 0° à 90°, on parle de polarisation rectiligne, pour ceux polarisés de 45° à 135°, de polarisation diagonale. Chaque état polarisé représente une valeur binaire et par convention, on applique la valeur 0 à la polarisation horizontale et diagonale alors la valeur 1 sera appliquée à la polarisation verticale et anti-diagonale. [Web 6]

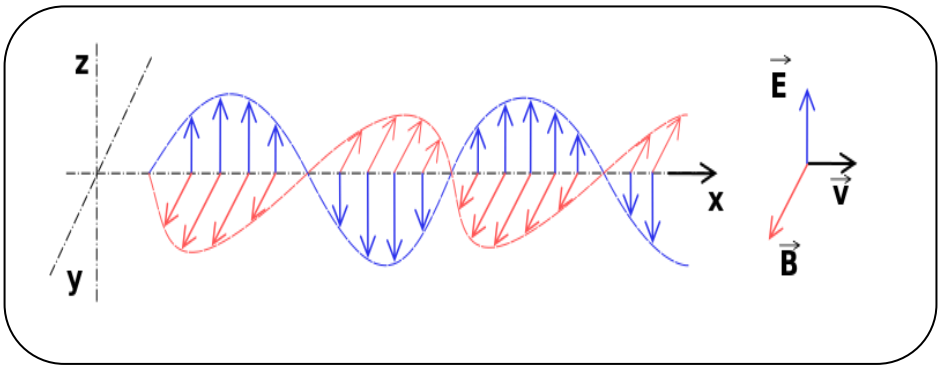


Figure III-2 : Cas d'une polarisation rectiligne

Nom	Horizontal	Diagonal	Vertical	Antidiagonal
Angle de	0°	45°	90°	135°

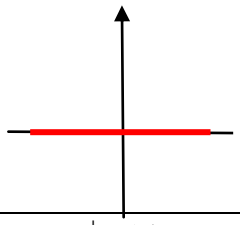
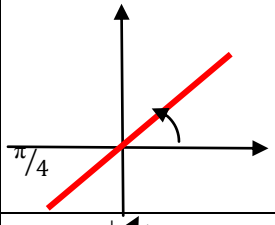
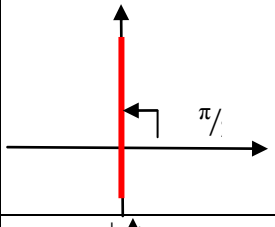
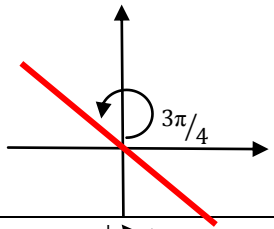
polarisation				
Polarisation				
Vecteur unitaire	|→>	|↗>	|↑>	|↖>

Figure III-3 : Polarisation des photons

Détection de la polarisation des photons :

Pour pouvoir détecter la polarisation des photos, on utilise un filtre polarisant suivi d'un détecteur de photons (figure III-4).

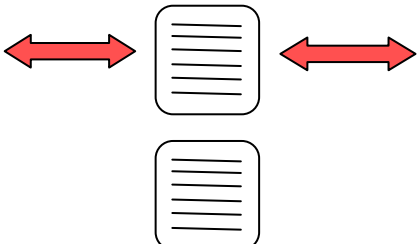
1- Si un photon polarisé à 0° rencontre un filtre polarisant orienté à 0°, il traverse ce filtre polarisant et est enregistré par le détecteur placé juste après (le photon est polarisé parallèlement à l'angle d'orientation du filtre).

2- Si un photon polarisé à 90° rencontre le même filtre, il est immédiatement stoppé (absorber), et le détecteur n'enregistre rien (le photon est polarisé perpendiculairement à l'angle d'orientation du filtre).

De la même façon, on peut utiliser un filtre polarisant orienté à 45° : il laisse passer les photons polarisés à 45°, stoppe ceux polarisés à 135°.

3- Si le photon est polarisé selon une direction intermédiaire, alors ce photon sera transmis avec une probabilité $\cos^2(\alpha)$, ou α est l'angle de polarisation du photon mesuré par rapport à l'angle d'orientation du filtre. C'est-à-dire que, si le photon est polarisé selon un angle Ω et que le filtre orienté selon un angle ß, alors α= Ω-ß. Si le photon est transmis, alors sa nouvelle polarisation correspondra à l'angle d'orientation du filtre.

- La polarisation de la lumière transmis par le filtre polarisant est donc égale à l'angle d'orientation du filtre. Ainsi, les photons initialement polarisés selon un angle Ω ont maintenant tous une polarisation correspondante à l'angle du filtre, soit un angle ß, ce qui implique que leur polarisation initiale est complètement perdue.



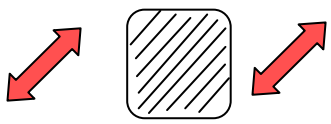


Figure III-4 : Polarisation des photons

III-2-Principe de cryptographie quantique :

La cryptographie quantique utilise un fonctionnement différent de la cryptographie que l'on utilise au quotidien. Elle se base sur les lois de la physique quantique, permet de sécurisé la transmission de données en utilisant des clés générées et échangées à l'aide de particule quantique, Photons. Comme la mécanique quantique stipule que toute observation de l'état quantique d'une particule modifie cet état (principe d'indétermination), toute tentative d'interception de la clé par un espion peut en principe être repérée par les utilisateurs. S'il n'y a pas eu espionnage, une clé parfaitement secrète, et celle-ci peut être utilisée dans tout algorithme de chiffrement symétrique. La sécurité des protocoles de cryptographie quantique est donc absolue et garantie par le fonctionnement des lois de la physique.

Dans les systèmes de télécommunication quantique, les transmissions se font généralement par l'intermédiaire de deux canaux d'échanges différents :

- Un canal quantique (fibre optique) pour la transmission de la clé.
- Un canal de communication classique (canal radio).

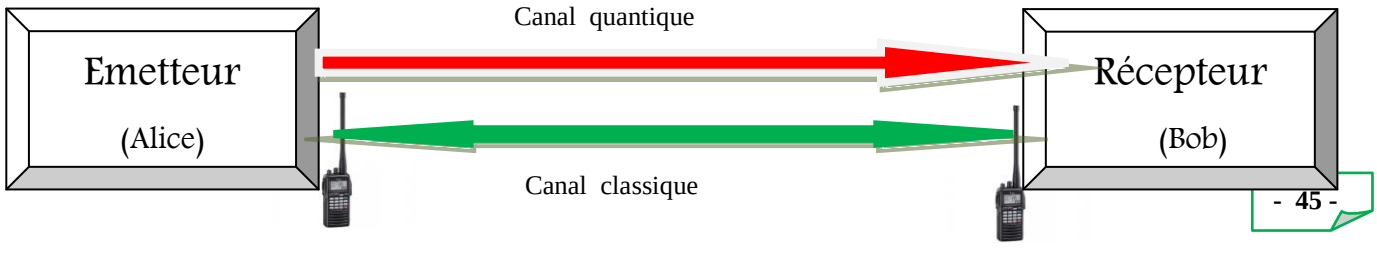


Figure III-5 : schéma du principe du protocole quantique.

Imaginons qu'un espion parvienne à placer des détecteurs sur le canal de communication quantique d'Alice et Bob: qu'est-ce qui l'empêcherait d'intercepter les photons, de les analyser et d'en renvoyer une copie conforme, ni vu ni connu?

Ce sont les lois de la physique quantique qui lui interdisent de procéder à une copie. En effet, si l'on cherche à mesurer, par exemple, la vitesse d'une particule quantique avec la plus grande précision possible, on ne peut pas déterminer sa position avec la même précision; inversement, si vous voulez savoir où est le photon, il faudra renoncer à connaître sa vitesse. C'est le principe d'incertitude d'Heisenberg, du nom du physicien qui l'a décrit en 1927. C'est donc la nature elle-même qui interdit de connaître, à chaque instant, la description complète de l'état quantique d'une particule. Donc jamais un espion ne pourra copier des photons afin d'obtenir discrètement un double de la clef de chiffrement qu'ils transportent. Il faut bien comprendre que la clef n'est pas définie au départ; elle va être constituée à l'issue de la communication.

III-2-1- Création de la clé quantique :

Dans la création de la clé quantique, l'information est transmise par des photons. Alice émet, sur le canal quantique, une suite de photons polarisés au hasard parmi 0°, 45°, 90° et 135°. A l'autre bout, Bob reçoit les photons et mesure aléatoirement ou leur polarisation rectiligne (filtre placé à 0°), ou leur polarisation diagonale (filtre placé à 45°). Si le photon traverse le filtre, Bob note 0, sinon il note 1.

Bien sûr, certaines mesures de Bob (en moyenne, une sur deux) n'ont pas d'intérêt : il a pu essayer de mesurer la polarisation rectiligne d'un photon polarisé à 45°, ce qui n'a pas de sens et donne un résultat aléatoire (par exemple, le photon a été bloqué par le filtre, Bob note donc 1 alors qu'Alice avait envoyé 0). Pour éliminer ces bits sans sens, il indique à Alice, par le canal radio, quel type de mesure (Rectiligne ou diagonale) il a fait pour chaque photon. Par le même canal radio, Alice lui indique quelles sont les mesures correctes (photon polarisé à 0° ou 90° avec filtre rectiligne, photon à 45° ou 135° avec filtre diagonal), dans l'exemple ci-dessous. Les bits 1,3,4,7 sont connus à la fois de Bob et d'Alice, et constituent leur clé secrète commune (0110).

Exemple : Génération d'une clé

Alice émet des photons Valeur en bit :	 0	 0	 1	 1	 1	 0	 0	 1
Bob reçoit les photons à travers un filtre Le photon passe? Valeur en bit :	 OUI 0	 NON 1	 NON 1	 NON 1	 NON 1	 OUI 0	 OUI 0	 OUI 0
---Canal radio--- Bob : ma mesure Alice : correct	diag oui	diag non	rect oui	rect oui	rect non	rect non	rect oui	diag non
Clé reconstituée	0		1	1			0	

III-2-3- Détection d'un espion et preuve de sécurité :

La cryptographie quantique s'inspire des protocoles classiques, à la différence que le support de l'information est maintenant une particule se comportant de manière quantique. Un objet élémentaire, tel qu'un photon ou un électron, est soumis aux principes suivants, posés par la physique quantique :

• Théorème de non-clonage : Il est impossible de réaliser des copies parfaites (des clones) d'états quantiques inconnus.

• Relations d'incertitude de Heisenberg : Si deux observables d'un état quantique ne commutent pas, il est impossible de réaliser simultanément, et avec une précision arbitraire, les mesures correspondantes. Si, maintenant, Eve cherche à mesurer l'information codée sur l'état quantique, ces deux principes nous assurent qu'elle va (1) faire une erreur sur sa mesure, et (2) perturber l'état qui transite entre Alice et Bob. Elle va donc introduire du bruit sur la mesure de Bob, et ce dernier va donc remarquer la présence de l'espion. Ainsi, dans un système de cryptographie quantique, tout espionnage se traduira par du bruit ajouté sur la mesure de Bob, ce bruit devenant donc la signature de l'espion. Preuves de sécurité. Le raisonnement qualitatif ci-dessus ne suffit bien entendu pas à démontrer la sécurité d'un protocole, et il est nécessaire de formaliser les notions d'information, de mesure, etc., dans un calcul dit « preuve de sécurité inconditionnelle ». Ces preuves utilisent des outils de théorie de l'information et de physique quantique pour déterminer la quantité d'information accessible à l'espion, et donc la quantité d'information secrète ayant été transmise par le canal quantique.

III-3-**Protocoles quantique de génération de clés :**

En cryptographie quantique, il existe différents protocoles permet de sécuriser la transmission de données en utilisant des clefs générées et échangées. Chaque protocole contient généralement trois phases :

1- Transmission des qubits et réconciliation des bases.
2- Réconciliation des clés.
3- Distillation de secret.

Sur cette base plusieurs protocoles ont été développés, ce sont :

III-3-1- **Protocole BB84 :**

Le protocole de cryptographie quantique le plus couramment utilisé est désigné par l'acronyme BB84, et a été proposé par C. H. Bennett et G. Brassard en 1984. Ce protocole est assez simple à comprendre. La grandeur physique mesurée est la polarisation d'un photon unique, qui correspond à la direction de vibrations de l'onde lumineuse. Le protocole BB84 n'utilise que des polarisations linéaires, à quatre états non-déterministe, ce qui signifie qu'il distribue une suite aléatoire de bits. Il ne peut être utilisé pour la transmission d'un message déterminé, et les systèmes opérationnels à l'heure actuelle sont tous fondés sur des variantes de ce protocole.
Autrement dit, il est à la base de tous les protocoles qui ont été développés par la suite.

III-3-2- **Protocole B92 (à deux états) :**

En 1992, Charles Bennett propose un protocole pour QKD basé sur deux états non orthogonaux et connus sous le nom de B92. Le protocole B92 est similaire au protocole BB84, mais il utilise uniquement Deux états au lieu de quatre états. Le protocole B92 se révèle être sécurisé sans conditions. Une preuve remarquable de la sécurité inconditionnelle de B92 est la Preuve de Tamaki (Tamaki, 2003). Cela signifie que cette preuve Garantit la sécurité de B92. Toutefois en pratique, ce protocole n'est pas vraiment efficace. En effet, bien que deux états non-orthogonaux ne puissent pas être distingués clairement, on peut les distinguer clairement au coût d'une certaine perte.

III-3-3- Protocole à trois états :

Ce protocole est l'amélioration de BB84. Le protocole BB84 est symétrique dans son utilisation de polarisation. Après la génération de la clef, il est nécessaire d'échanger d'autres informations pour le secret de la clef. Le protocole à trois-état a proposé d'employer trois états, au lieu de quatre dans BB84, et trois détecteurs, au lieu de deux pour BB84, pour casser la symétrie de BB84. Ceci réduit la probabilité

d'espionnage pour obtenir de bons états, et ainsi que minimise la quantité de l'information utile envoyée par Alice.

III-3-4- Protocole à six états :

Tandis que deux états sont suffisants et quatre états sont standard, un protocole à six-état respecte plus la symétrie de l'espace d'état de qubit, (Figure III-6). Les six états constituent trois bases, par conséquent la probabilité qu'Alice et Bob choisissent la même base est seulement 1/3, mais la symétrie de ce protocole simplifie considérablement l'analyse de sécurité et réduit le gain optimal de l'information de l'espion pour un taux donné d'erreur. Si l'espion mesure tous les photons, induira un taux d'erreur de 33%, en comparaison à 25% dans le cas du protocole BB84.

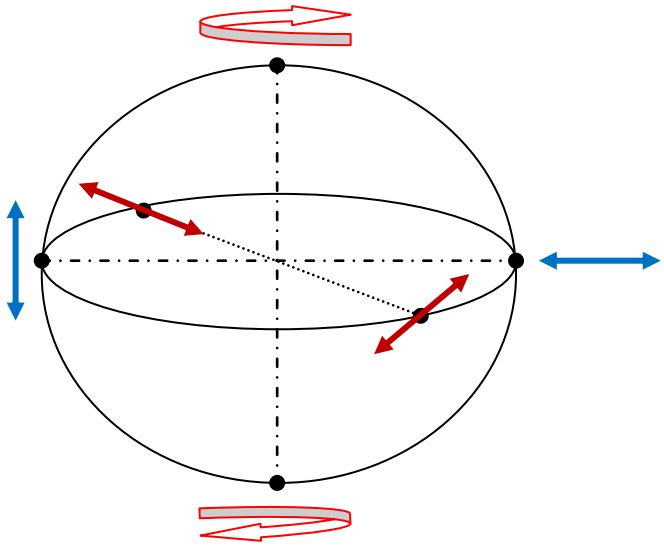


Figure III-6 : Trois paires de bases utilisées dans le protocole à six-état.

III-4- Description du Protocole BB84 :

BB84 est le protocole de distribution de clef de quantum le plus bien connu en utilisant quatre états différents qui font une paire des états de base. Le bit classique est codé par des états de quantum. Chaque état de quantum peut représenter les deux bits classiques, le 1 ou le 0, et inversement, chaque 0 ou 1 correspond à un mélange de deux états égaux de quantum probablement non-orthogonaux. Ce qui ne donne quatre états $|0>$, $|1>$, $|0'>$, $|1'>$.

L'information transmise dans le canal de quantum est souvent sous la forme de photons polarisés. Le codage des bits classiques est fait en utilisant la direction de polarisation. Le schéma de codage de BB84 et donné par la figure III-7, le bit classique 0 est représenté par un photon polarisé à 0° et 45° de l'axe horizontal, et les deux directions orthogonales correspondantes, 90° et 135°, sont employées pour le bit 1. Dans la description du protocole, on emploi le prénom classique pour les différents éléments du protocole. Le prénom **Alice** est employé pour l'initiateur du protocole. Le prénom **Bob** est employé pour

le destinataire. Typiquement, Alice communique avec Bob tandis qu'un espion essaye d'écouter ou de perturber la communication. Cet Espion est habituellement appelé **Eve** (Eavesdropper en anglais).

BB84 est un protocole non-déterministe. Cela signifie qu'il distribue une suite aléatoire des bits. BB84 ne peut pas être employé pour la transmission d'un message déterminé. La communication entre Alice et Bob étant simplement réussie, se base sur l'aspect aléatoire de chaque étape du protocole.

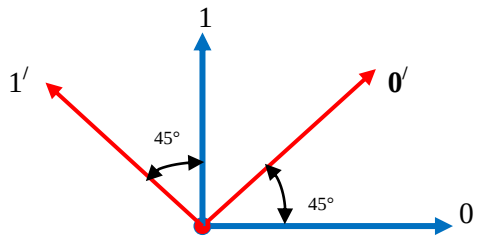


Figure III-7: quatre états Non-orthogonaux utilisés dans le protocole BB84.

Selon la mécanique quantique, il n'y a aucune manière de différencier sûrement deux états non-orthogonaux. Ainsi une mesure de quantum doit être effectuée pour déterminer l'état reçu et grâce à cela pour obtenir le rendement classique. Et ce principe d'incertitude fournit les propriétés cryptographiques requises dans la cryptographie de quantum. Pour BB84, il y a deux mesures employées pour distinguer les différents états de quantum:

1- la mesure permettant d'identifier clairement entre deux états $|0\rangle$ et $|1\rangle$, Cette mesure s'appelle également la mesure dans la base rectiligne $\oplus$.

2- la mesure permettant d'identifier clairement entre deux états $|0'\rangle$ et $|1'\rangle$. Cette mesure s'appelle également la mesure dans la base diagonale $\otimes$.

En général, l'échange de clef de quantum employant le protocole BB84 pour la clef secrète se compose de six étapes suivantes:

1. Transmission de quantum.
2. Annonce de bases.
3. Estimation d'erreurs.
4. Réconciliation.
5. Confirmation.
6. Purification.

1- Transmission de quantum :

Cette phase est la première étape dans une distribution de clef de quantum. Dans cette phase, une chaîne aléatoire de *n* bits classiques sera créée par Alice et envoyée à Bob. Chaque bit de cette chaîne sera codé par une base non déterministe. Un bit classique codé par quantum s'appelle un qubit. A l'autre côté de la transmission, Bob reçoit le qubit et il prend par hasard une base rectiligne (ou diagonale) pour la mesurer. Quand la transmission est finie, Bob obtiendra une chaîne de bits classiques, appelée clef crue, différente de celle d'Alice en beaucoup de positions, environ 50% même dans le cas de la communication sans erreur de quantum ou beaucoup de plus comme le taux d'erreur d'appareil est inclus. La prochaine étape du protocole aidera à remplacer les bits non-corrélatifs entre la chaîne d'Alice et celle de Bob, qui sont probablement des erreurs provoquées par l'espion ou la transmission bruyante de quantum.

2- Annonce de bases :

Comme mentionné dans la partie précédente, dans cette phase, toutes les positions, où les mêmes bits sont partagés, seront conservées et le reste sera jeté. Premièrement, à l'aide du canal classique, Bob envoie à Alice toutes les bases qu'il avait utilisées pour mesurer la chaîne des qubits d'Alice. Alice alors compare cet ordre des bases avec le sien et révèle toutes les positions non-corrélatives sur le canal classique à Bob. Après ça, Alice et Bob enlèvent tous les bits aux positions informées par Alice. Et la partie distillée de la clef crue, appelée la clé plaine, est totalement la même entre celles d'Alice et de Bob sans tenir compte des effets des erreurs d'appareil mais encore ou même tout à fait différent de l'un, de l'autre en fait.

3- Estimation d'Erreurs :

Pour réduire la différence entre la clef plaine d'Alice et de Bob due à l'imperfection d'appareil, il est nécessaire de corriger des erreurs. C'est la phase de l'estimation d'erreur de la clef plaine. Il sera exécuté comme suit. Alice extraira une petite série des bits de la clef plaine et l'envoi à Bob. Alice informera Bob un sous-ensemble de positions de taille *K* et les valeurs de bits à ces positions dans la clé plaine obtenue dans la dernière étape. L'émetteur et le récepteur doivent calculer le taux d'erreurs observées et gardent cette transmission si le taux d'erreur est moins qu'un seuil désiré. Et en fin, ils enlèvent *K* bits vérifiés et observent le reste. Dans le cas où le taux d'erreur estimé est plus que le seuil, la clef sera rejetée.

4- Réconciliation :

Après cette phase, une clef réconciliée sera obtenue après application d'un protocole de réconciliation à la clef plaine. La réconciliation est un processus interactif, ayant lieu dans le canal public. Le but de cette phase est de corriger les erreurs, pour réduire d'une manière équivalente la différence, entre les clefs plaines de l'expéditeur et du récepteur. Mais il est important de prendre note que peu de bits en tant que possible sont envoyés à travers le canal public car l'espion peut exploiter cette information.
Un exemple d'algorithme simple est le suivant : Alice choisit aléatoirement des paires de bits et divulgue publiquement le numéro de ces bits ainsi que leur somme XOR. Si Bob obtient le même résultat, il gardent le premier bit de la paire et jettent le second. Dans le cas contraire, il jette les deux bits. Cet algorithme permet de réduire le taux de différence entre la clé d'Alice et de Bob.
Un autre type d'algorithme nommé cascade est appliqué ici. La cascade effectue de correction d'erreurs en envoyant très peu de l'information à travers le canal public et a été proposée par Gilles Brassard et Louis Salvail. La cascade fonctionne dans un certain nombre de ronds. La phase suivante est la confirmation de l'égalité des clefs réconciliées de Alice et de Bob. Et si la cascade finit avec succès, la phase suivante confirmera le résultat.

5- **Confirmation :**

Afin de s'assurer qu'aucune erreur ne sera trouvée, Alice et Bob échangeront et compareront la parité des sous-ensembles aléatoires de positions. En général, si une comparaison à z bits de parité est faite et il n'y a aucune différence, alors la clef partagée courante est identique au taux de 2^{-z}. Et si cette phase est réussie, nous croyons probablement que la clef partagée est maintenant la même, peut-être avec l'erreur mais à un taux acceptable si z est assez grand.

6- **Purification :**

Enfin, Alice et Bob peuvent avoir une clef identique, mais que pensez-vous de l'espion? Après toutes les phases précédentes, peut-être elle a obtenu de l'information, donc la sécurité de la clef partagée Identique est menacé. Que doit-on faire pour résoudre ce problème? La purification est exactement la réponse.

Le but de cette phase est réduire au minimum aussi loin que possible les informations de l'espion sur la clef et de produire une clef plus courte mais plus confiante. Pour régler ce problème une technique se nomme distillation de secret (privacy amplification en anglais). Considérons à nouveau le cas le plus simple :

Alice choisit des paires de bits dont elle prend la somme XOR, mais cette fois ci, elle divulgue seulement le numéro des bits. Alice et Bob remplacent simplement la valeur de chacun des bits par leur somme XOR. Ainsi, ils n'engendrent pas de nouvelle différence entre leur clé et réduisent l'information d'Eve. En effet, si Eve ne connait que la valeur de premier bit mais pas celle du deuxième, elle n'a aucune information sur leur somme XOR.

Conclusion :

La cryptographie quantique, résout donc, en se basant sur les règles de la mécanique quantique, l'une des plus importantes problématiques de la cryptographie classique, à savoir la distribution de clé. En effet, en cryptographie quantique, et en utilisant la polarisation du photon pour représenter l'information, il est possible aux traditionnels communicants Alice et Bob, de partager une clé tout à fait sûre. Bien entendu, cela se fait, comme il vient d'être introduit dans le présent chapitre, après la génération aléatoire d'une

chaîne de bits qui devra passer par une série d'étapes de traitement dans le protocole BB84 (correction d'erreur, amplification de la confidentialité). C'est cet aspect aléatoire même, qui procure la sûreté de la clé quantique.

D'autre part, c'est ce même aspect de l'aléatoire qui a fait que la cryptographie quantique ne soit pas de la cryptographie proprement dite mais plutôt de la distribution de clé quantique. En effet, il serait impossible, avec cette politique, d'envoyer un message prédéfini puisqu'il est impossible de définir les valeurs du message à partager au préalable vu le choix aléatoire des bases. Par contre, cela ne nuit pas à la clé qui se doit d'être aléatoire pour des fins de sécurité.

Cela dit, une fois la clé finale obtenue, il est possible de l'utiliser à des fins de cryptage, procurant une sécurité inconditionnelle aux participants d'une communication qui se fera sur un canal classique, assurant ainsi la fiabilité d'un système de cryptage intégré dans l'une des applications nécessitant un degré élevé de sécurité.

Introduction :

QuCrypt est le logiciel développé pour conduire l'expérience cryptographique quantique. Le logiciel, écrit en Java, met en œuvre toutes les étapes nécessaires pour que la distribution des clés de secret quantique se déroule. En outre, il fournit une interface avec le canal quantique réel utilisé dans l'expérience, il permet également d'exécuter la distribution des clés quantiques dans un mode de démonstration sans avoir besoin de matériel spécialisé. Dans ce document, nous décrivons l'architecture du logiciel, les fonctionnalités mises en œuvre et la procédure d'installation.

IV-1-Architecture générale

Le logiciel vise à fournir un environnement aussi proche que possible d'une situation réelle où un protocole quantique à 2 parties pourrait être utilisé. Bien que cette version ne fournisse qu'une distribution de clé secrète, l'architecture permet d'inclure n'importe quel protocole cryptographique à 2 parties dans le futur. Les deux parties impliquées dans un protocole peuvent s'exécuter sur une seule machine ou peuvent être connectées à distance via une connexion socket. En plus des deux parties, des serveurs pour la connexion quantique sont fournis. Ces serveurs peuvent être utilisés soit pour simuler un canal quantique, soit pour l'interface des deux extrémités d'un canal quantique réel, les serveurs peuvent être exécutés sur une seule machine ou à distance. La sortie générée par une exécution de protocole est une collection de fichiers HTML donnant des commentaires pour toutes les phases pertinentes du protocole.
L'architecture logicielle contient 4 entités principales:

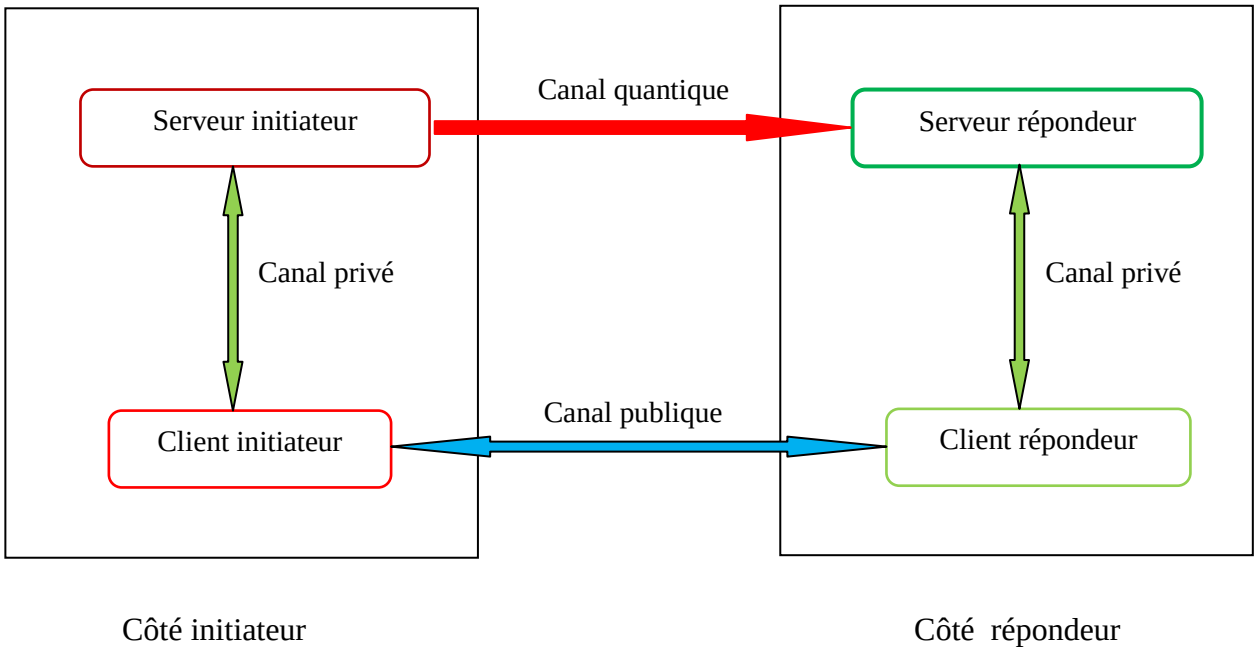


Fig IV-1 : Architecture générale

Serveur Initiateur : Responsable du côté initiateur du canal quantique. Il transmet des qubits à travers le canal quantique et les renvoie au lecteur initiateur. La transmission quantique est réalisée selon le codage quantique que les clients ont décidé d'utiliser pendant le protocole.

Serveur répondeur : Responsable du côté répondeur du canal quantique. Il reçoit les qubits envoyés par le serveur initiateur et renvoie au client répondeur le résultat des mesures qui lui sont appliquées. Les mesures sont sélectionnées en fonction du codage quantique que les clients ont convenu.

Client initiateur: Il choisi les paramètres à utiliser, lance l'exécution du protocole quantique. Il établit également une connexion avec le serveur initiateur à partir duquel l'information quantique est obtenue, généralement supposé fonctionne avec le serveur initiateur sur la même machine.

Client répondeur : Le client répond-il à la demande et donc connecté au client initiateur et au serveur répondeurs. Il obtient le résultat de la transmission quantique à partir du serveur répondeur, généralement supposé fonctionner avec le serveur répondeur sur la même machine.

IV-2-Serveur Initiateur

Le serveur initiateur est un processus chargé de fournir au client initiateur une connexion au canal quantique. Le canal quantique peut être réel ou virtuel. Dans le cas d'une transmission virtuelle, le serveur initiateur simule un canal quantique en communiquant de manière classique avec le serveur répondeur (via une connexion IP). Lorsque la communication quantique est réelle, le serveur initiateur se connecte également au canal quantique. Il s'agit d'une connexion IP normale à l'appelant de saisie de données quantique. En règle générale, le serveur initiateur s'exécute sur la même machine que le client initiateur. Lors de l'exécution, le serveur initiateur se présente comme suit:

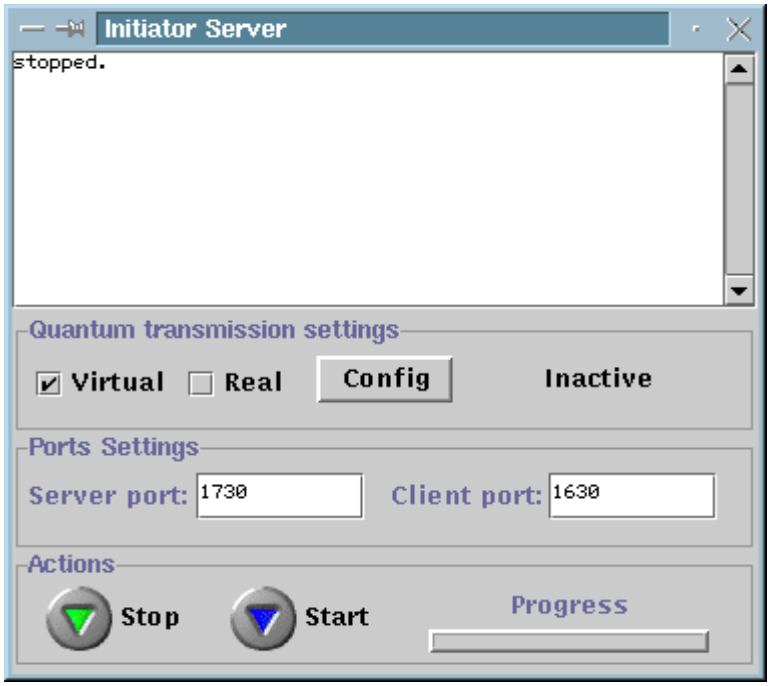


Fig IV-2 : Serveur initiateur

La fenêtre supérieure affiche des informations à l'utilisateur pendant que le serveur fonctionne. En outre, la fenêtre du serveur initiateur contient les éléments suivants rassemblés dans 3 groupes:

Paramètres de transmission quantique:

Les cases à cocher Virtual et Real : indiquent transmission quantique suivante se fait à travers un canal quantique simulé ou à travers l'expérience réelle.

Le bouton Config : permet la configuration des modes de transmission quantique virtuelle et réelle. Il agit différemment compte tenu du mode sélectionné:

Configuration virtuelle : permet de définir le taux d'erreur pour la prochaine transmission virtuelle. Un qubit virtuel transmis sera retourné avec la probabilité sélectionnée avant d'atteindre le côté du répondeur.

Configuration réelle: permet de sélectionner l'adresse IP et le numéro de port pour la connexion au canal quantique à travers une entité appeler l'enregistreur de données quantiques. Il permet également de sélectionner la taille du seau qui est le nombre d'impulsions collectées par le canal quantique avant de renvoyer leurs valeurs au serveur initiateur. Comme la plupart des impulsions sont vides, la sélection de grandes tailles de seaux augmente la vitesse de transmission. Lorsqu'un seau a été obtenu, les impulsions non vides dans ce seau sont renvoyées à l'initiateur tout à la fois.

Paramètres des ports:

Le port de serveur : est le numéro de port auquel le serveur initiateur attend une connexion avec le serveur répondeur.

Le port client : est le numéro de port auquel le serveur initiateur attend une connexion avec le client initiateur.

Actions:

Le bouton démarrer / arrêter : sert à démarrer ou à arrêter le serveur. Pour l'instant, pour chaque transmission quantique, le serveur initiateur doit être redémarré. Après chaque transmission, le serveur s'arrête. Si le bouton d'arrêt est enfoncé pendant que le serveur est en cours d'exécution, l'exécution est interrompue.

La barre de progression : indique la progression pendant la transmission quantique.

IV-3-Serveur répondeur :

Le serveur répondeur est un processus responsable de fournir au client répondeur une connexion au canal quantique. Le canal quantique peut être réel ou virtuel. Dans le cas d'une transmission virtuelle, le serveur répondeur simule une chaîne quantique en communiquant de manière classique avec le serveur initiateur (via une connexion IP). Lorsque la communication quantique est réelle, le serveur répondeur se connecte également au canal quantique. C'est grâce à une connexion IP normale avec le capteur de données

quantique. En règle générale, le serveur répondeur fonctionne sur la même machine que le client répondeur.

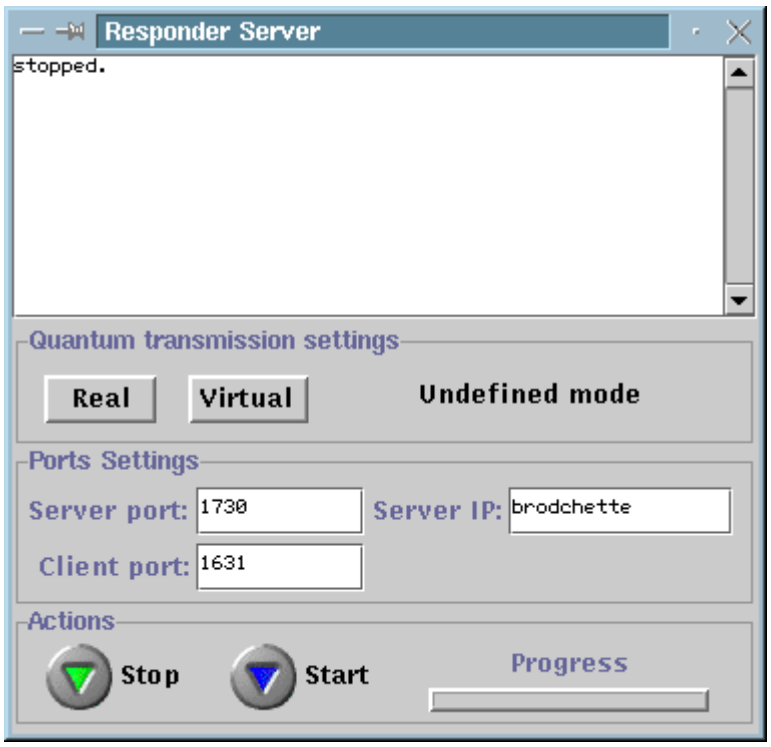


Fig IV-2 : Serveur répondeur

La fenêtre supérieure affiche des informations à l'utilisateur pendant que le serveur fonctionne. De plus, la fenêtre du serveur répondant contient les éléments suivants réunis en 3 groupes:

Paramètres de transmission quantique:

Bouton virtuel: permet de définir le taux d'erreur pour la prochaine transmission virtuelle du côté correspondant. Un qubit virtuel reçu sera retourné avec la probabilité sélectionnée lorsqu'il sera détecté.
Bouton réel: permet de sélectionner l'adresse IP et le numéro de port pour la connexion au canal quantique ou plus précisément au capteur de données quantique.
Paramètres des ports:
L'adresse IP du serveur est l'adresse IP du serveur initiateur. Lorsque le serveur de répondeurs est lancé, il essaie de se connecter au serveur initiateur.
Le port du serveur est le numéro de port auquel le serveur de répondeur se connecte au serveur initiateur.
Le port client est le numéro de port auquel le serveur répondeur attend une connexion avec le lecteur répondeur.

Actions:

Le bouton démarrer / arrêter : sert à démarrer ou à arrêter le serveur. Pour l'instant, pour chaque transmission quantique, le serveur répondeur doit être redémarré et une nouvelle connexion avec le serveur initiateur doit être établie. Après chaque transmission, le serveur s'arrête. Si le bouton d'arrêt est enfoncé pendant que le serveur est en cours d'exécution, l'exécution est interrompue.
La barre de progression : indique la progression pendant la transmission quantique.

IV-4-Client initiateur :

Le client initiateur est la partie qui lance l'exécution d'un protocole quantique. Il se connecte au serveur initiateur et s'attend à une connexion du client répondeur. Une fois que ces connexions sont établies, le client initiateur envoie la description du protocole avec les paramètres sélectionnés au client répondeur. Si les paramètres sont acceptés par le client répondeur, l'exécution commence. Par conséquent, le client initiateur est responsable de sélectionner à la fois le protocole et la configuration avant qu'une connexion avec le client répondeur ne soit établie. Lorsqu'il est lancé, le client initiateur ressemble à ceci:

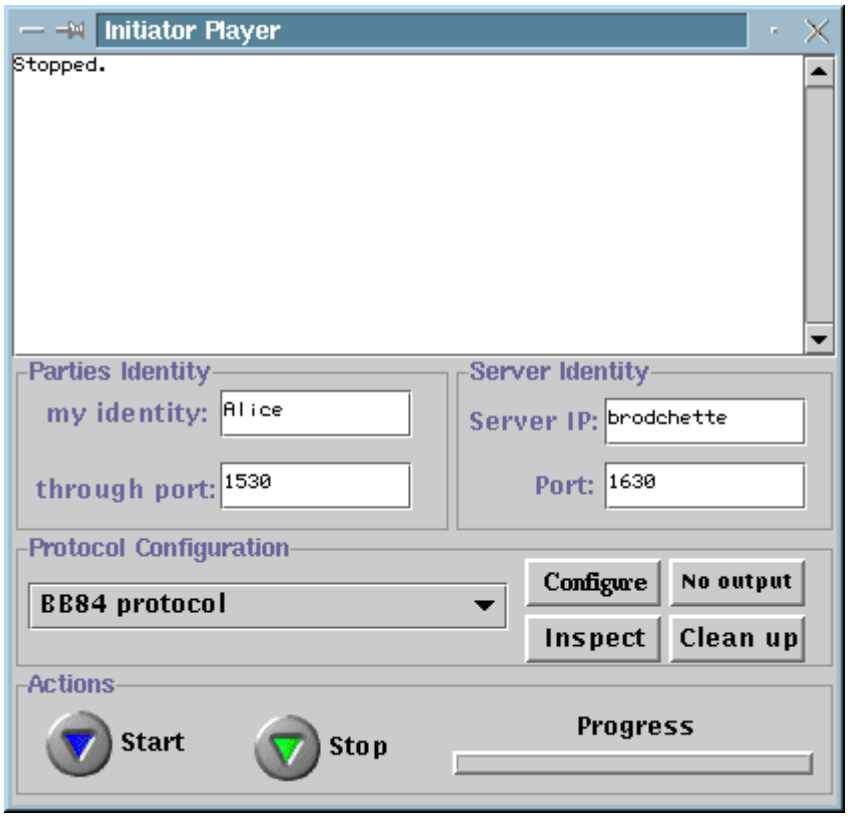


Fig IV-3 : Client initiateur.

La fenêtre du lecteur initiateur contient 4 groupes d'éléments comme suit:

Identité des parties:

Mon identité est une chaîne pour l'identification du client initiateur. Le client initiateur est généralement appelé Alice dans la littérature cryptographique.
Le port **par port** est le numéro de port auquel le client répondeur doit se connecter avec le client initiateur.

Identité du serveur:

L'adresse IP serveur : l'adresse IP du serveur initiateur.
Le port : est le numéro de port auquel le client initiateur se connecte au serveur initiateur.

Configuration du protocole:

Le bouton de configuration du protocole : permet de configurer le protocole sélectionné pour la prochaine exécution. Les valeurs seront envoyées au client répondeur lors de la connexion. La configuration doit avoir lieu avant de commencer le lecteur afin que les changements se produisent.
Le bouton Inspecter : permet de voir les valeurs définies pour tous les paramètres du protocole sélectionné. Cela ne permet pas de les modifier.

Le bouton de sélection de sortie : permet de définir ou de désactiver la génération de sortie HTML. Lorsque le bouton indique Non Sortie, aucune sortie ne sera générée après l'exécution. Lorsque le bouton indique la sortie HTML, la sortie de l'exécution suivante sera disponible à partir du fichier index du répertoire de modèles précédemment sélectionné. Le répertoire de modèle valide fourni avec le logiciel est situé dans qrypto / templates / initiateur pour l'expéditeur du client initiateur.

Le bouton Nettoyer : permet de supprimer certaines exécutions du fichier index du répertoire de modèles sélectionné pour la génération de sortie HTML.

Actions:

Le bouton démarrer / arrêter déclenche et arrête l'action. Le démarrage doit être pressé après que le protocole a été sélectionné et paramétré. Il fait ce qui suit:

1- Connectez-vous au serveur initiateur.
2- Attendez que le lecteur répondeur se connecte.

3- Lors de la connexion, le protocole sélectionné est envoyé au client répondeur.
4- Si le client répondeur a accepté le protocole, le client initiateur indique au serveur initiateur d'initier une transmission quantique du type choisi.
5- Le protocole est exécuté.
6- Après l'exécution, le lecteur s'arrête automatiquement.

La barre de progression : Indique la progression et la phase de protocole actuellement exécutée.

IV-5-Client répondeur :

Le client répondeur est la partie qui répond à une demande d'exécution d'un protocole quantique. Il se connecte à la fois au serveur répondeur et au client initiateur. À partir de ce dernier, il reçoit la description du protocole à exécuter avec la configuration. Si le client répondeur accepte la demande, le protocole commence à s'exécuter. Lorsqu'il est lancé, le client répondeur ressemble à ceci:

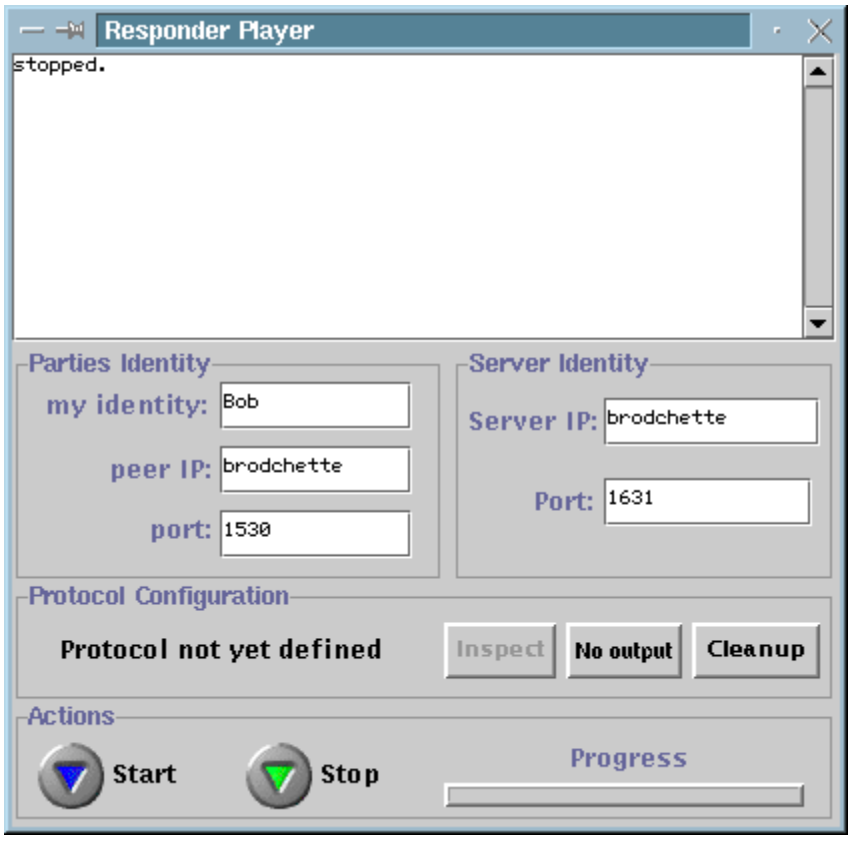


Fig IV-2 : Client répondeur

La fenêtre du lecteur initiateur contient 4 groupes d'éléments définis comme suit:

Identité des parties:

Mon identité est une chaîne pour l'identification du client répondeur, généralement appelé Bob dans la littérature cryptographique.

L'adresse IP du serveur : Est l'adresse IP du serveur répondeur.

Port : Est le numéro de port auquel le client répondeur se connecte au client initiateur.

Identité du serveur:

L'adresse IP du serveur : Est l'adresse IP du serveur initiateur.

Le port : Est le numéro de port auquel le client initiateur se connecte au serveur initiateur.

Configuration du protocole:

L'étiquette du protocole indique le protocole sélectionné par le client initiateur et convenu. Le protocole sélectionné est l'un parmi une liste de protocoles pris en charge. Cette étiquette indique toujours un protocole qui n'est pas encore défini lorsqu'il est éteint.

Le bouton Inspecter : permet de voir les valeurs définies pour tous les paramètres du protocole convenu avec le client initiateur. Cela ne permet pas de modifier les paramètres.

Le bouton de sélection de sortie : Permet de définir ou de désactiver la génération de sortie HTML. Lorsque le bouton indique Non Sortie, aucune sortie ne sera générée après l'exécution. Lorsque le bouton indique la sortie HTML, la sortie de l'exécution suivante sera disponible à partir du fichier index du répertoire de modèles précédemment sélectionné. Le répertoire de modèle valide fourni avec le logiciel se trouve dans qrypto / templates / répondeur pour l'expéditeur du client initiateur.

Le bouton Nettoyer : Permet de supprimer certaines exécutions du fichier index du répertoire de modèles sélectionné pour la génération de sortie HTML.

Actions:

Le bouton démarrer / arrêter déclenche et arrête l'action. Le démarrage doit être pressé après la sélection et la configuration du protocole. Il fait ce qui suit:

1- Connectez-vous au client initiateur.
2- Attendez la description du protocole et la configuration, elle accepte ou rejette la configuration reçue.
3- Lors de l'acceptation, il se connecte au serveur répondeur et transmet l'information quantique demandée selon la sélection du protocole.

4- Le protocole est exécuté.
5- Après l'exécution, le lecteur s'arrête automatiquement.

La barre de progression : Indique la progression et la phase de protocole actuellement exécutée.

Les problèmes de sécurité:

- La connexion entre le serveur initiateur et le client initiateur doit être privée afin que le protocole quantique soit mis en œuvre de manière sécurisée. Cela peut être réalisé en exécutant à la fois le serveur initiateur et le client initiateur sur la même machine.
- La connexion entre le serveur répondeur et le client répondeur doit être privée afin que le protocole quantique soit mis en œuvre de manière sécurisée. Cela peut être réalisé en exécutant à la fois le serveur répondeur et le client répondeur sur la même machine.
- La connexion entre le serveur initiateur et le serveur répondeur n'a pas besoin d'être privée mais doit être authentifiée.

Conclusion :

Le présent chapitre était dédié à l'étude d'un cas de l'application QuCrypt de la cryptographie quantique, ou plus précisément, du protocole de distribution de clé quantique BB84 étudié au niveau du chapitre trois, et qui, en bénéficiant des lois de la mécanique quantique, offre une haute sécurité de la clé partagée.

Renforcer la sécurité d'un réseau, nécessite une implémentation physique, Ceci dit, une implémentation physique est trop couteuse et difficile à mettre en œuvre, chose qui pousse alors à la simulation comme solution inévitable. Nous avons alors présenté, en bref la base de ce qui pourrait faire l'objet d'une plateforme de simulation du protocole standard de base, à savoir le BB84, initialisé par les différents paramètres qui le régissent, la présentation de la simulation est alors accompagnée de certaines copies d'écran de quelque test d'exécution.

Conclusion générale et perspectives

Suite à cette étude nous pouvons affirmer que la cryptographie est la sciences du secret par excellence tel qu'on la définit dans la littérature du domaine, et qu'elle a permis de sécuriser, tout au long du siècle passé bien des systèmes d'information et de transmission malgré les différentes attaques auxquelles elle était sujette. Ceci dit, la technologie avance à pas de géant, et la force de calcul des machines va en augmentant, surtout avec l'apparition de la machine quantique, qui ne fait plus partie de l'imaginaire ou de la science fiction mais qui a mis un pas en dehors des laboratoires. Si une telle machine est disponible à de moindres coûts les systèmes de cryptage actuels seraient tous mis en péril.

En effet, ce type de machine pourrait réaliser des opérations jusque là considérées comme impossibles, comme la factorisation des grands nombres, opération sur laquelle repose toute la sécurité des systèmes de cryptages asymétriques actuels.

Dans le cadre de ce travail, il était question d'étudier, dans un premier temps, les techniques de cryptographie classique, à la fois symétrique et asymétrique, algorithmes de chiffrement, puis dans un second temps, d'étudier la cryptographie quantique ; nouvelle technique de cryptage basée non plus sur la complexité des calculs et les concepts mathématiques, mais plutôt sur les lois de la mécanique quantique. Il s'avère suite à cette étude que la cryptographie quantique n'est pas un algorithme de cryptage proprement dit, mais plutôt une technique sûre de *distribution* de clé. La distribution de clé étant une problématique d'envergure et dont souffrent particulièrement les systèmes de cryptages symétriques, si l'on suppose que les systèmes asymétriques n'en sont pas sujets et que l'on ne considère pas la fragilité de ces systèmes face à la puissance de calcul quantique.

Le protocole BB84 nous permet de créer une clé inconditionnellement sûre, puisque sa sûreté ne dépend plus de la complexité des calculs mathématiques, comme nous venons de l'introduire, mais plutôt des lois approuvées de la physique et de la mécanique quantique.

Notre étude a été orienté vers une approche analytique, mais elle pourrait être validée par une implémentation physique réelle, en vue d'obtenir des résultats plus concrets, or ceci semble difficile, à la fois sur le plan physique et financier dans le contexte actuel, ce qui nous mène à réfléchir à une solution de second ordre, la simulation.

Nous avons alors pris l'initiative de présenter une application QuCrypt possédant une plateforme qui servira ultérieurement, comme composante essentielle pour une architecture de simulation qui regroupe à cet effet plus d'un paramètre pour la simulation du protocole de base que nous avons étudié dans le cadre de cette recherche, en l'occurrence le BB84.

Ceci étant, il existe d'autres protocoles qui peuvent aussi, faire l'objet d'éventuelles futures recherches, tel que le B92, basé seulement sur deux états quantiques, à la différence du BB84 qui, lui, se base sur quatre états, ainsi que les protocoles basés sur les paires EPR qui constituent aussi une révolution en matière de cryptographie quantique. En somme, il n'est pas exagéré de dire que le domaine de la cryptographie quantique est toujours un domaine d'actualité déjà par les principes sur lesquels il se base, mais aussi parce qu'il regorge de pistes novatrices à méditer mais surtout à explorer.

Bibliographie :

1- [Cla-15] : Claude Aslangul-Mécanique quantique 2-Développement et application à basse énergie-3ème Edition –Master de physique (M1 et M2) –Ed de boeck.

2- [Dam-12] : Damien Vergnaud –préface de jaques stern-Exercice et problème de cryptographie - Ed Dunod.

3- [Dje-09] : Djellab Rima - Cryptographie Approche Quantique - Master en informatique industriel - Université Batna - 2009.

4- [Elb -10] - International Journal of Universal Computer Sciences (Vol.1-2010/Iss.2) Mohamed Elboukhari*, Mostafa Azizi**, Abdelmalek Azizi*,/ Quantum Key Distribution Protocols : A Survey / pp. 59-67.

5- [Ngu 04] Rapport de stage -Etudier et implémenter une simulation du protocole d'échange de clef quantique BB84 - réalisé par Nguyen Thanh - Mai IFI-P8-Paris, Mai 2004 - Janvier 2005.

6- [Pie-12]- Barthelemy P., Rolland R., Véron Pascal- Cryptographie : principes et mises en œuvre – 2ème édition revue et augmentée- Lavoisier 2012.

7- [Sti-10] : Stiti karima & Bellouni Souad –Sécurité des données par la cryptographie quantique- Mémoire de fin d'étude Master Réseau et telecom –Département Electronique- UMMTO-2010.

8- [Sou-08] : Souici Ismahane - Cryptographie Nouvel Algorithme de Chiffrement Evolutionnaire basé Occurrences (ACEO)- Magister en Informatique - Université de Guelma - 2008.

Sites internet :

- [Web 01] - http://nopb.chez.com/crypto2.html. - 28/05/2017
- [Web 02] - http://www.apprendre-en-ligne.net/crypto/ - 29/05/2017
- [Web 03] - https://fr.wikipedia.org/wiki/Carré_de_Polybe - 01/06/2017.
- [Web 04] - https://fr.wikipedia.org/wiki/RC2.RC5.RC6. - 05/06/2017
- [Web 05] - (http://www.bibmath.net/crypto/index.php?action=affiche&quoi=moderne/)-07/06/2017
- [Web 06] - (http://www.nymphomath.ch/crypto/quantique/index.html)-07/06/2017