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

UNIVERSITE M'HAMED BOUGARA – BOUMERDES

Faculté de Technologie Département

Ingénierie des Systèmes Electriques **Mémoire de Master**

Présenté par

HEMAMDIA ALI et FOUDAD ABDERRAHMANE

Filière : Télécommunication

Spécialité : Réseaux et Télécommunication

ECHANGE DE CLE QUANTIQUE ET CHIFFREMENT SYMETRIQUE

Soutenu le 13 / 0 9 /2020 devant le jury composé de:

-	MESSAOUDI	Noureddine	MCA	UMBB	Président
-	MEHIBEL	Nissa	MAB	UMBB	Examinatrice
-	HAMADOUCHE	M'hamed	Prof	UMBB	Encadreur

Année Universitaire : 2019/2020

TABLE DES MATIÈRES

LISTE DES TABLEAUX

TABLE DES FIGURES

DÉDICACE

A mes chèrs parents pour leur patience, leur amour, leur soutien et leurs encouragements

A mes chers enfants

"MOHAMED ANAS - SERINE DJENANE- MAJED ESLAME "

A mes chers frères et ma grande familles

A mes amis, mes proches et mes camarades

"AFANE - KHELIFA -KAMAL - SEFIANE - AMIN - RABAH -LAKHDAR "

A mon binôme« ABDRRAHMAN » et à toute sa famille.

A tous mes collègues particulièrement RAOUF, AMIN Massy, Zakaria,

A toutes mes proches Et tous mes amis (es).

A toute personnes qui m'ont encouragé ou aidé tout au long de

mes études.

HEMAMDIA ALI

DÉDICACE

C'est avec un grand plaisir que je dédie ce modeste travail, fruit de mes études, en exprimant ma profonde gratitude à tous mes proches

particulièrement à :

Mes précieux parents pour leur amour, affection et compréhension, leurs

patiences et leurs sacrifices.

Mon père pour, ses encouragements et son soutien. Ma mère pour l'éducation qu'elle m'a inculquée et toutes les peines

qu'elle s'est donnée pour ma réussite.

Mes très chers sœurs, je leurs souhaite tout le bonheur et la réussite qu'ils

espèrent.

A mes tantes et mes oncles, ma grande mère, mes cousins et mes

cousines.

A mon binôme« Ali » et à toute sa famille.

A tous mes collègues particulièrement Merzak, Walid , Lamine, Massy, Fares,Omar, Yanis, Anis, Sidali,Oussama, Zakaria, Boudjou.

A toutes mes proches Et tous mes amis (es).

A toute personnes qui m'ont encouragé ou aidé tout au long de

mes études.

foudad abdrrahman

REMERCIEMENT

Au terme de ce travail, nous tenons à exprimer notre gratitude et nos remerciements pour toutes les personnes qui ont contribué à sa réalisation.

Nous tenons tout d'abord a‘ remercier notre promoteur Mr HAMADOUCHE M'hamed pour son aide, son encouragement, et sa disponibilité´ dans ce projet.ainsi que Mr. Oussama. M et Mr : RAMDANI Abderahmen pour leurs aides et leurs conseils dans la réalisation de dans ce projet en particulier dans la partie pratique.

Nous exprimons également notre gratitude à tous les professeurs et enseignants qui ont collaboré a‘ notre formation depuis notre premier cycle d'étude jusqu'à la fin de notre cycle universitaire.

Nos vifs remerciements vont au président et aux membres de jury, devant qui nous avons l'honneur d'exposer notre travail, et qui ont eu la peine de lire avec soin ce mémoire pour juger son contenu. Sans omettre bien sur de remercier profonde´ment tous ceux qui ont contribué de prés ou de loin à la réalisation du présent travail.

LISTE DES ABRÉVIATIONS

— ***AES*** ▷ ***Advenced Encryption Standard***
— ***Alice*** ▷ ***Pour disigner l'emetteur du message***
— ***BB84*** ▷ ***Protocole de Bennett et Brassard présente en 1984,***
— ***Bob*** ▷ ***Pour disigner le récepteur du message***
— ***B92*** ▷ ***Protocole de Bennett, présenté en 1992***
— ***DES*** ▷ ***Data Encryption Standard***
— ***Eve/Oscar*** ▷ ***Pour disigner l'intrus qui va essayer d'intercepter le message***
— ***IR*** ▷ ***L'attaque Intersept-Resend***
— ***OTP*** ▷ ***One Time Pad***
— ***QBER*** ▷ ***Quantum Bit Error Rate***
— ***QKD*** ▷ ***Distribution de Clé Quantique***
— ***RSA*** ▷ ***Rivest, Shamir, Adleman***
— ***XOR*** ▷ ***Ou Exclusive***
— ***A*** ▷ ***Anti diagonale***
— ***H*** ▷ ***Horizontale***
— ***V*** ▷ ***Verticale***
— ***D*** ▷ ***Diagonale***

INTRODUCTION

A l'heure de la quatrième révolution industrielle, où les données sont collectées ,transférées et stockées dans des réseaux a l'échelle globale, **la cybersecurite et la cryptographie deviennent des sujets de grande importance**.

Le développement des technologies telles que l'Internet des Objets, l'Intelligence Artificielle augmente le trafic des données dans des réseaux, tandis que le fonctionnement quotidien des industries, des administrations et des individus, impliquant des transactions des données personnelles de santé ou financières, et des secrets commerciaux ou nationaux, est de plus en plus confrontée à la transmission des données sensibles ou même critiques, qui nécessitent de la protection, notamment face à des menaces contre leur confidentialité a long terme. Il existe en effet des données qui doivent être gardées secrètes pendant des décennies.

Les systèmes usuels de cryptographie sont fondés sur la complexité algorithmique de leur déchiffrement, qui permettent la sécurisation numérique des données. En particulier, la cryptographie a clé publique (ou cryptographie asymétrique) permet actuellement des communications sécurisées sur Internet. Elle se base cependant sur des hypothèses calculatoires, comme la difficulté de factorisation des grands nombres pour le protocole **RSA** [1] ; elle est donc de façon inhérente vulnérable aux futures avancées matérielles et algorithmiques,y compris la construction d'un ordinateur quantique de grande taille. Cette menace devenant plausible dans un horizon moyen-terme grâce a des récents progrès fulgurants, il devient crucial de faire évoluer nos pratiques cryptographiques .

Depuis sa découverte en 1927, le **principe d'incertitude de Heisenberg** a souvent été perçu comme une limitation fondamentale imposée par la mécanique quantique aux mesures physiques. La cryptographie quantique exploite cette « faiblesse » et en fait une force, permettant de garantir un secret absolu sur des communications cryptées. Fondée sur une idée originale de **S. Wiesner** [2] refusée en 1969 par une revue scientifique . Wies-

1. Rivest, Adi Shamir et Leonard Adleman ont publié leur chiffrement en 1978 dans A Method for Obtaining Digital Signatures and Public-key Cryptosystems.

2. Stephen J. Wiesner (né en 1942) est un physicien de recherche . En tant qu'étudiant diplômé de l'Université Columbia à New York à la fin des années 1960 et au début des années 1970, il a découvert plusieurs des idées les plus importantes de la théorie de l'information quantique .

ner, dans son papier, il a premièrement proposé en 1970 la **CQ**[3] qui n'avait pas été publié jusqu'en 1983 , ou plus précisément Distribution de Clé Quantique, propose une méthode alternative aux méthodes mathématiques. Elle se base sur les lois de la physique et vise à assurer la sécurité inconditionnelle de l'échange de clé. Elle a ouvert devant les scientifiques un nouvel axe de recherche pour résoudre le problème courant de la cryptographie ,en 1984, C.H. Bennent et G. Brassard ont développé un protocole d'échange quantique de clés nommé BB84 basé sur le principe d'incertitude de Heisenberg, ils ont prouvé que ce protocole est inviolable grâce au principe du non-clonage .

La cryptographie sécurise l'information transmise de plusieurs manières :

- **La confidentialité** :seul le destinataire peut récupérer la version non cryptée de l'information transmise ;
- **L'intégrité** :l'information n'a pas été modifiée pendant sa transmission ;
- **L'authentification** :chacun est bien celui qu'il prétend être ;
- **La non-répudiation** :l'émetteur ne peut pas nier avoir transmis l'information cryptée **;**
- **Le contrôle d'accès** :seuls les personnes autorisées par l'émetteur et le récipiendaire peuvent accéder à l'information non cryptée **;**

Dans cet humble travail de mémoire de fin d'études il nous a été demandé par un cahier des charges de faire une étude et analyse d'un système d'échange de clé quantique (**QKD**)[4] qui, semble-t-il, selon la littérature est bien sécurisée et de l'appliquer pour un crypto-système de Vernam.

Pour ce faire, nous avons opté à pour la repartition notre mémoire en trois chapitres :

- Le premier chapitre est dédié aux généralités de la cryptographie et l'échange de clé. Nous exposerons quelques définitions, les concepts de sécurité et les menaces cryptographiques, ensuite les types de la cryptographie moderne, puis la gestion de clé.
- Le deuxième chapitre présente les principes de base de la mécanique quantique, les protocoles de distribution de clés ainsi que la polarisation des photons .
- Le dernier chapitre constitue l'essentiel du travail, où nous analysons et simulons à l'aide du MATLAB le protocole de distribution de clé quantique **BB84**[5] son aspect probabiliste,ensuite nous exploitons la clé obtenue pour le chiffrement de Vernam d'un message clair. nous terminons ce mémoire par une conclusion et perspectives.

3. Cryptographie quantique

4. La distribution de clés quantiques est une méthode de communication sécurisée qui implémente un protocole cryptographique impliquant des composants de la mécanique quantique

5. le protocole BB84 est le premier mécanisme d'échange de clé quantique à avoir été formalisé, et en fait le premier protocole de cryptographie quantique. Il a été proposé en 1984 par Charles Bennett et Gilles Brassard

CHAPITRE 1

GÉNÉRALITÉS SUR LA CRYPTOGRAPHIE ET L'ÉCHANGE DE CLÉS

1.1 Introduction

Le développement technologique relatif au domaine de la communication de l'information connait, à nos jours une progression intéressante dans les applications de la téléphonie, internet, Wifi etc. Ces applications sont exploitées par des individus dans le domaine politique, économique, militaire et autres domaines. Cette progression porte un risque potentiel pour la vie privée des individus et la confidentialité des informations, par conséquent la protection de l'information est nécessaire par les techniques cryptographiques[1].

Dans ce chapitre nous définissons d'abord quelques termes et notations importantes. Ensuite, nous décrivons l'évolution de la cryptographie au cours du temps jusqu'à la cryptographie moderne et l'échange de clés cryptographiques.

1.2 Définitions et terminologie :

1.2.1 Définitions

La cryptologie est la science qui traite de la communication en présence d'adversaire. Le but d'un système cryptographique est de chiffrer un texte clair en un texte chiffré en utilisant une clé. Ce texte chiffré est ensuite transmis a son destinataire sur un canal. Le destinataire légitime doit pouvoir de´chiffrer le texte chiffré a l'aide d'une clé pour obtenir le texte clair original. La cryptologie comprend principalement deux champs d'étude, la cryptographie et la cryptanalyse (Voir figure 1.1)[1].

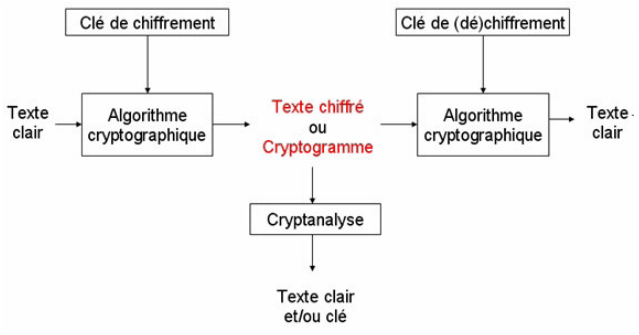


FIGURE 1.1 – Protocole de chiffrement

La cryptographie

La cryptographie Est l'art de rendre inintelligible, de crypter, de coder, un message pour ceux qui ne sont pas habilités à en prendre connaissance[1].La cryptographie a pour but d'assurer :

- **Confidentialité :** C'est Garantir que le contenu d'une communication ou d'un fichier ne soit pas accessible aux personnes non concernés.

- **l'authenticité :** Le destinataire d'un message doit être certain de son origine.

- **l'intégrité des informations :** C'est assurer la non-modification de l'information.

– **Non-répudiation :** un émetteur ne doit pas pouvoir nier l'envoi d'un message.

La cryptanalyse

La cryptanalyse Art de « casser » les messages chiffrés d'un cryptosystème.

1.2.2 Terminologies :

- **Le chiffrement :** l'action de chiffrer un message en clair, en un message chiffré, et cela de façon à ce qu'il soit impossible de retrouver le message en clair à partir du message chiffré sans la clé.

- **Le déchiffrement :** L'action inverse du chiffrement. Cette action s'effectue uniquement en possession de la clé secrète.

- **Clé :** Il s'agit du paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement. Dans le cas d'un algorithme symétrique, la clé est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, les clés sont différentes pour les deux opérations.

- **Cryptosystème :** Il est défini comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné **(Voir figure 1.2)**.

- **Protocole :** description de l'ensemble des données nécessaires pour mettre en place le mécanisme de cryptographie : ensemble des messages clairs, des messages cryptés, des clés possibles, des transformations**[1] .**

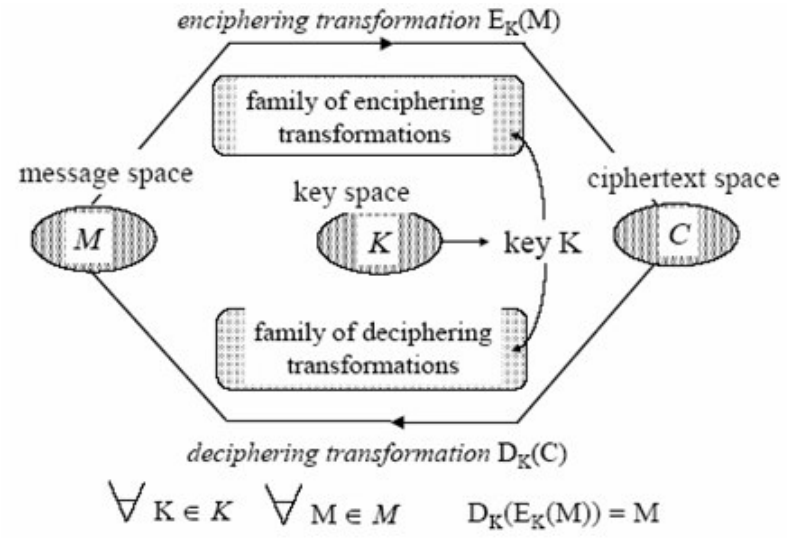


FIGURE 1.2 – Schéma d'un Cryptosystème

1.3 Notions mathématiques

On définie quelques structures algébriques que nous utiliserons par la suite.

1.3.1 Structures algébriques

Groupe

Un groupe est un couple (E,$*$) ou E est un ensemble et est (*) une loi de composition interne qui combine des éléments de E pour obtenir un troisième élément appartenant à E .

Il faut que la loi satisfasse les quatre axiomes ci-dessous :

- **Fermeture** : $\forall$(a,b) $\in$ E$|$ (a$*$ b) $\in$ E

- **Associativité** : $\forall$(a,b)$\in$E$|$ (a$*$ b)$*$ c = a$*$ (b $*$c)

- **Élément neutre** : $\exists_e \in$ E $|$ a $*$ e = e $*$a = a

- **Symétrique** : $\forall$a$\in$ E ;$\exists$b $\in$ E$|$ a$*$ b = b $*$a = e $\Longrightarrow$ b =a^{-1}.

1. **Sous groupes**

Un sous-groupe est une partie du groupe G qui a la même structure de la loi de composition interne de G.

2. **Groupe fini**

Un groupe fini est un groupe dont le nombre d'élément est fini. Ce nombre d'élément est appelé l'ordre du groupe , qui représente le cardinal de l'ensemble fini .

3. **Groupe abélien**

Un groupe est dit abélien ou commutatif si la loi de composition interne est commutatif.

$\forall$(a,b) $\in$ E $|$ a$*$ b = b$*$ a (1.1)

4. **Groupe cyclique**

Un groupe fini G est dit cyclique s'il existe un élément particulier g tel que tous les éléments du G puissent s'exprimer sous forme d'une puissance (en notation multiplicative) ou d'un multiple (en notation additive) de g. ce dernier est appelé l'ordre du groupe cyclique .

Tout groupe cyclique est abélien car : $\forall n$,m $\in Z$

$g^n g^m = g^{(n+m)} = g^{(m+n)}$ (en notation multiplicative) (1.2)

n * g = g * n (en notation additive) (1.3)

Noter que la réciprocité est fausse. L'ordre d'un élément e d'un groupe cyclique est le nombre entier n positif le plus petit tel que e * n = 0 (en notation additive) ou en e^n = 1 (en notation multiplicative) .

Anneau

Un anneau est un ensemble non vide E sur lequel sont définis deux opérations, notées respectivement +(addition) et ∗ (Multiplication). E est un anneau si les conditions suivantes sont vérifiées :

- E muni de son addition est un groupe commutatif.

- La multiplication est associative : ∀x,y, z ∈ A, x*(y*z) = (x*y)*z .

- La multiplication est distributive par rapport 'a l'addition :

$$\forall x,y,z \in A,\ x*(y+z) = (x*y) + (x*z) \tag{1.4}$$

- la loi + possède un élément neutre, noté 0 tel que : $\forall x \in Z^* \mid x + 0 = 0 + x = x$ - et -$\forall x \in E, \exists y \in E \mid x + y = 0$ (1.5)

Si la multiplication possède un élément neutre, noté 1 tel que : $\forall x \in Z \mid x*1 = 1*x = x$, l'anneau est dit unitaire.

Un anneau est dit commutatif si la loi est commutative .

Corps

Un corps est un anneau dont les éléments non nuls possèdent un inverse pour la loi de composition interne (α) [multiplication].

Corps fini

Un corps fini F est un corps commutatif dont le nombre d'éléments est fini. Il est déterminé par son cardinal qui est toujours une puissance d'un nombre premier $q = p^n$ avec p un nombre premier appelé la caractéristique du corps et $n \in Z+$.

1.Corps premier : Un corps premier F_p est un corps fini constitué uniquement des nombres entiers.$[0; 1; 2; 3.....; p-1]$, $\forall a \in Z$; $r = a \ mod\ p$ ou l'ordre du corps est $q = P^n$ avec p un nombre entier et r le reste unique compris entre $[0, p-1]$.

2.Corps binaire : Un corps fini de l'ordre 2^n est un corps binaire de caractéristique p = 2 , noté F_{2^n} , qui peut être construit en utilisant une représentation polynomiale .

Les éléments du corps sont des polynômes binaires dont les coefficients $a_i \in \{0, 1\}$ et les degrés sont inférieurs à n .

C'est-à-dire : $F_{2^n} = a_{n-1}z^{n-1} + a_{n-2}z^{n-2} + ... + a_0$ (1.6)

1.4 Histoire de la cryptographie :

L'histoire de la cryptographie s'est faite dès la création de l'écriture. En effet à partir du moment où l'on a pu conserver les grands faits de l'homme, elle a débuté. Sans l'écriture la cryptologie n'aurait jamais existé [2].

L'être humain a toujours eu le besoin de cacher des informations. Voici une liste non exhaustive de différentes techniques utilisées au fil des siècles qui marquent l'évolution de la cryptographie à travers les âges. Les premières traces de cryptographie remontent à l'Antiquité, plus précisément aux alentours du XVIème siècle avant J.-C. Un potier en Irak avait gravé sur une table en argile sa recette en supprimant les consonnes et en modifiant l'orthographe des mots .

1.4.1 Méthode de Scytale

Au Xème et VIIème siècle avant J.C., les Grecs ont utilisés une technique de chiffrement dite « par transposition » à des fins militaires. Cela consistait à effectuer des changements de positions de lettres dans le message. L'outil employé est connu sous le nom de « Scytale » (Voir figure 1.3), autour duquel ils enroulaient en spires jointives une bande de cuir et y inscrivaient le message. Une fois déroulé, le message était envoyé au destinataire qui devait posséder un bâton identique, c'est à dire ayant le même diamètre, pour le déchiffrage. Cette technique fut vite obsolète car son seul avantage réside dans le procédé de chiffrement : un bâton de diamètre Quasi-identique suffit à déchiffrer le texte [3] .

FIGURE 1.3 – La Scytale grec .

1.4.2 Chiffre de César

Quelques siècles plus tard, Jules César a également utilisé un procédé de sécurisation de données. Il utilise un chiffre qui consiste simplement à décaler les lettres dans l'alphabet. L'exemple de la figure 1.3, montre un décalage à droite des lettres dans l'alphabet français par trois positions (Voir figure 1.4) .

Cependant le message en texte clair « attaque demain » donne le message chiffré « CVVCSWG FGOCKP ». Cette méthode n'est pas efficace car un simple teste peut détecter la clé du message [4].

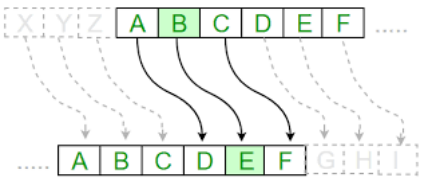


FIGURE 1.4 – Principe du chiffrement de César

1.4.3 Chiffre de Vigenère

Le chiffre de Vigenère est une amélioration décisive du chiffre de César, Sa force réside dans l'utilisation non pas d'un, mais de 26 alphabets décalés pour chiffrer un message (Voir tableau 1.1) On peut résumer ces décalages avec un carré de Vigenère. Ce chiffre utilise une clé qui définit le décalage pour chaque lettre du message[4].

Si on veut al'aide du tableau de Vignére chiffrer le texte «CHIFFRE DE VIGENERE» par la clé BACHELIER on obtient le texte chiffrer D H K M J C M H V W I I L R P Z I .

Message clair	C	H	I	F	F	R	E	D	E	V	I	G	E	N	N	R	E
clé	B	A	C	H	E	L	I	E	R	B	A	C	H	E	L	I	E
Décalage	1	0	2	7	4	11	8	4	17	1	0	2	7	4	11	8	4
Message chiffré	D	H	K	M	J	C	M	H	V	W	I	I	L	R	P	Z	I

TABLE 1.1 – Chiffre de Vigénére

1.4.4 Chiffre de Hill (1929)

Lester Hill, mathématicien cryptographe (1891-1961) publie en 1929 dans la revue American Mathematical Monthly un article intitulé Cryptography in an algebraic alphabet, où il détaille un nouveau type d'algorithme de chiffrement. Son idée est de continuer à utiliser des décalages du même type que celui du chiffre de César, mais en effectuant c*Introduction*es décalages simultanément sur des groupes de m lettres. Bien sûr, plus m est grand, plus les analyses statistiques deviennent difficiles[6].

En cryptographie symétrique, le chiffre de Hill est un modèle simple d'extension du chiffrement affine à un bloc. Ce système étudié par Lester S. Hill [6], utilise les propriétés de l'arithmétique modulaire et des matrices. Il consiste à chiffrer le message en substituant les lettres du message, non plus lettre à lettre, mais par groupe de lettres. Il permet ainsi de rendre plus difficile le cassage du code par observation des fréquences. Lester S. Hill a aussi conçu une machine capable de réaliser mécaniquement un tel codage (Voir figure 1.5).

$$M = \begin{pmatrix} a & b \\ 5 & 3 \end{pmatrix} \qquad N = \frac{1}{\det(M)} \begin{pmatrix} 3 & -b \\ -5 & a \end{pmatrix} \qquad Q = \begin{pmatrix} 6 & 3 \\ 5 & 3 \end{pmatrix} \qquad X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

tableau de correspondance

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

FIGURE 1.5 – Principe du chiffrement de Hill

1.5 Quelques concepts de sécurité

La théorie quantique a quelques propriétés intéressantes qui peuvent être exploitées dans le développement systèmes cryptographiques : cependant, avant de se lancer dans les merveilles du monde quantique, il est nécessaire de faire un bref détour par Quelques concepts de sécurité.

1.5.1 Types de sécurité

Il existe deux types de sécurité cryptographique qui seront pertinents dans ce rapport : la sécurité informatique et la sécurité de l'information (aussi appelée inconditionnelle ou sécurité parfaite).

Sécurité informatique

Ceci décrit un crypto système qui est théoriquement cassable (en essayant toutes les clés possibles - l'attaque par force brute) mais l'effort de calcul requis pour le faire est si long et couteux qu'il n'est pas économiquement viable pour un attaquant de prendre en compte.

Sécurité théorique de l'information

Cela décrit les cas où, même si un attaquant dispose de ressources infinies, le crypto-système ne peut tout simplement pas être cassé. Ceci est clairement beaucoup plus fort que la sécurité informatique, mais n'est pas nécessairement réalisable.

Le père fondateur de la théorie de l'information, Claude Shannon , a prouvé qu'une sécurité inconditionnelle était possible si la clé secrète avait la même longueur que le message en clair à chiffrer.

Théorie de l'information à diverses utilisations en cryptographie : il peut être utilisé pour prouver la sécurité inconditionnelle des systèmes, déterminer la possibilité de sécurité inconditionnelle dans les limites supérieures et inférieures, ou réduire la tâche de rupture d'un crypto système jusqu'à l'équivalence d'une rupture de l'une de ses primitives cryptographiques sous-jacentes (par exemple une fonction unidirectionnelle), peut être une tâche tout à fait plus facile [6].

1.5.2 Chiffre de Vernam (The One Time Pad)

Dans sa preuve, Shannon a utilisé un cas particulier de chiffrement symétrique pour fournir une sécurité inconditionnelle : le One Time Pad (OTP), inventé en 1926 par Vernam et Mauborgne . Il existe des exigences fondamentales pour l'utilisation de l'OTP :

- La clé est aléatoire et non répétitive .
- La clé est aussi longue que le message.
- La clé est utilisée une seule fois et ensuite rejetée - jamais réutilisée.

Si ces conditions sont remplies, une opération de chiffrement simple (telle qu'un XOR logique) produira un texte chiffré incassable. Même si un attaquant a une puissance de

calcul infinie, il ne sera pas capable de dériver des informations à partir d'un texte chiffré intercepté. Aussi intéressants soient-ils en théorie, les OTP ont d'immenses difficultés pratiques : générer des clés longues et vraiment aléatoires est problématique, distribuer les clés aux destinataires est un cauchemar logistique, l'expéditeur et le récepteur doivent être totalement synchronisés pour s'assurer que les mêmes clés sont utilisées pour le même message, et s'assurer que les clés ne sont jamais réutilisées. Pour cette raison, les OTP sont actuellement rarement utilisés dans la pratique, mais dans les sections suivantes de ce rapport, il sera montré qu'ils deviennent une perspective beaucoup plus attrayante lorsqu'il est utilisé en conjonction avec les protocoles QKD (Voir figure 1.6) [5].

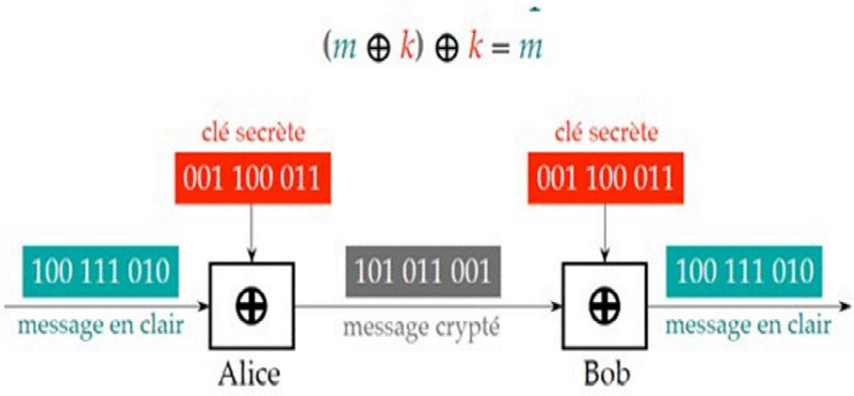


FIGURE 1.6 – Exemple de chiffre de Vernam

1.5.3 Principe de Kerckhoffs

Le principe de Kerckhoffs est l'un des principes de base de la cryptographie moderne. Il a été formulé à la fin du XIXe siècle par le cryptographe néerlandais Auguste Kerckhoffs [1]. Le principe est le suivant : ***« Un système cryptographique doit être sécurisé même si tout ce qui concerne le système, à l'exception de la clé, est de notoriété publique ».***

Les publications les plus connues de Kerckhoffs sont deux articles publiés en 1883 dans le "Journal des Sciences Militaires" sous le titre commun "La Cryptographie Militaire". Les articles traitaient des solutions de cryptographie militaire les plus récentes à ce moment-là. Ils ont donné une approche pratique basée sur l'expérience, y compris six principes de conception pour les chiffrements militaires :

- Le système doit être pratiquement, indéchiffrable.
- IL ne doit pas être exigé d'être secret, et il doit pouvoir tomber dans les mains de l'ennemi sans inconvénient.
- La clé doit être communicable et conservable sans l'aide de notes écrites, et changeable ou modifiable au gré des correspondants.
- Il doit être applicable à la correspondance télégraphique.
- Les appareils et les documents doivent être portables, et leur utilisation et leur fonction ne doivent pas nécessiter le concours de plusieurs personnes.

1. Auguste Kerckhoffs von Nieuwenhoff (19 janvier 1835 - 9 août 1903) est un cryptologue militaire néerlandais.

- Enfin, il est nécessaire, compte tenu des circonstances qui commandent son application, que le système soit facile à utiliser, ne nécessitant ni effort mental ni connaissance d'une longue série de règles à observer.

- Le principe de Kerckhoffs est appliqué dans pratiquement tous les algorithmes de cryptage moderne (DES, AES, etc.). Ces algorithmes sont considérés comme surs et soigneusement étudiés.La sécurité du message chiffré dépend uniquement de la sécurité de la clé de chiffrement secrète. .

Garder les algorithmes secrets peut constituer un obstacle important à la cryptanalyse, mais seulement si ces algorithmes sont utilisés dans un cercle strictement limité, ce qui protège l'algorithme d'être révélé. La plupart des chiffrements gouvernementaux sont tenus secrets. Les algorithmes commerciaux de cryptage, lancés sur le marché, ont pour la plupart été brisés assez rapidement [6].

1.5.4 Les menaces cryptographiques

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 [6] :

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

2. ***Les menaces intentionnelles*** : 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 actives appartiennent principalement à quatre catégories illustrées à la figure 1.7 :

- **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

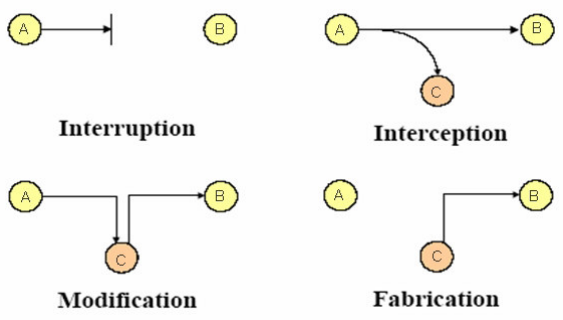


FIGURE 1.7 – Types de menaces actives

1.6 la cryptographie moderne

Aujourd'hui, des centaines de millions d'individus, à travers le monde, ont en permanence sur eux un ou plusieurs processeurs cryptographiques, pour leur téléphone mobile ou leur carte bancaire et cherchent à protéger les codes d'accès ou clés.

Dans la cryptographie moderne toute la sécurité est basée sur cette clé et non dans les détails des algorithmes utilisés. Cela signifie qu'un algorithme peut être publié et analysé, mais la clé doit rester secrète [7] (Voir figure 1.8).

La cryptographie moderne se décompose en deux types :

- La cryptographie à clés secrètes ou cryptographie symétrique.
- La cryptographie à clés publiques ou cryptographie asymétrique.

Ils ont tous les deux leurs avantages et leur inconvénients. La différence qui existe entre ces deux types se situe au niveau de la clé.

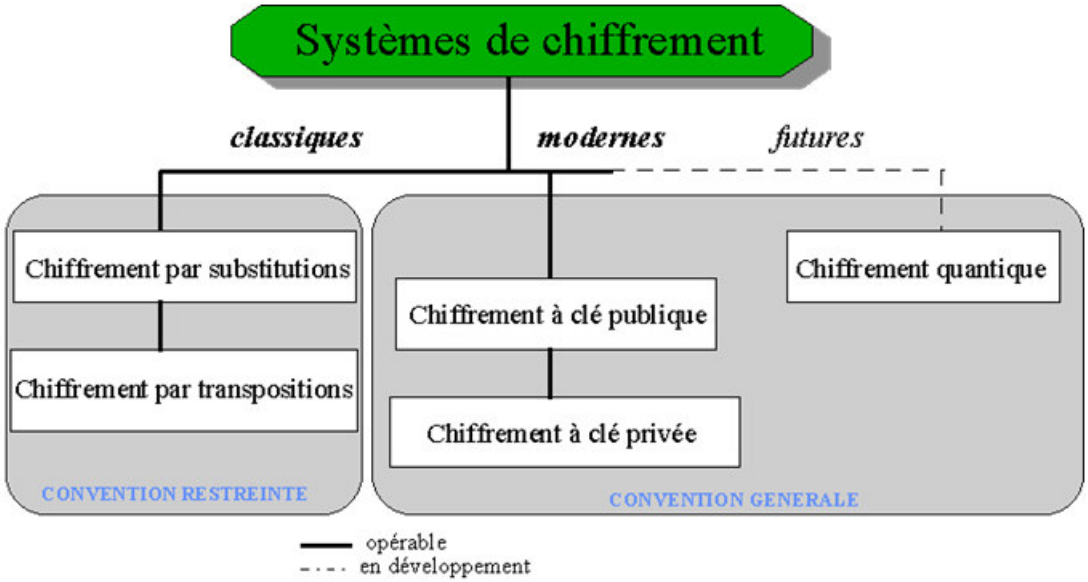


FIGURE 1.8 – les systèmes de chiffrement

1.6.1 La cryptographie symétrique

Le cryptage symétrique ou chiffrement à clé privée ou clé secrète a besoin d'une seule clé pour crypter et déchiffrer le message. Cette clé commune partagée entre l'émetteur et le récepteur doit être tenue secrète. L'avantage est que ces algorithmes sont assez rapides et permettent le cryptage d'un grand nombre de données. Mais l'inconvénient est que la clé doit être transmise par un canal secret [8] (Voir figure 1.9).

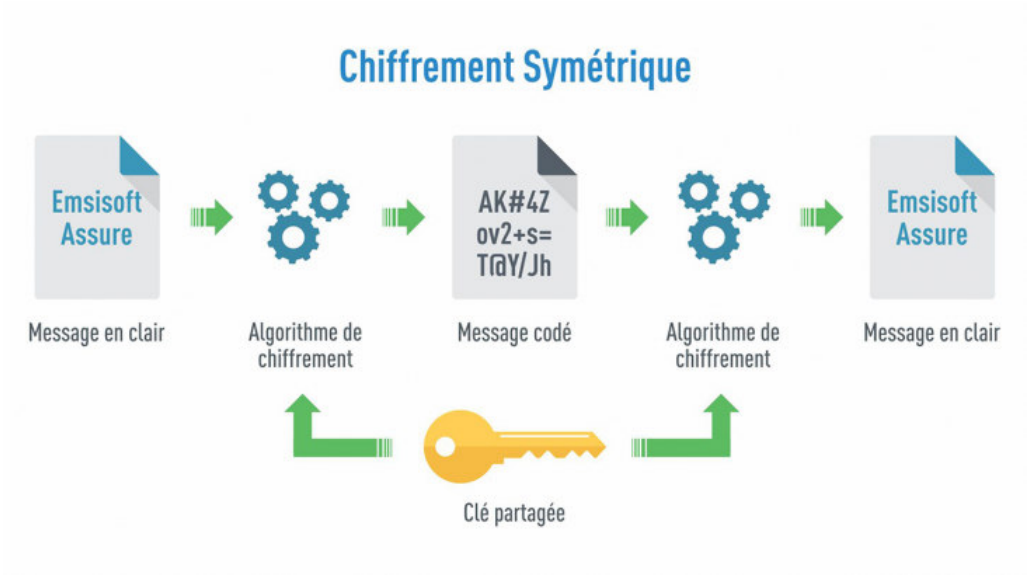


FIGURE 1.9 – La cryptographie symétrique.

Les méthodes de chiffrement symétrique se divisent naturellement en deux familles, le Chiffrement par bloc et le chiffrement par flot décrites ci-dessous :

Chiffrement par flot

Dans un cryptosystème par flots, le cryptage des messages se fait caractère par caractère ou bit par bit, au moyen de substitutions générées aléatoirement, la taille de la clé est donc égale à la taille du message.

Un des algorithmes de chiffrement par flot le plus repandu est RC4, il a été conçu en 1987 par Ronald Rivest (Voir figure 1.10). [8].

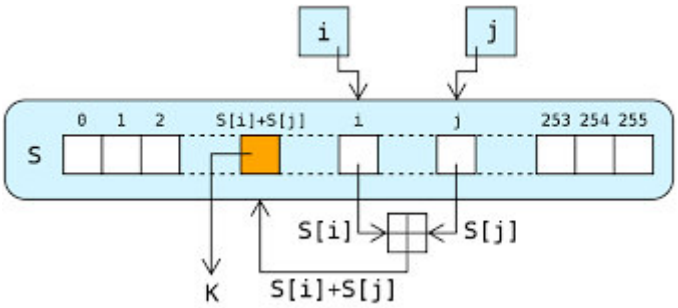


FIGURE 1.10 – L'algorithme RC4 de chiffrement par flot

Chiffrement par bloc

Dans un algorithme de chiffrement par bloc, chaque message clair est découpé en blocs de taille fixe de même longueur et chiffré à l'aide d'une clé unique. Ces algorithmes sont en général construits sur un modèle itératif. Il utilise une fonction $\boldsymbol{F}$ qui prend une clé secrète $\boldsymbol{k}$ et un message $\boldsymbol{M}$ de n bits. La fonction $\boldsymbol{F}$ est itérée un certain nombre de fois (nombre de tours). Lors de chaque tour, la clé $\boldsymbol{k}$ est différente et on chiffre le message qui vient d'être obtenu de l'itération précédente. Les différentes clés $\boldsymbol{k(i)}$ qui sont utilisées sont déduites de la clé secrète $\boldsymbol{k}$. Les deux algorithmes de chiffrement par bloc les plus repandu sont DES et AES [9].

Avantage

- Adapté au grand flux de données à chiffrer.

- Simple et facile à implémenter.

Inconvénients

- Nécessite la connaissance de la clé par l'émetteur et par le destinataire.

- Toute personne interceptant la clé lors d'un transfert peut ensuite lire ou même modifier ou falsifier toutes les informations cryptées.

Les algorithmes les plus connus des systèmes cryptographique symétriques sont : l*e* *DES*[2] et l'*AES*.

DES (Data Encryptions Standard)

Le DES est un algorithme de chiffrement par blocs produit par le ***NIST***[3] et consiste à découper les données en blocs de taille fixe. Le point faible de cet algorithme reste sa clé qui est de petite taille (64 bits), et non variable, ce qui n'offre pas la possibilité de l'adapter aux différents niveaux de sécurité (Voir figure 1.11)[14].

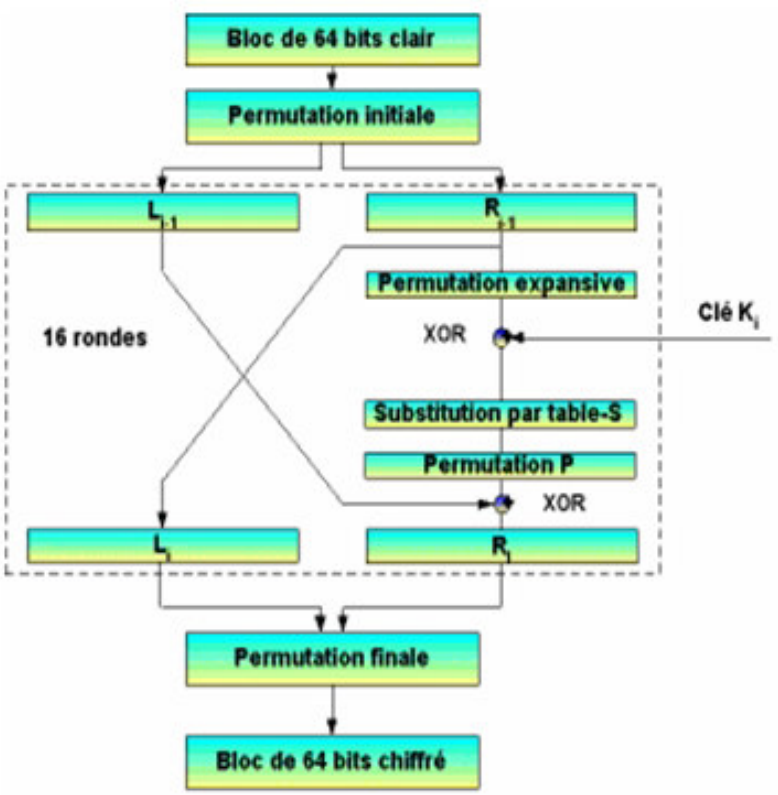


FIGURE 1.11 – Algorithme principal du DES

2. Data Encryptions Standard
3. National Institute of Standards and Technologies

AES (Advanced Encryption Standard)

Tant que la taille de la clé de DES est de 64 bits ; qui est trop courte pour les puissances de calcul actuelles, et tant que en 1997 la clé a été cassée ; le DES n'est plus sûr et adapté au besoin. En janvier 1997, le NIST (National Institute of Standards and Technologies) des Etats-Unis lance un appel d'offres pour une nouvelle méthode, appelée, ***AES***[4] qui devrait satisfaire les besoins suivants [11] :

- Une grande sécurité.
- La rapidité.
- Une lecture facile de l'algorithme.
- La méthode doit utiliser des clés plus grandes que DES.

L'AES est un standard de cryptage symétrique destiné à remplacer le DES qui ne résiste pas aux attaques. L'AES est un algorithme de chiffrement par bloc utilisant des clés de 128 bits, 192 ou 256 bits. Enfin, les algorithmes de chiffrement par bloc sont plus rapides que les algorithmes par flot à cause de leur parallélisme. Sans oublier que tout secret doit être dans la clé et pas dans l'algorithme car les algorithmes de chiffrement sont connus par tout le monde (Voir figure 1.12) [8].

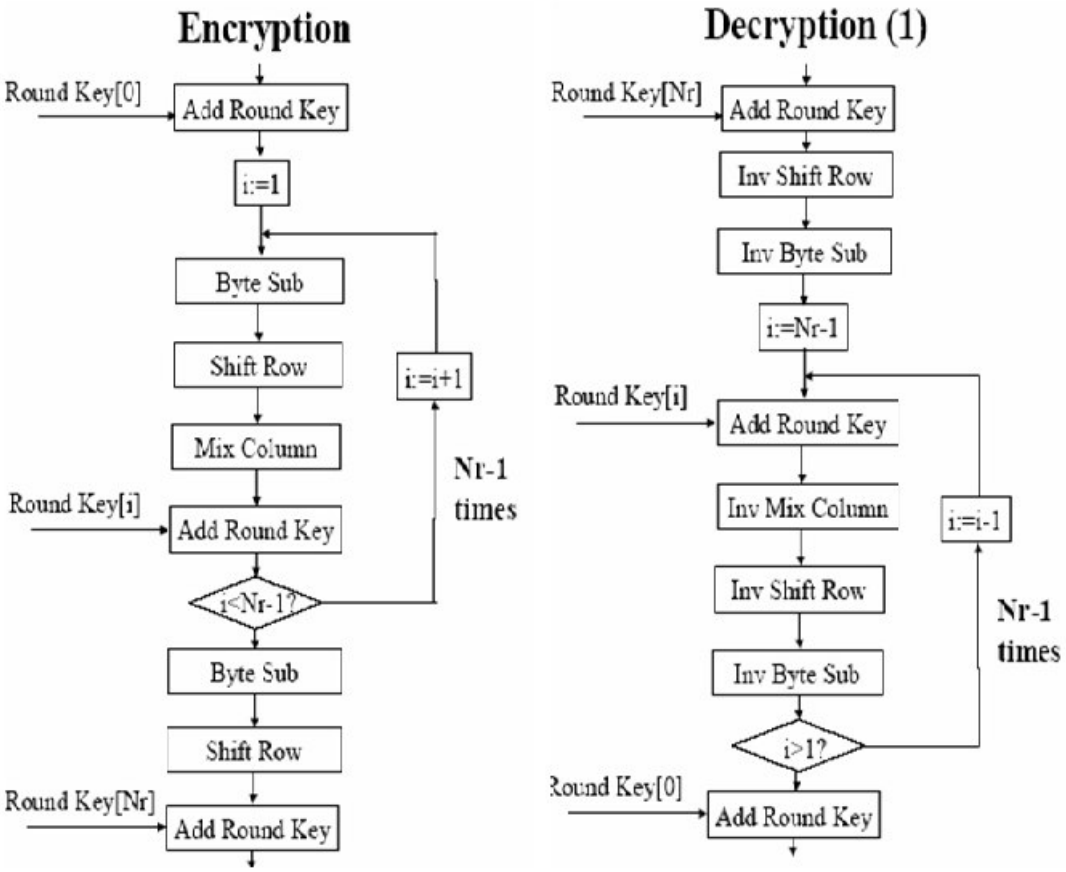


FIGURE 1.12 – Algorithme principal du AES

4. Advanced Encryption System

1.6.2 La cryptographie asymétrique

Le cryptage asymétrique ou chiffrement à clé publique fonctionne sur le principe d'une paire de clés, Ne demande aucun échange préalable de secret. Chaque utilisateur possède deux clés mathématiquement liées : l'une est publique, l'autre est privée, La clé publique est accessible à tout le monde et permet de crypter les messages. La clés privée,qui elle doit être tenue secrète. sert à décrypter le message chiffré. On s'aperçoit alors que tout le monde peut crypter un message et que seule la personne détenant la clé secrète peut le décrypter. En plus il est impossible de déduire la clé privée de la clé publique [12] (Voir figure 1.11).

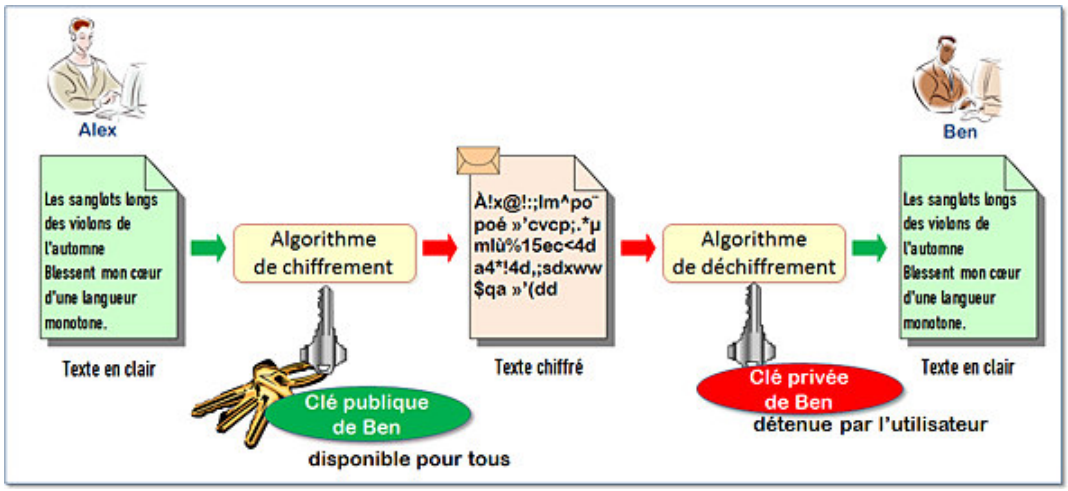


FIGURE 1.13 – La cryptographie asymétrique

Mathématiquement, le chiffrement asymétrique est basé sur des fonctions à sens unique qui est une fonction facile à calculer et difficile à inverser. Casser ce type de chiffrement ne veut pas dire nécessairement essayer toutes les clés possibles (attaque exhaustive), mais signifie trouver l'inverse de cette fonction. La difficulté ici est comprise dans le sens calculatoire, elle dépend des ressources de calcul que dispose l'espion et du temps de calcul [13] .

Le protocole le plus connu dans le chiffrement asymétrique est le protocole *RSA*[5].

Avantage

- L'échange des messages de manière sécurisé.
- L'expéditeur et le destinataire n'ont plus besoin de partager des clés secrètes via une voie de transmission sécurisée.
- Les communications impliquent uniquement l'utilisation de clés publique et aucune clé privée n'est transmise ou partagée.

5. *Rivest, Shamir et Adleman*

Inconvénients

- Le traitement de données est lent et demande beaucoup de calculs.

RSA (Rivest, Shamir et Adleman)

Le cryptage RSA, du nom de ses concepteurs, Ron Rivest, Adi Shamir et Leonard Adleman, est le premier algorithme de chiffrement asymétrique. Il a été découvert en 1977 Le RSA est basé sur la théorie des nombres premiers, et sa puissance vient du fait qu'il n'existe aucun algorithme qui permet la décomposition d'un grand nombre en deux facteurs premiers [14].

1.7 La gestion des clés

La gestion des clés est principalement constituée de quatre domaines [15] :

1. **La génération des clés** : il faut prendre garde aux caractères choisis, aux clés faibles, ... et veiller à utiliser des générateurs fiables,
2. **Le transfert de la clé** : l'idéal est de se rencontrer, ou d'utiliser un canal de transmission protégé. Mais cela est souvent impossible. Aussi, si A et B ont des communications sures avec un tiers C, ce dernier peut relayer la clé entre A et B. Un tiers est un intermédiaire de confiance, à qui tous les usagers font confiance, pour négocier l'établissement de transmissions sures entre elles,
3. **La vérification des clés** : par hachage, ou utilisation de certificats,
4. **Le stockage des clés** : que ce soit dans des fichiers, sur supports extérieurs.

1.7.1 Distribution des clés

Clés symétriques

Dans ce cas, il est nécessaire pour les deux usagers de partager une clé secrète commune. Bien souvent, l'échec d'un système sûr est dû à une rupture dans le schéma de distribution des clés [16].

Comment distribuer surement cette clé ? .

- Physiquement : par une rencontre, un canal de transmission protégé,
- Utiliser un tiers de confiance (Celui-ci choisit et fournit la clé),
- Utiliser une ancienne clé pour chiffrer une nouvelle clé (ce qui suppose cependant un échange préalable de cette ancienne clé),
- Distribution automatique de clés à la demande des utilisateurs (Cette solution existe, mais elle nécessite une totale confiance au système).

Clés asymétriques

Le chiffrement par clé publique permet de résoudre les problèmes de distribution de clés secrètes. Malgré tout, la distribution des clés publiques continue de poser problème, principalement au niveau de l'authentification des utilisateurs liés à ces clés . Les différentes techniques de distribution d'une clé publique sont regroupées dans les catégories suivantes :

— Annonce publique .
— Annuaire publiquement disponible.
— Autorité de clés publique .
— Certificats de clé publique.

Annonce publique

La distribution des clés publiques se fait directement aux destinataires ou par broadcaste à la communauté. Il est par exemple possible d'apposer les clés aux emails ou les poster dans des newsgroups ou mailing-liste. Mais le risque majeur avec cette méthode est la contrefaçon : n'importe qui peut créer une clé en prétendant être quelqu'un d'autre et la publier. La mascarade continuera tant que la contrefaçon n'est pas découverte [10].

Annuaire publiquement disponible

On enregistre ici des clés dans un annuaire public, ce qui implique de faire confiance à cet annuaire. Ce dernier doit avoir plusieurs propriétés :

— Il doit contenir les entrées {nom, clé publique},
— Il doit être possible de s'inscrire de manière sécurisée dans l'annuaire,
— On doit pouvoir remplacer la clé à tout moment,
— L'annuaire doit être publié périodiquement,
— Il devrait également permettre la consultation électronique.

Même si il est clairement plus sûr que les annonces publiques individuelles, il reste vulnérable. Il est en effet nécessaire que l'annuaire soit sécurisé. Dans le cas contraire, un individu pourrait détourner l'annuaire et fournir des clés publiques contrefaites, voire à ne pas transmettre les clés correspondant aux demandes des entités communicantes [6].

Autorité de clés publique

Il s'agit de renforcer le contrôle de la distribution des clés à partir de l'annuaire. Il

dispose des mêmes propriétés que ce dernier. Cependant, la sécurité est renforcée. En effet, dans le cas présent, chaque entité dispose de la clé publique de l'autorité. Ainsi, lorsqu'une entité désirera obtenir la clé publique d'un correspondant, il enverra une requête à l'autorité. Celle-ci contiendra la requête proprement dite et un marqueur temporel (timestamp). En retour, l'autorité renverra la clé demandée, le timestamp pour prouver le non-rejeu d'un ancien message, le tout chiffré avec sa clé privée. De cette manière, l'entité A, possédant la clé publique de l'autorité, pourra vérifier la bonne provenance de la clé publique de B. L'entité B pourra pratiquer de la même manière [6].

Certificats de clé publique

Il est possible d'opter pour les certificats pour distribuer les clés publiques de façon sécurisée. Chaque entité peut alors contacter une entité de certificat pour la création de son certificat. L'entité en question pourra, par la suite, diffuser non plus sa clé publique, mais son certificat qui contient, entre autres, sa clé publique . Ainsi l'entité A voulant communiquer avec B récupèrera son certificat où elle trouvera la clé publique de B mais pourra aussi vérifier son authenticité auprès de l'autorité de certificat [6].

1.8 L'échange des clés cryptographiques

1.8.1 L'Échange de clé Diffie-Hellman

Échange de clé de Diffie-Hellman [6] utilisé d'une fonction à sens unique comme l'exponentiation modulaire permet le partage de clé. Supposons donc que Alice et Bob souhaitent partager une clé secrété **k**. Ils conviennent d'abord d'un entier premier p et d'un générateur g du groupe cyclique $\mathbf{Z}_P$. On peut alors définir le protocole suivant dans les étapes suivantes leur permettant de construire **k** en secret :

1. Seule, Alice choisit un nombre $\mathbf{a} \in \mathbf{Z}_P$ secret et calcule $\mathbf{A} = \mathbf{g}^a \bmod \mathbf{p}$ puis Alice envoie **A** à Bob.

2. Symétriquement, Bob choisit un nombre $b \in \mathbf{Z}_P$ secret et calcule $\mathbf{B} = \mathbf{g}^b \bmod \mathbf{p}$ puis Bob envoie **B** à Alice.

3. Alice calcule seule $\mathbf{k} = \mathbf{B}^a \ \mathbf{mod\ p.}$

4. Symétriquement, Bob calcule de son coté $\mathbf{A}^b \mathbf{mod\ p}$.

A la fin, Alice et Bob partagent la même clé secréte $\mathbf{K} = \mathbf{g}^{b*a} \mathbf{mod\ p}$ sans l'avoir jamais communiquée directement[15].

Cet algorithme fonctionne de la façon illustrée dans la figure (Voir figure 1.12).

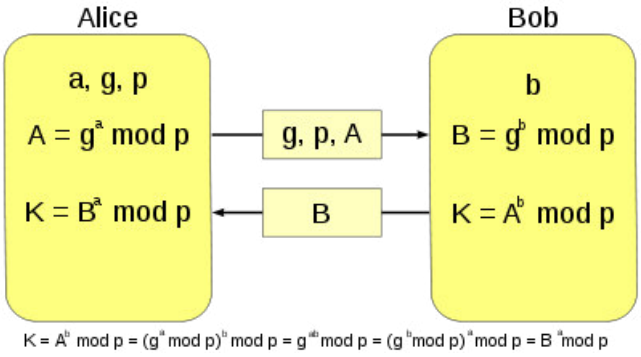


FIGURE 1.14 – Échange de clé Diffie-Hellman

Exemple :

Alice et Bob choisissent un nombre premier p et une base g :
Dans notre exemple : p = 23 et g = 3.
Alice choisit un nombre secret a= 6
Elle envoie à Bob la valeur $g^a \bmod p = 3^6 \bmod 23 = 16$.
Bob choisit à son tour un nombre secret b = 15
Bob envoie à Alice la valeur $g^b \bmod p = 3^{15} \bmod 23 = 12$.
Alice peut maintenant calculer la clé secrète :
$(g^b \bmod p)^a \bmod p = 126 \bmod 23 = 9$.
Bob fait de même et obtient la même clé qu'Alice :

$$(g^b \bmod p)^a \bmod p = 16^{15} \bmod 23 = 9 .$$

Alice et Bob tombent sur le résultat 9, qui est à présent leur clé secrète.

6. l'échange de clés Diffie-Hellman, du nom de ses auteurs Whitfield Diffie et Martin Hellman, est une méthode1, publiée en 1976, par laquelle deux agents, nommés par convention Alice et Bob,

1.8.2 L'Échange de clé El Gamal

L'algorithme de chiffrement basé sur le problème de logarithme discret est El-Gamal[7] qui à été propose par Tahar El-Gamal en 1985. Ce système repose sur le principe suivant :

1. Supposons qu'Alice souhaite envoyer à Bob un message chiffré. Pour cela : Alice et Bob doivent se mettre d'accord sur un groupe **G** sur lequel travailler, sur son générateur **g** et sur un nombre premier **p**.

2. Alice et Bob sélectionne respectivement une paire de clé, l'une privé a et b et l'autre publique $Q_a = g^a$ et $Q_b = g^b$.

3. Alice choisit aléatoirement un nombre **R** dans **G**, premier avec p - 1.

4.Pour chiffrer le message **M** Alice calcule $\mathbf{C = M(Q_a)^R mod\ p}$ **et** $\mathbf{k = g^R mod\ p}$ Le texte chiffre est alors (c ;k).

5. Pour déchiffrer le message M, Bob calcule $M = Ck^{-1}$[16].

Exemple :

Soient g = 2, h = 949, p = 2579.
- Clé privée : x=(765)
- Clé publique PK = (2, 949, 2579) car $2^{765} \text{mod} 2579 = 949$.

Pour chiffrer m = 1299, on choisit y = 853.
Il vient :

$C1 = 2^{853} \text{mod} 2579 = 435$
$C2 = 1299 * (949)^{853}\ \text{mod} 2579 = 2396$
On peut effectivement vérifier que $2396/\ 435^{765}\ \text{mod} 2579 = 1299$.

7. Taher Elgamal naît en 1955 en Égypte. Il obtient un bachelor of science de l'université du Caire en 1981, puis un master of science et enfin un doctorat de l'université Stanford. En 1985, il publie un article intitulé un cryptosystème à clef publique et un schéma de signature basé sur les logarithmes discrets connu désormais sous le nom d'algorithme El Gamal.

1.9 Conclusion

La cryptographie constitue la meilleure méthode de sécurisation de la confidentialité des données.Dans ce chapitre, on a présenté les différents types de chiffrement et on a donné un bref historique sur le développement de la cryptographie, ces types (***symétrique et asymétrique***), on conclut que pour chaque type il y a un problème d'échange des clés.

L'avantage majeur du cryptage symétrique est sa rapidité (quelques dixièmes de secondes pour un message d'un mégaoctet). Son principal inconvénient réside dans l'échange de la clé. Le cryptage asymétrique est beaucoup plus robuste. Cependant sa faiblesse est la lenteur du traitement des données due à des calculs compliqués.

Il existe une nouvelle technique assez récente d'échange de clés cryptographique appelée : « ***la cryptographie quantique*** » Dans le prochain chapitre on va présenter la technique de distribution de clé quantique (***QKD***) et les différents protocoles utilises.

CHAPITRE 2

ÉCHANGE DE CLÉS QUANTIQUE

2.1 Introduction

La cryptographie quantique ou la distribution de clé quantique (QKD) [1] est une technique relativement récente dans le monde de la sécurité de l'information. C'est une solution au problème majeur en cryptographie qui est le transfert de clé entre les deux parties communicantes. Il s'agit d'exploiter les lois de la mécanique quantique pour créer de nouvelles primitives cryptographiques.

Dans ce chapitre nous revenons sur le fondement de la QKD ainsi que les propriétés quantique du photon, preuve qu'il ne s'agit plus d'un simple concept théorique, mais aussi sur certains protocoles phares. Nous mettrons l'accent sur le protocole BB84 que nous étudions en détails, allant de l'idée de base du protocole jusqu'aux attaques contre ce dernier.

Une bref Historique

Depuis sa découverte en 1927, le principe d'incertitude de Heisenberg a souvent été perçu comme une limitation fondamentale imposée par la mécanique quantique aux mesures physiques. La cryptographie quantique exploite cette « faiblesse » qui est fait une force, permettant de garantir un secret absolu sur des communications cryptées. Fondée sur une idée originale de S. Wiesner, refusée en 1969 par une revue scientifique [17]. Wiesner, dans son papier, a premièrement proposé en 1970 la CQ [2] qui n'avait pas été publié jusqu'en 1983 , ou plus précisément Distribution de Clé Quantique, propose une méthode alternative aux méthodes mathématiques. Elle se base sur les lois de la physique et vise à assurer la sécurité inconditionnelle de l'échange de clé. Elle a ouvert devant les scientifiques un nouvel axe de recherche pour résoudre le problème courant de la cryptographie.
La cryptographie quantique s'est développée à partir de la publication par C.H. Bennett et G. Brassard, en 1984 [3], d'un protocole d'échange quantique de clés. C'est aujourd'hui un domaine pluridisciplinaire en pleine expansion, a la veille d'applications commerciales et militaires [17].

1. Quantum key distribution/Distribution de clés quantiques
2. Cryptographie quantique
3. le protocole BB84 est le premier mécanisme d'échange de clé quantique à avoir été formalisé, et en fait le premier protocole de cryptographie quantique. Il a été proposé en 1984 par Charles Bennett et Gilles Brassard

2.2 Le fondement de cryptographie quantique

2.2.1 Le Qubit

Le ***bit*** est le concept fondamental de l'information classique. Il s'agit d'un chiffre binaire pouvant prendre la valeur **0** ou **1**. Une succession de bits constitue un message. En information quantique,les bits quantiques (***qubits***) sont utilisés[23].Il s'agit de la plus petite unité de stockage de l'information quantique. Un qubit est une superposition linéaire de deux états de base notes, suivant la notation de Dirac, ***|0> et |1>.***

Un qubit |ψ > s'écrit :

$$|\psi> = \alpha\,|0> + \beta\,|1>$$

ou α et β «l'amplitude de probabilité» sont des nombres complexes qui vérifient la condition de normalisation du qubit :

$$|\,\alpha\,|^2 + |\,\beta\,|^2 = 1$$

- $|\alpha|^2$: représente la probabilité d'avoir le bit 0.
- $|\beta|^2$: représente la probabilité d'avoir le bit 1.

1 Et 0 : représente deux états orthogonaux dans le système quantique.

Comme les coefficients α et β sont complexes, il est possible de représenter un qubit par un point sur une sphère de rayon 1, appelée sphère de Bloch [4] [18] (voir figure 2.1).

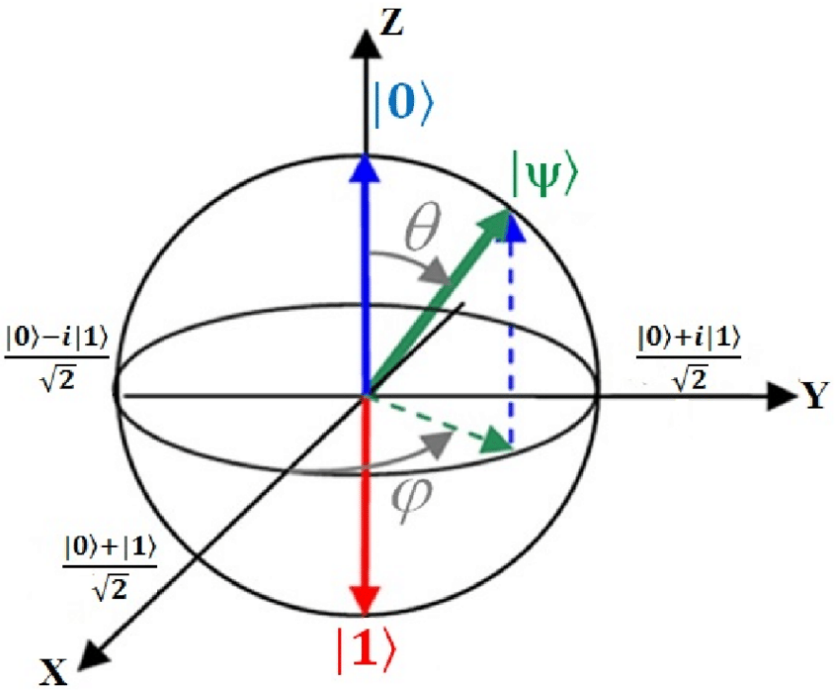


FIGURE 2.1 – Un qubit est un point sur la sphère de Bloch

4. La sphère de Bloch, du nom du physicien et mathématicien Félix Bloch, ou sphère de Poincaré (comme cas d'application de celle-ci), est une représentation géométrique d'un état pur d'un système quantique à deux niveaux ; c'est donc, entre autres, une représentation d'un qubit.

Une autre particularité du qubit par rapport à un bit classique est qu'il ne peut être dupliqué. En effet, pour le dupliquer, il faudrait pouvoir mesurer α et β d'une qubit (tout en préservant l'état du qubit), de sorte à préparer un autre qubit dans le même état **α |0>** + **β |1>** .

Ceci est doublement impossible :

- Il est impossible de lire un qubit sans détruire définitivement son état (puisque après mesure le qubit est dans l'état mesurée) .

- Une mesure d'un qubit ne donne aucune information sur **α et β** puisque le résultat est soit **|0>** soit **|1>** ce qui équivaut à **(α , β)** = (1,0) ou (0,1) ce qui ne correspond pas aux valeurs initiales de α et β.

2.2.2 L'intrication quantique (phénomène purement quantique)

L'intrication quantique, ou enchevêtrement quantique, est un phénomène dans lequel deux particules (ou groupes de particules) forment un système lié et présentent des états quantiques dépendant l'un de l'autre quelle que soit la distance qui les sépare. Un tel état est dit « intriqué » car il y a des corrélations entre les propriétés physiques observées de ces particules distinctes. Ainsi, deux objets intriqués $\Theta_1$et Θ_2. ne sont pas indépendants même séparés par une grande distance, et il faut considérer $\{\Theta_1 + \Theta_2\}$comme un système unique (Voir figure 2.2.)[1] .

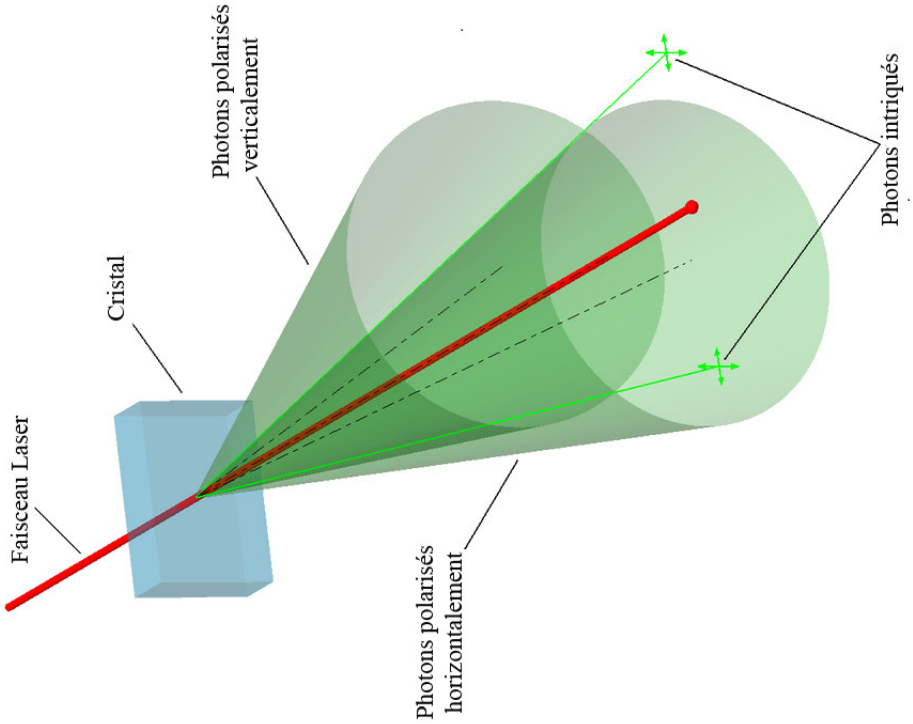


FIGURE 2.2 – Vue d'artiste d'un système de génération de deux photons intriqués

2.2.3 La superposition quantique

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 : $\alpha\ |0> + \beta\ |1>$. Les coefficients α et β étant deux nombres complexes vérifiant la relation $|\alpha|^2+|\beta|^2=1$[1].

En particulier, le principe de superposition est à l'origine de ce qu'on appelle le problème de la mesure quantique, La physique quantique répond à cela en le plaçant dans des états superposés : le chat de Schrödinger [5] (Voir figure 2.3) est à la fois mort et vivant .

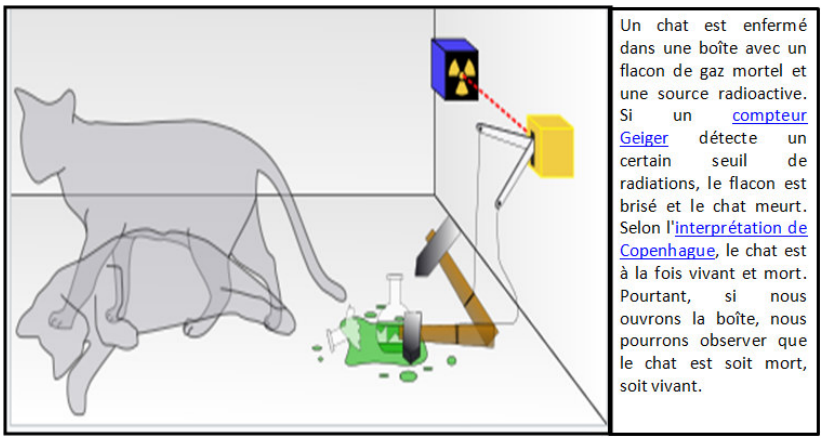


FIGURE 2.3 – La physique quantique dans l'affaire du Chat de Schrödinger

2.2.4 Le principe d'incertitude de Heisenberg

Au cours des années 1920, les physiciens ont découvert qu'il était fondamentalement impossible de tout savoir sur les propriétés physiques d'un objet microscopique. Chaque grandeur physique mesurable ***(appelée « observable »***), par exemple la position d'un objet, est intrinsèquement reliée à une autre, par exemple la vitesse, de sorte qu'on ne puisse jamais mesurer simultanément deux observables complémentaires avec une précision arbitraire. Ce « ***principe d'incertitude*** » nous dit que le prix à payer pour mesurer très précisément une observable est alors de détruire complètement l'information sur l'autre. La mesure simultanée des deux observables est toutefois possible, mais limitée à une précision « moyenne ». Bien sûr, cet effet est infinitésimal pour des objets de taille macroscopique, mais est essentiel aux petites échelles (***photons uniques, atomes, électrons, etc***.). En d'autres termes, ce « ***principe d'incertitude de Heisenberg*** » limite la quantité d'information disponible sur les propriétés physiques de ces objets. De plus, il nous dit qu'en général une mesure perturbe le système, ce qui limite la précision des mesures ultérieures.

Ce principe, bien ancré dans les principes fondamentaux de la mécanique quantique, est une loi de la physique et non une limitation technologique : c'est ce qui lui donne sa pertinence dans les systèmes de cryptographie quantique[19].

5. Le chat de Schrödinger est une expérience de pensée imaginée en 1935 par le physicien Erwin Schrödinger afin de mettre en évidence des lacunes supposées de l'interprétation de Copenhague de la physique quantique, et particulièrement mettre en évidence le problème de la mesure.

2.2.5 Le non-clonage

Une autre particularité du qubit par rapport à un bit classique est qu'il ne peut être dupliqué. En effet, pour le dupliquer, il faudrait pouvoir mesurer les amplitudes α et β du qubit unique initial, tout en préservant son état, de sorte à préparer un autre qubit dans le même état $\alpha\ |0> + \beta\ |1>$. Ceci est doublement impossible en raison du théorème de « non-clonage »[18].

2.3 Propriétés quantiques du photon

Le problème majeur en cryptographie symétrie 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 pré occupe ici consistera en l'envoi de photons. On utilisera la technique dite de « ***Polarisation de photons*** »[6].

2.3.1 Polarisation de photons

Chaque photon peut être polarisé, c'est-à-dire qu'il est possible d'imposer une direction à son champ électrique. Cette polarisation des photons est réalisée par un angle variant de 0°'a180°. Par simplification des modelés théoriques, on considéré souvent 4 angles précis : 0°, 45°, 90°et 135°(Voir figure 2.4).

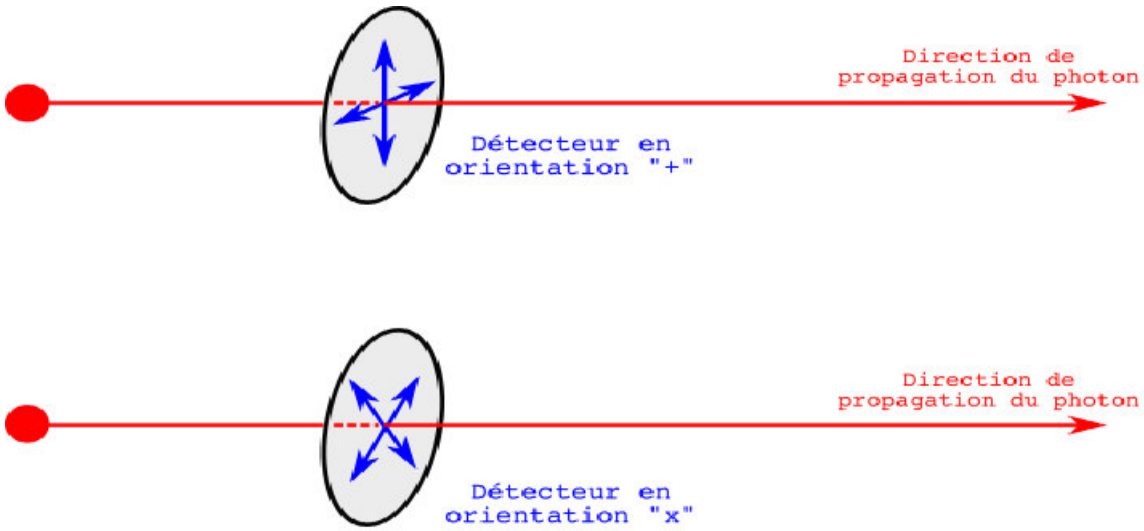


FIGURE 2.4 – polarisation des photons

On parlera de polarisation rectiligne pour les angles de 0° et 90° (***pour lesquelles on utilisera des filtres ou bases "standards"***), et de polarisation diagonale pour les angles de 45° et 135°(***filtres ou bases "diagonales")***. Pour la détection, on aura recours à un filtre (ou base) polarisant(e) et un détecteur de photons. Ce type de filtre permet la lecture de photons polarisés d'une certaine manière. En conséquence, il bloque les photons polarisés dans la direction perpendiculaire. le filtre est positionné de telle sorte qu'il laisse passer les photons polarisés à 0° et par conséquent, bloque les photons polarisés à 90°. On est donc ici en présence d'un filtre standard [6] (Voir figure 2.4).

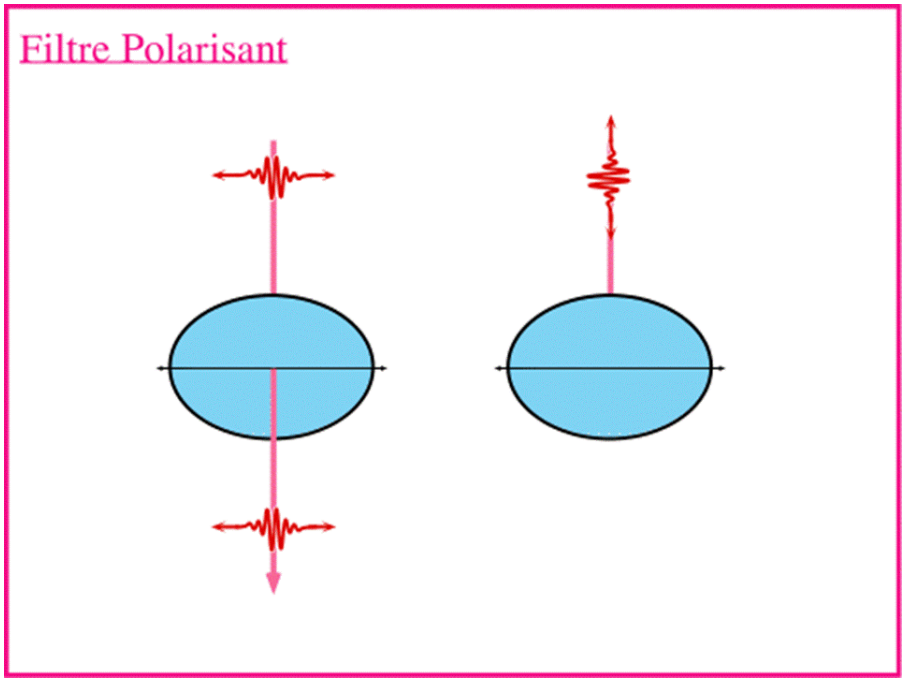


FIGURE 2.5 – Polarisation des photons

2.3.2 Photodétecteur

A la suite de ce filtre, il est nécessaire d'utiliser un outil permettant de détecter ces photons. On parle de photodétecteurs. Soit un ***angle*** ϑ pour le photon à détecter.

Après un passage dans un filtre standard, le photo-détecteur placé à sa suite détectera le photon avec une probabilité de $cos^2\ \vartheta$ et de ne pas le détecter avec une probabilité de $sin^2\ \vartheta$.

Pour un filtre d'angle quelconque, le raisonnement est semblable. Soit a l'angle du photon et b l'angle du filtre. On obtient une transmission avec une probabilité de $cos^2\ (a - b)$ et non transmission avec une probabilité de $sins^2\ (a - b)$.Ceci est illustré dans la figure 2.*5.*

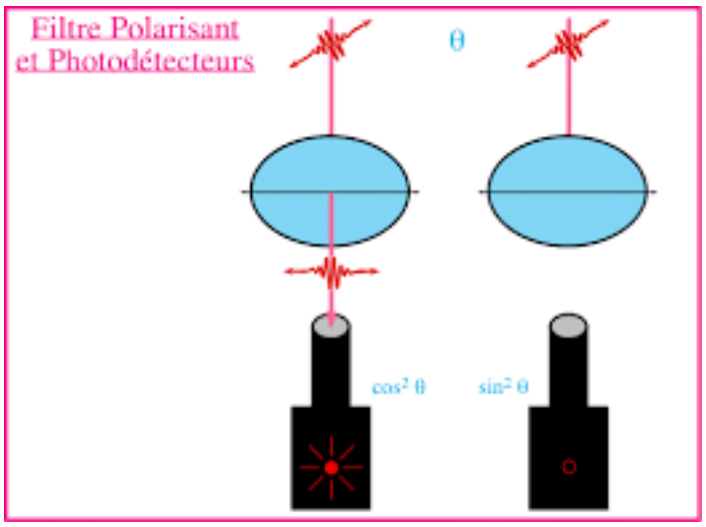


FIGURE 2.6 – filtre polarisant et photodetecteurs

Par convention, on considéré qu'un photon polarisé à 0° ou 45° représentent un 0 et un photon polarisé 'a 90° ou 135° représente un 1. Le problème intervient lorsqu'un photon polarisé diagonalement (45° ou 135°) rencontre un filtre standard (et inversement). Il y a alors incertitude, comme il est montré sur la figure suivant [21] :

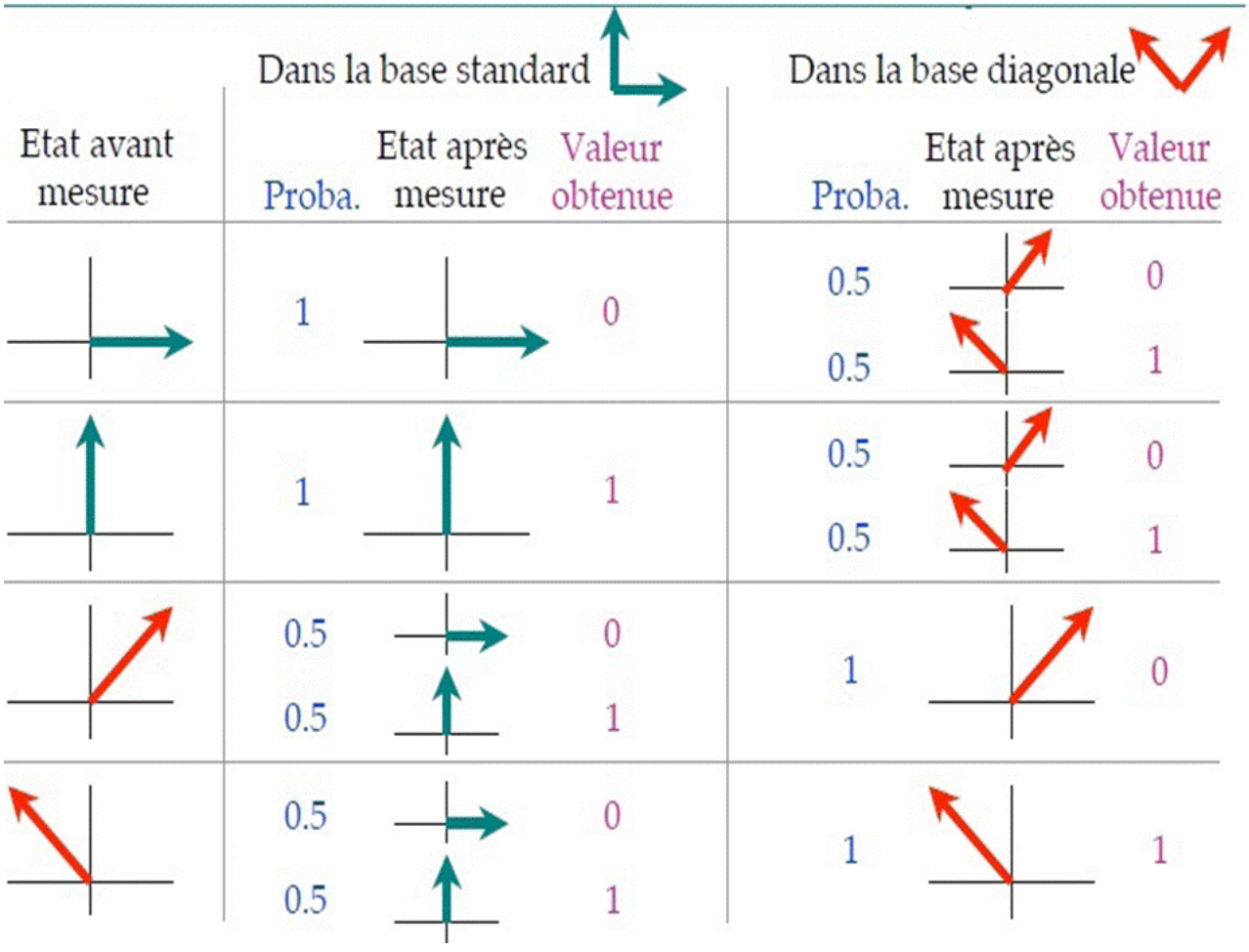


FIGURE 2.7 – Détection sure et incertitude

2.4 Distribution des clés Quantiques (QKD)

La distribution quantique des clés (***QKD***)[6], fournit une méthode de codage parfaitement sécurisée qui résout le problème de la distribution des clés, elle est actuellement l'application la plus mature dans le domaine de l'informatique quantique [1] .

La distribution de clé quantique (***QKD***) exploite les principes fondamentaux de la mécanique quantique. Le premier de ces principes est le principe d'incertitude de Heisenberg, Le deuxième principe est le théorème de non-clonage [1] .

Ces deux principes forment la base de tous les protocoles de distribution de clés quantiques et sont la clé de la contribution majeure de ces protocoles, qui est la capacité de détecter toute écoute indiscrète sur le canal. Le modèle fondamental des protocoles ***QKD*** implique deux parties, appelées Alice et Bob, souhaitant échanger une clé à la fois avec l'accès à un canal de communication publique classique et un canal de communication quantique. Ceci est illustré dans la figure 2.7. Une oreille indiscrète, appelée Eve, est supposée avoir accès aux deux canaux et aucune hypothèse n'est faite sur les ressources à sa disposition [22].

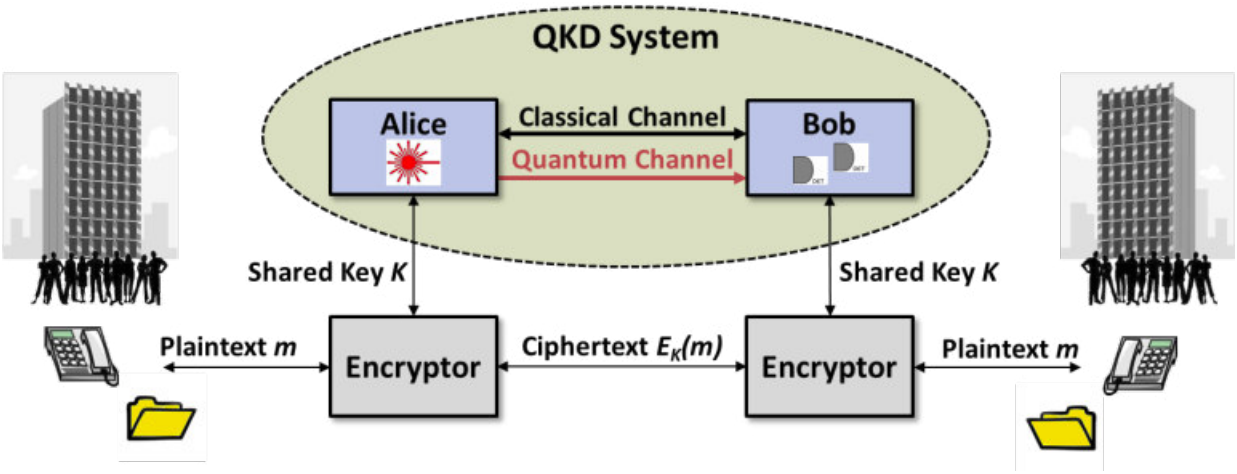


FIGURE 2.8 – Modèle de communications de distribution de clés quantiques

2.5 Quelques protocoles de distribution de clé quantique

Plusieurs protocoles sont proposés afin de discipliner la cryptographie quantique. Nous récapitulons certains d'entre eux dans la suite de cette section.

2.5.1 Protocole BB84

BB84[7]est un protocole de distribution de clé le plus connu et développé par Charles Bennett et Gilles Brassard en 1984 [23]. Il utilise quatre états différents qui font une paire des états de base.

BB84 est un protocole non déterministe. Cela signifie qu'il distribue une suite aléatoire des bits. Le but est de générer une clé partagée entre Alice et Bob n'autorisant aucun tiers à acquérir une information pertinente sur cette clé. Cette clé doit pouvoir servir à un chiffre

6. Quantum key distribution/Distribution de clés quantiques
7. Charles Bennett et Gilles Brassard en 1984

de Vernam, et conduire ainsi a‘ une transmission d'informations inconditionnellement sures.

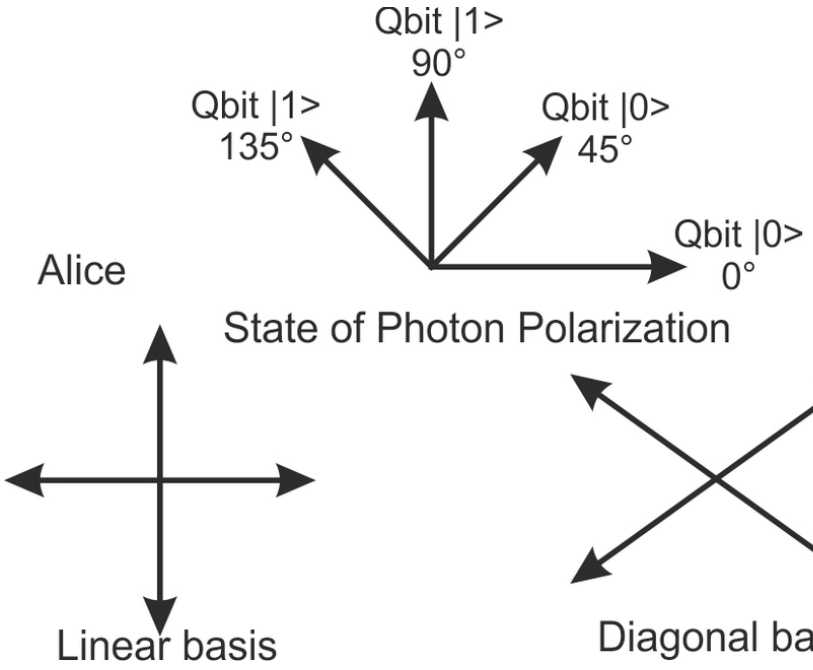
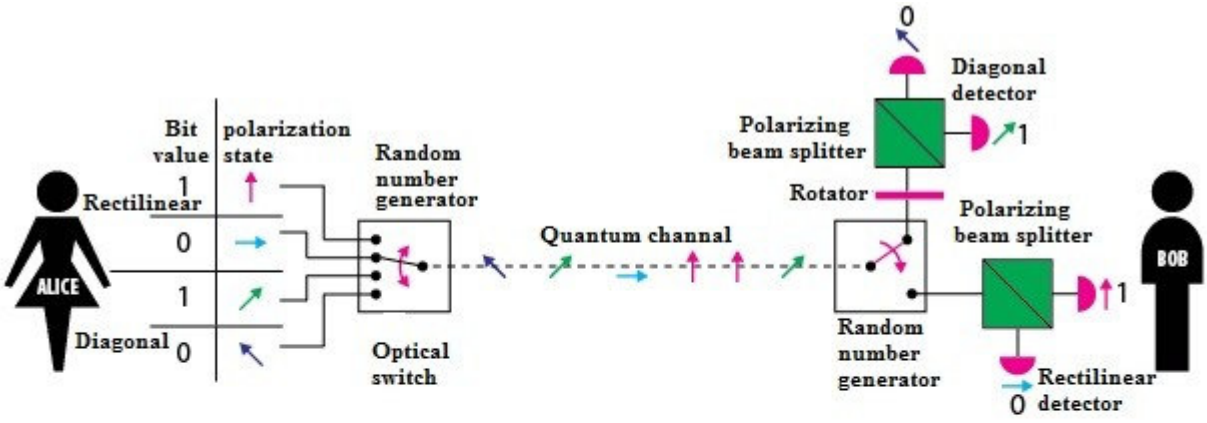


FIGURE 2.9 – Les quatre états non orthogonaux utilisés dans le protocole BB84

En général, l'échange de clé de quantum employant BB84 pour la clé secrète se compose de cinq étapes suivantes :



Quantum transmission & detection	ALICE sends photons	↖	↗	→	↑	↑	↗	↖	↑
	ALICE random bits	0	1	0	1	1	1	0	1
	BOB's detection events	↑	↗	↖	↑	↗	↗	↖	↖
	BOB's detected values	1	1	0	1	1	1	0	0
Public discussion (i.e.sifting	BOB tells ALICE the basis choices he mode	Rect	Diag	Diag	Rect	Diag	Diag	Diag	Diag
	ALICE tells BOB which bits to keep	✔		✔		✔	✔		
	ALICE and BOB's shared sifted key	1	–	1	–	1	0	–	

FIGURE 2.10 – l'échange de clé de quantum employant BB84 pour la clé secrète

Transmission des Qubits

Cette phase est la première étape dans une distribution de clé quantique. 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 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 clé 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 des qubits [23] (Voir figure 2.11).

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 clé 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 [24] (Voir figure 2.11).

Estimation du taux d'erreur quantique (QBER)

Afin de réduire la différence de la clé tamisé entre Alice et de Bob due a‘ l'imperfection d'appareil ou bien à une éventuelle intrusion, il est nécessaire de corriger les erreurs.C'est la phase ou‘ l'erreur de la clé tamisée est estimée. Ils choisissent alors de se révéler une fraction de la clé tamisé ($\sim 10\%$) , ceci leur permet d'estimer le QBER de la transmission. L'émetteur et le récepteur doivent calculer le taux d'erreurs observées et gardent cette transmission si le taux d'erreur est inférieur à un seuil désiré, sinon la clé sera avortée[24] .(Voir figure 2.11).

Réconciliation

Après cette phase, une clé réconciliée sera obtenue après application d'un protocole de réconciliation à la clé 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 clés 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 [24] (Voir figures 2.10 et 2.11).

Purification (Amplification de la sécurité)

A la fin de la communication, Alice et Bob ont la mémé clé réconciliée, ils vont essayer de réduire l'information que possède Ève.Alors ils utilisent un algorithme dit « Purifica-

tion du sécurité». Pour ce faire, Alice choisit à nouveau des paires de bits dont elle prend leur somme XOR, mais cette fois-ci, elle annonce seulement le numéro des bits. Alice et Bob remplacent simplement la valeur de chacun de ces deux bits par la valeur de leur somme XOR. Ainsi, Alice et Bob n'engendrent pas de nouvelles différences entre leur clé et déduisent l'information d'Ève au détriment bien sur de la longueur de leur clé. En effet, si Eve ne connait que la valeur du premier bit mais pas du deuxième, elle n'a aucune information sur leur somme XOR. Finalement Alice et Bob disposent d'une clé secrété et sans erreur à propos de laquelle Eve n'a aucune information.[25](Voir figures 2.10 et 2.11).

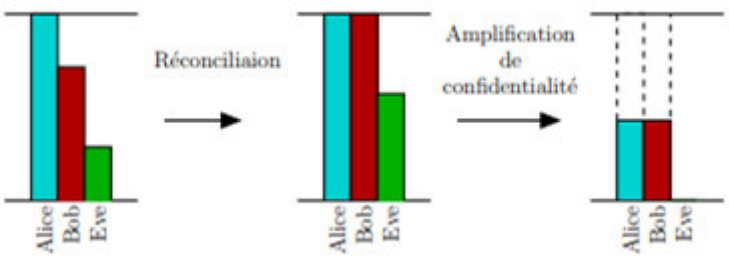


FIGURE 2.11 – Évolution de l'information d'Alice, Bob et Ève au cours d'une distribution quantique de clé

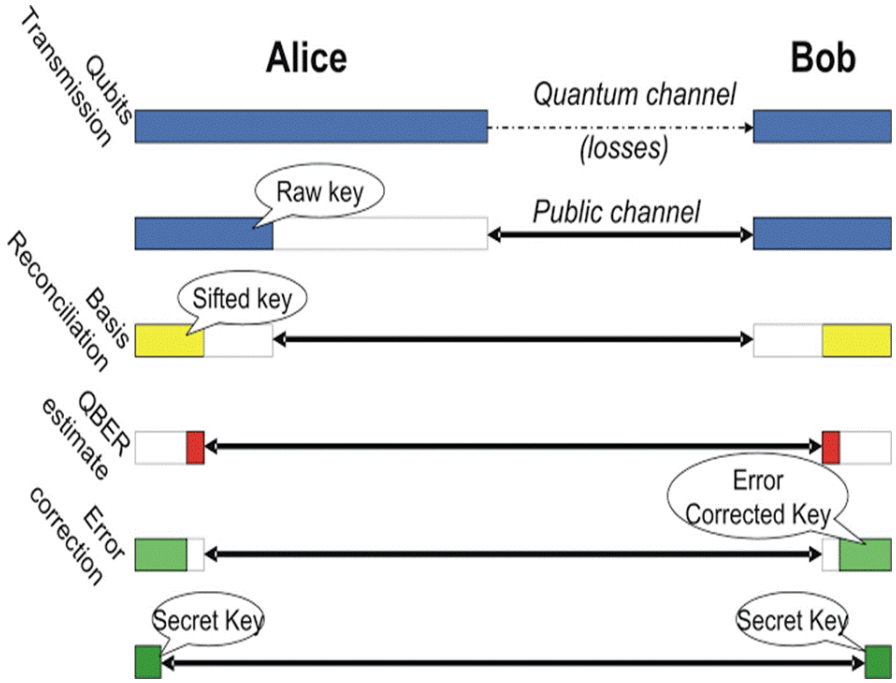


FIGURE 2.12 – Évolution de la taille de la clé quantique

2.5.2 Protocole Ekert91

Le protocole Ekert utilise des paires de photons intriqués . Ceux-ci peuvent être créés par le client A, par le client B, ou par une source distincte de chacun d'eux, y compris Eve.

Les photons sont distribués de sorte que le client A et le client B reçoivent chacun un photon de chaque paire. Le protocole repose sur deux propriétés d'intrication. Première-

ment, les états intriqués sont parfaitement corrélés dans le sens où si le Client A et le Client B mesurent tous deux si leurs particules ont des polarisations verticales ou horizontales, ils obtiendront toujours la même réponse avec une probabilité de 100%. Il en est de même si tous les deux mesurent une autre paire de polarisations complémentaires (orthogonales).

Cependant, les résultats particuliers étant complètement aléatoires, il est impossible au client A de prédire si le client B obtiendra une polarisation verticale ou une polarisation horizontale. Deuxièmement, toute tentative d'écoute par Eve détruira ces corrélations d'une manière que le client A et le client B peuvent détecter sa présence [26].

2.5.3 Protocole B92

En 1992, Charles Bennett a proposé le protocole B92 dans son article "***Cryptographie quantique utilisant deux états non orthogonaux"***. Le protocole B92 est une version modifiée du protocole ***BB84***, la principale différence entre les deux étant que, tandis que le protocole ***BB84*** utilise quatre états de polarisation différents du photon, le protocole B92 en utilise deux (***un de la base rectiligne, conventionnellement un état de polarisation H et un de la base diagonale, conventionnellement + 45 ° - état de polarisation***). Le protocole B92 peut être résumé dans les étapes suivantes :

1. Alice envoie une chaine de photons en état de polarisation H ou en état de polarisation + 45 °, choisis au hasard. L'état H correspondra au bit "0" tandis que l'état + 45 ° correspondra au bit "1"[27].

2. Bob choisit au hasard entre une base rectiligne et diagonale, pour mesurer la polarisation du photon reçu. Si Bob mesure sur la base rectiligne, il y a deux circonstances possibles - Si le photon incident est polarisé H, alors le résultat de la mesure sera l'état H avec la probabilité 1 alors que si le photon incident est polarisé à + 45 °, alors le le résultat de la mesure sera soit l'état H soit l'état V avec une probabilité de 0,5. Ainsi, si seul le résultat est un état V, Bob peut inférer avec confiance que l'état de polarisation incident du photon est '+ 45 °'[27].
3. Un argument similaire sera applicable si Bob mesure en diagonale, où le résultat de mesure de -45 ° indique que l'état de polarisation incident du photon est «H». Après la transmission de la chaîne de photons, Bob annonce les cas dans lesquels le résultat de la mesure était soit «V» soit «-45 °» et le reste est rejeté par les deux[27].

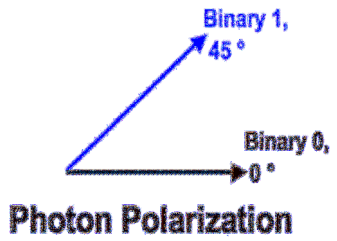


FIGURE 2.13 – Encodage à deux états.

- Ces résultats peuvent être utilisés pour générer une Amplification haine de bits aléatoire entre ***Alice*** et ***Bob***. Pour la vérification de l'écoute, ***Bob*** et Alice partagent publiquement une partie de la chaine de bits aléatoires générée et si l'erreur dépasse une limite tolérable, le protocole est abandonné. Sinon, ils ont maintenant pu générer une clé sécurisée et symétrique entre eux. Un autre contraste important avec le protocole ***BB84*** est que ***Bob*** n'a pas à annoncer le choix de la base dans la communication après transmission, c'est-à-dire que le tamisage n'est pas requis[27].

- Le tableau suivant montre différents cas possibles de transmission et de résultats de mesure :

Valeur Alice	1	0	1	1	0	0	0	1
Polarisation Alice	↗	↑	↗	↗	↑	↑	↑	↗
Valeur Bob	0	1	1	1	0	0	1	0
Polarisation Bob	↖	→	→	→	↖	↖	→	↖
Décision	non	non	non		non		non	non

TABLE 2.1 – Démonstration expérimentale du protocole B92

2.5.4 Autres protocoles :

Il est certain que le protocole de distribution de clé quantique le plus connu est le ***BB84***. Il existe de nombreux autres protocoles, Voilà un tableau dont tous les protocoles ***QKD*** concernés sont observés et en quelle année respectivement [1] :

Année	**Nom du protocole**	**Principe**
1984	BB84	Principe d'incertitude de Heisenberg
1991	E91	Intrication quantique
1992	B92	Principe d'incertitude de Heisenberg
1999	SSP	Principe d'incertitude de Heisenberg
2002	GG02	Principe d'incertitude de Heisenberg
2003	DPS	Intrication quantique
2004	SARG04	Principe d'incertitude de Heisenberg
2004	COW	Intrication quantique
2009	KMB09	Principe d'incertitude de Heisenberg
2012	S09	Cryptographie à clé publique
2013	S13	Principe d'incertitude de Heisenberg

TABLE 2.2 – Tableau des protocoles QKD

2.6 Quelques types d'attaques d'Ève contre BB84

Pendant la communication entre Alice et Bob, Ève peut essayer d'écouter le canal de quantum et aussi celui classique. On suppose qu'elle puisse prendre facilement tout ce qui voyage au travers le canal vulnérable (celui classique). Et pour le canal de quantum, Ève applique quelques stratégies typiques d'attaque suivantes à la fouille en tant que le maximum d'information possible [22].

Intercept-Resend (Attaque interception-renvoi)

Grâce au principe d'incertitude d'Heisenberg qui s'agit que un seul quanta ne peut pas être dédoublé et son état ne peut pas être copié, Eve n'a pas la capacité de surveiller passivement les transmissions. Alors pour obtenir de l'information sur la clé secrète qu'Alice et Bob tentent d'échanger, Ève doit intercepter les photons envoyés par Alice, effectuer des mesures sur ces photons et les renvoyer à Bob. Cette attaque va introduire des incohérences dans les données d'Alice et Bob.

Beam-splitting (Attaque par séparation du nombre de photons)

Deuxième attaque fréquente est beam-splitting. Ève tire profit de l'imperfection du système pour extraire l'information. C'est dû aux impulsions produisant non seulement d'un photon mais deux ou de plus. Et grâce à ces photons en excès, Ève emploie la forme de miroir à moitié-argent pour la décomposition du faisceau des photons. Elle garde alors un ou deux photons pour mesurer et laisse les autres voyager à Bob. Dans ce cas, il est difficile à détecter la présence d'Ève. Car décomposer quelques photons d'un faisceau de photons n'affecte pas la polarisation de lui. Néanmoins, Alice et Bob peuvent prendre un pré-compromis sur le retard du photon pour découvrir l'apparition d'Ève [22].

Homme-au-milieu :

Une imperfection évidente de BB84 est le manque d'authentification. En plus, avec la technologie de niveau élevé, Ève peut penser à une attaque appelée homme-au-milieu ou attaque intermédiaire, dans laquelle Eve devient un truqueur. Elle intercepte le canal sécurisé (celui de quantum) et joue comme Bob avec Alice et inversement. En faisant ainsi, elle peut recueillir toute l'information échangée entre Alice et Bob sans leur doute. Donc, à la fin de casser ce genre d'attaque, l'authentification est le souci le plus grand pour le protocole ***BB84*** [22].

2.7 Avantages et inconvénients de la cryptographie quantique

La cryptographie quantique présente un avantage fondamental par rapport à sa contrepartie classique :

- lorsqu'elle est correctement mise en œuvre, elle est inviolable par principe, quelles que soient les capacités technologiques de l'adversaire.

- Un message chiffré aujourd'hui avec une clé transmise par cryptographie quantique ne pourra jamais être décrypté sans que la clé soit rendue publique.

- Les systèmes de cryptographie classiques sont au contraire perpétuellement menacés d'obsolescence, et la protection qu'ils fournissent ne peut guère être évaluée au-delà de quelques dizaines d'années.

Le revers de cette inviolabilité se retrouve sur les contraintes imposées sur le support physique constituant le canal quantique :

- les pertes et les fluctuations de polarisation sont les problèmes majeurs des transmissions par fibres optiques, dont la portée maximale pour la cryptographie quantique est aujourd'hui de l'ordre de 100 km ; des distances supérieures ont été atteintes, mais avec des débits très faibles.

- Les systèmes de transmission à l'air libre sont eux limités par des difficultés de visée et de lumière parasite.

2.8 Conclusion

La cryptographie quantique exploite la loi de la mécanique quantique pour la sécurisation de l'information. C'est une distribution quantique des clés de cryptage entre un émetteur et un récepteur communiquant sur la fibre optique.

C'est le premier concept quantique fondamental en cours de faire la transition de la recherche purement scientifique vers une application industrielle. Ainsi, il est souhaitable de rendre les systèmes plus stables et plus faciles à utiliser pour certains utilisateurs finaux intéressés par une communication sécurisée.

Dans le prochain chapitre, nous allons implémenter et simuler les bases de la cryptographie quantique, la simulation du protocole BB84 et le type d'attaque utilisé **« intersept-resend »** (**Interception - renvoi »** . Au final, nous analysons les résultats et interprétations .

CHAPITRE 3

ANALYSE ET SIMULATION DU PROTOCOLE DU BB84 SOUS MATLAB

3.1 Introduction

Après avoir étudié le fondement et les principes de fonctionnement de la cryptographie quantique, dans ce chapitre nous étudions l'aspect probabiliste du premier protocole de distribution quantique de clé (***QKD***) proposé par Bennett et Brassard en 1984 (***BB84***).

Ce protocole permet à deux personnes séparées de construire une clé secrète dont ils seront les seuls à connaitre, par l'envoi de l'un vers l'autre de photons unique qui proviennent d'une source de lumière cohérente, à traverse un canal de transmission quantique (***fibre optique, espace libre***) et d'un canal publique (***radio, internet***) .

N.B : Le protocole ***BB84*** ou Charles Bennett et Gilles Brassard 1984 est l'un des protocoles utilisant les propriétés de la mécanique quantique à travers le code de Vernam le plus utilisés **.**

La méthode que nous suggérons dans cette humble recherche est d'échanger une clé quantique en utilisant la simulation de protocole BB84 pour obtenir une clé binaire pour effectuer une opération ***XOR (ou exclusive)*** entre la clé et les messages pour chiffrement et Déchiffrement par le chiffre de (***Vernam (One-Time-Pad***) .On peut résumer les étapes de simulation dans la figure 3.1.

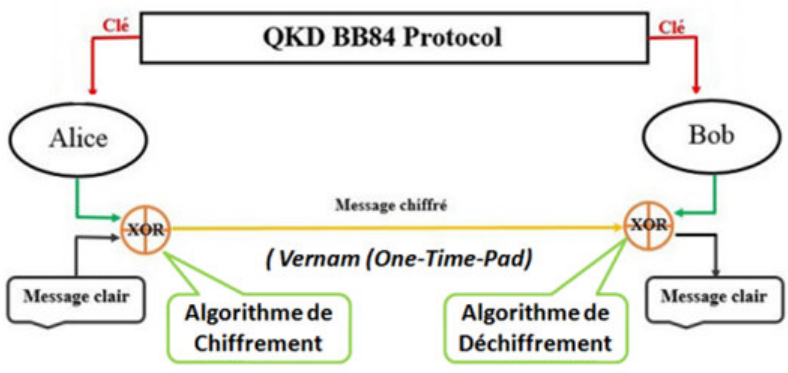


FIGURE 3.1 – Schéma synoptique du protocole BB84

3.2 Étude et simulation du protocole BB84

Le protocole BB84 nécessite deux parties : *Alice*, l'expéditeur, et *Bob*, le récepteur. Alice prépare une séquence binaires aléatoires, nous supposons qu'elle prépare les qubits en "codant" des photons uniques avec quatre états de polarisation orthogonaux. Deux bases sont alors utilisées, la base *verticale/horizontale* et la base *diagonale/antidiagonale* (Tableau 3.1) :

Bob prépare une séquence de base aléatoire et mesure les états des qubits, en utilisant le résultat de la mesure, il obtient un message binaire brut. Alice et Bob comparent ensuite via le canal publique les bases choisit préalablement et rejettent les bits qui ne correspondent pas [28].

Ce protocole utilise la transmission de photons uniques polarisés (quatre polarisations) regroupées en deux bases non orthogonales différentes. En général, les deux bases non orthogonales sont :

- La base $\oplus$: les valeurs binaires '0' et '1' correspondent respectivement aux photons portant des polarisations à 0° et 90°.
- La base $\otimes$: les valeurs binaires '0' et '1' correspondent respectivement aux photons portant des polarisations à 45° et 135°.

Elle envoie ensuite les qubits préparés à Bob à travers un canal quantique.

Codage quantique du protocole BB84

Bit	La base B_+(la base standard)			La base B_X(la base diagonale)		
0	Qubit	L'état de photon	l'angle	Qubit	L'état de photon	l'angle
	$\|0_+\rangle = \|0\rangle$	$\|\rightarrow\rangle$	0°	$\|0_x\rangle = \frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$	$\|\nearrow\rangle$	45°
1	$\|1_+\rangle = \|1\rangle$	$\|\uparrow\rangle$	90°	$\|1_x\rangle = \frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$	$\|\nwarrow\rangle$	135°

TABLE 3.1 – Tableau du codage d'un bit en fonction du choix d'une base

3.3 Les hypothèses

— On suppose que le canal est parfait**.**
— La source (Alice) peut envoyer des photons unique i.e. les dispositifs d'envoi et de réceptions de la part du Bob est parfait (**Modèle de simulation** (Voir figure 3.2)).
— Le type d'attaque appliqué par Ève est ***« intersept - resend »*** (**Interception - Renvoi**) .

« Mais la question qui se pose dans notre projet de fin d'études, comment peut-on détecter la présence d'un espion ? ».

3.3.1 Définition Le type d'attaque utilisé « intersept-resend (I-R) »

Ce type d'attaque correspond à celles qui sont les plus immédiates à mettre en œuvre, elle consiste pour Ève, à mesurer individuellement les impulsions lumineuse émises par Alice, puis à envoyer vers Bob un photon codé dans l'état correspondant au résultat de mesure qu'elle a obtenu.
Si Ève a choisi la même base qu'Alice, elle ne sera pas détectée car l'état de polarisation du photon ne sera pas perturbé et Bob mesurera la polarisation avec une probabilité d'erreur égale à 0.

Par contre, si Ève choisira une base différente, elle aura une chance sur deux de se tromper. Quand Bob reçoit le photon envoyé par Ève, il a une chance sur deux d'avoir un résultat différent avec le photon original,par conséquent, pour chaque photon intercepté par Ève, il y a une chance sur quatre que Bob reçoit une information erronée. Ainsi, dans le cas ou Ève intercepte toutes les photons, la probabilité d'erreur induite sera égale à 25%[29].

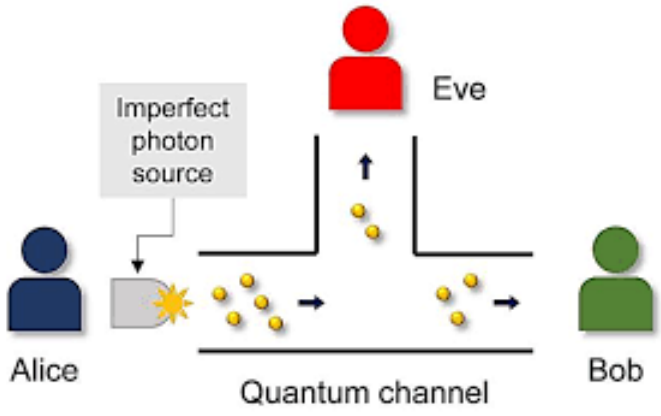


FIGURE 3.2 – Type d'attaque appliquée par Ève (intersept-resend).

3.4 Les étapes de simulation

3.4.1 Transmissions quantiques (1^{er} étape)

1. Envoi d'encodages quantiques de bits de clé .

— Alice génère une clé aléatoire k = k_1,...,k_n ; avec chaque k_i ∈{0,1}.
— Alice prépare un photon à un état quantique P ∈ {⟶, ↕, ↗, ↖} pour chaque bit k_i il génère des bases (ou manières d'encoder) aléatoires b = b_1,..,b_n ; avec chaque b_i ∈{ ⊗,⊕}.
— Alice envoie à Bob tous les bits k_i avec les encodages b_i correspondants.

2. Bob mesure chaque photon Bob P ∈ {⟶, ↕, ↗, ↖}, il choisit des encodages ***b' (*** b' = b'$_1$,...,b'_n ; avec chaque b'_i ∈ { ⊗,⊕ }***) aléatoires et va essayer de les récupérer malgré tout. Il indique à Alice qu'il a reçu tous les qubits*** [26].

3.4.2 Discussion publique (2^{ieme} étape)

1. Pour chaque bit k_i en k :
- Bob sur le canal classique envoie la valeur de k_i à Bob.
- Alice répond à Alice en déclarant s'il a utilisé la même base pour les mesures. Alice et Bob rejettent b_i et b'_i si $b_i \neq b'_i$.

2. Alice choisit un sous-ensemble aléatoire des bits restants en d environ ($\sim 10\%$) et révélé leurs valeurs à Bob sur le canal classique (sur internet par exemple). Si le résultat des mesures de Bob pour l'un de ces bits ne correspond pas aux valeurs divulguées, l'écoute clandestine est détectée et la communication est interrompue.

3. la clé secrète commune k = {0, 1} (la clé finale) est la séquence de bits restante en d'après que les bits divulgués à l'étape précédente sont retirés.

Pour détecter Eve, Alice et Bob effectuent un test (2^{ieme} étape du protocole), l'idée est que, si les bases de Bob et Alice sont identiques c.à.d ($b_i \neq b'_i$) doivent correspondre a' des bits identiques ($b_i = b'_i$), sinon une perturbation est détectée et on suppose que cette perturbation est due à un espion[26].

Que peut faire l'espion ?

Pour chaque qubit qui passe, l'espion peut :

1. *essayer récupérer le bit de clé pour un encodage donné.*
   • si l'encodage choisi est le même que celui utilisé par Alice : OK.
   • sinon : le qubit est détruit et Ève obtient une valeur aléatoire. On va faire en sorte que cela soit détecté.

2. laisser passer le qubit.

L'espion ne peut pas copier le qubit, garder une copie et laisser passer une autre (**Théorème de non-clonage**).

On peut résumer les étapes de simulation par l'exemple montré dans le tableau (3.3).

Les bits aléatoires d'Alice (K)	0	1	1	0	1	0	0	1
Les bases aléatoires d'Alice (b)	$\oplus$	$\oplus$	$\otimes$	$\oplus$	$\otimes$	$\otimes$	$\otimes$	$\oplus$
L'état de Photon	$\longrightarrow$	$\uparrow$	$\nwarrow$	$\longrightarrow$	$\nwarrow$	$\nearrow$	$\nearrow$	$\uparrow$
Les bases aléatoires mesure par Ève	$\oplus$	$\otimes$	$\oplus$	$\oplus$	$\otimes$	$\oplus$	$\otimes$	
Ève mesures la polarisation et envoie	$\longrightarrow$	$\nearrow$	$\longrightarrow$	$\uparrow$	$\nwarrow$	$\longrightarrow$	$\nearrow$	$\nearrow$
Les bases Aléatoires de Bob	$\oplus$	$\otimes$	$\otimes$	$\otimes$	$\oplus$	$\otimes$	$\oplus$	$\oplus$
Les bits interprétés par Bob	$\longrightarrow$	$\nearrow$	$\nearrow$	$\nwarrow$	$\longrightarrow$	$\nearrow$	$\uparrow$	$\uparrow$
Les bases rapportées Par bob	$\oplus$		$\otimes$			$\otimes$		$\oplus$
Alice confirme	***oui***		***oui***			***oui***		***oui***
Alice confirme (Erreurs de Clé)	***non***		***non***			***oui***		***oui***
Les bits secrets restés						**0**		**1**

TABLE 3.2 – Exemple de l'échange BB84

3.5 Résultats de simulation et interprétation

Transmissions quantiques (1er étape)

Alice envoi à Bob une séquence des photons, chaque photon est polarisé aléatoirement des quatre polarisations : **Vertical 0°, - Horizontale 90° , Diagonale - 45 degré , - Antidiagonale 135 degré** (1er étape)**.**

Cette séquence contient **N=1000** photons distribué selon une loi uniforme, Dans cette partie de simulation on a créé un vecteur **b= {b$_1$,b$_2$,b$_3$,. . . .,b$_n$}**de tel sort que : **b$_i$** $\in${$\rightarrow$,$\uparrow$,$\nwarrow$, $\nearrow$} **s$_i$= photon polarisé d'ordre i. avec la fonction 'randi' de Matlab**.La figure 3.2 montre l'uniformité de distribution des polarisations de quelques expériences dans l'intervalle de N valeurs. Pour chaque photon reçu par Bob, il choisit aléatoirement une base de mesure parmi ces deux bases {+, $\times$},on a utilisé la même fonction de la 1er étape (1) **.**

Discussion publique (2^{ieme} étape)

Bob annonce ces bases de mesure à Alice via le canal public et ils éliminent tous les évènements dont ils utilisent des bases distinctes (2^{ieme} étape).

Estimation du taux d'erreur quantique (QBER)

Afin de réduire la différence de la clé tamisé entre Alice et de Bob due à une éventuelle intrusion, il est nécessaire de corriger les erreurs.C'est la phase ou' l'erreur de la clé tamisée est estimée. Ils choisissent alors de se révéler une fraction de la clé tamisé ($\sim$ 10%) , ceci leur permet d'estimer le QBER de la transmission. L'émetteur et le récepteur doivent calculer le taux d'erreurs observées et gardent cette transmission si le taux d'erreur est moins a un seuil désiré, sinon la clé sera avortée .

Réconciliation

Après cette étape, la clé de règlement sera obtenue après l'application du protocole de règlement à la clé normale. La réconciliation est un processus interactif qui se déroule dans le canal public. Le but de cette étape est de corriger les erreurs et de réduire uniformément la différence entre les commutateurs normaux de l'émetteur et du récepteur. Comme 50 Bits ont été envoyés via le canal public et supprimés après la fin du processus de réconciliation afin d'éviter d'être exploités par l'espion et d'assurer une clé de cryptage plus secrète.

Amplification et purification

Cette opération est effectuée dans les deux extrémités (Alice et Bob) ceci est dans le but de renforcer la sécurité du protocole BB84 :

- dans une clé tamisée des langueur N bits chaque 3 bits , le 3^{eme}bit est supprimé .

- Nous effectuons l'opération « XOR logique » des troisième et quatrième bits et répétons le l'opération le long de la clé de cryptage obtenue et à la fin du processus nous obtenons une nouvelle clé de chiffrement .

Observation

On répété cette **expérience 20 fois** pour obtenir le graphe représenté dans la figure 3.9 Dans le cas où il se trouve un homme au milieu (Voir figure 3.2) de la communication pour jouer le rôle de Bob pour Alice et le rôle d'Alice pour Bob, en choisissant arbitrairement des bases pour mesurer les qubits. Grâce à la propriété de non-clonage,celui-ci va commettre un taux considérable des erreurs et les retransmis vers Bob. Après que les protagonistes termineraient toute la procédure, ils s'en apercevront que la présence d'un espion.

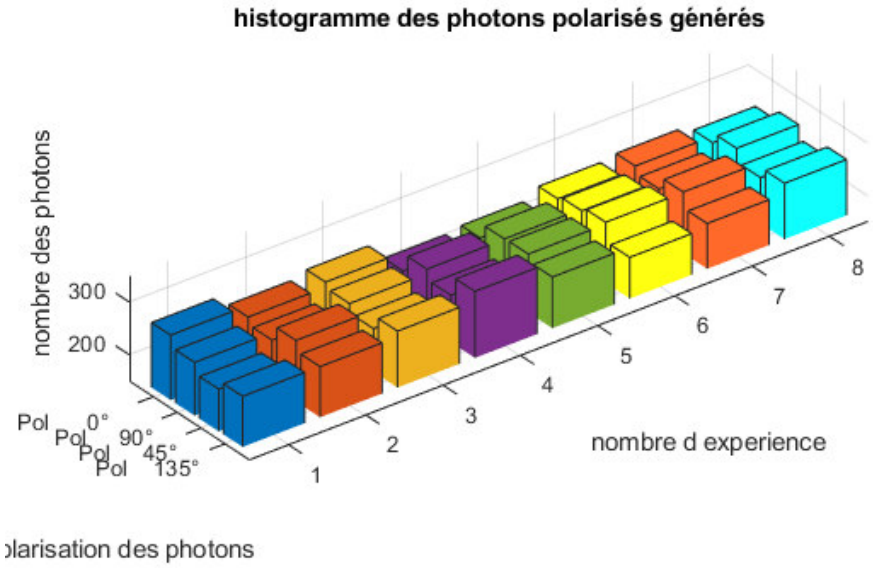


FIGURE 3.3 – Histogramme de la distribution uniforme de polarisation

La figure(3.3) montre l'uniformité de distribution des polarisations des photons générés par Alice ($\mathbf{N = 1000\ photons}$) dans ***les huit*** (***08***) expériences, et on voit que à chaque expérience Alice génère environ $[200 - 250]$ photons pour chaque polarisation.

Polarisation et choix de bases

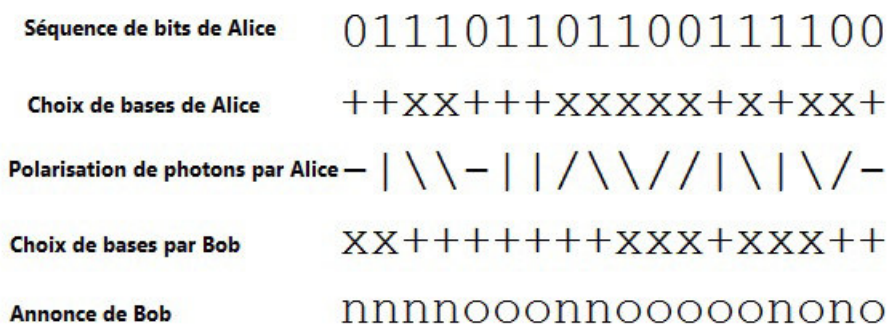


FIGURE 3.4 – Polarisation et choix de bases

La figure(3.4) montre une partie du choix de bases de Bob et Alice, et la polarisation des photons par Alice selon son choix de bases ou' on peut remarquer que pour chaque base « **Standard** » correspond un photon de polarisation 0◦ ou 90◦ c.à.d $\{\rightarrow, \uparrow\}$ et pour chaque base « **Diagonale** » correspond un photon de polarisation +45◦ ou +135◦ c.a'.d $\{\nearrow, \nwarrow\}$, ainsi que l'annonce de bases entre Alice et Bob dans le canal classique, si le choix de Bob et Alice est identique on met un « **o** » . sinon on met « **n** ». ce qui nous permet de rejeter et les bits qui correspondent a ces mauvais choix par la suite.

3.5.1 Protocole BB84 en absence de Eve.

La figures 3.5 représente l'échange de clé en l'absence d'espion, suivant le protocole BB84. Alice envoie des photons à Bob suivant les 4 polarisations -, \, | ou /. Bob mesure cette polarisation aléatoirement suivant les bases + et X. Il se trompe une fois sur deux (en jaune), ce qui induit des erreurs (en rouge). La clé secrète, constituée des bits correspondant aux bons choix de base, ne présente aucune erreur, ce qui garantit l'absence d'espion.

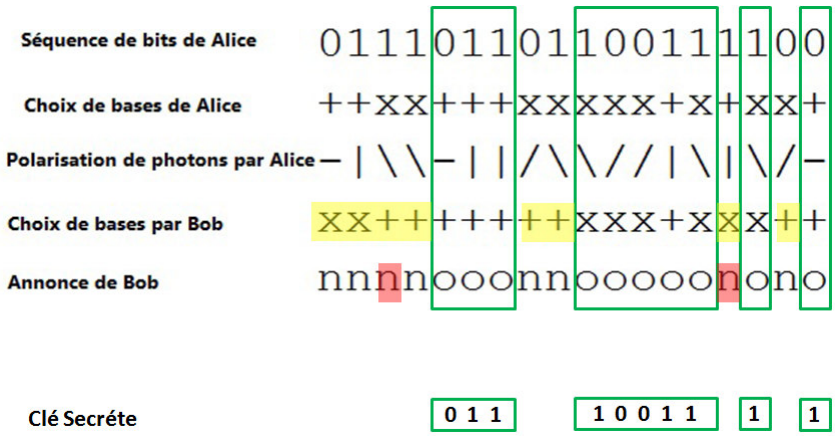


FIGURE 3.5 – Échange de clé en l'absence d'Eve

la figure 3.6, ci-dessus représente la longueur de clé après plusieurs expériences d'envoi de N=1000 photons en absence d'un espion, dont on voit qu'elle est toujours à l'ordre de 50% et sous les hypothèses on a montré l'aspect indétèrministe de la cryptographie quantique.

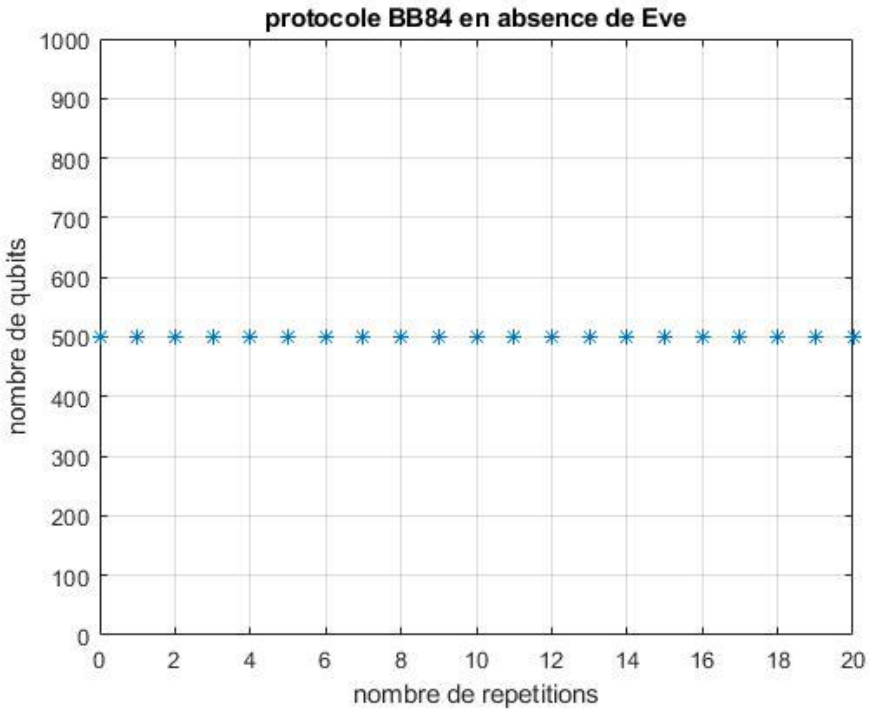


FIGURE 3.6 – Protocole BB84 en absence de Eve.

3.5.2 Protocole BB84 en présence de Ève

Influence de l'espion sur la polarisation

Dans la figure (3.7) on voit l'influence de l'espion sur la polarisation des photons, qui applique l'attaque « intersept-resend », Si Ève réussit à choisir la même base qu'Alice l'état de polarisation du photon qu'elle envoie à Bob ne sera pas perturbé, sinon elle prépare un qubit dans l'état correspondant à son résultat de mesure .

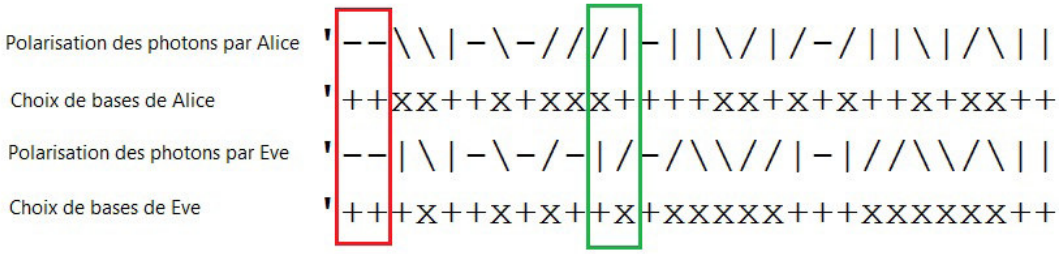


FIGURE 3.7 – Influence de l'espion sur la polarisation

Échange de clé en présence d'Ève

La figure 3.8 représente la longueur de clé crue après plusieurs expériences d'envoi de N=1000 photons en présence d'un espion, on remarque que le nombre des qubits dans chaque expérience est à l'ordre de 25%, et ça vérifie le théorème de non-clonage. A partir de ces résultats et sous les hypothèses pris en charge on peut conclure que le protocole BB84 est inconditionnellement sécurisé.

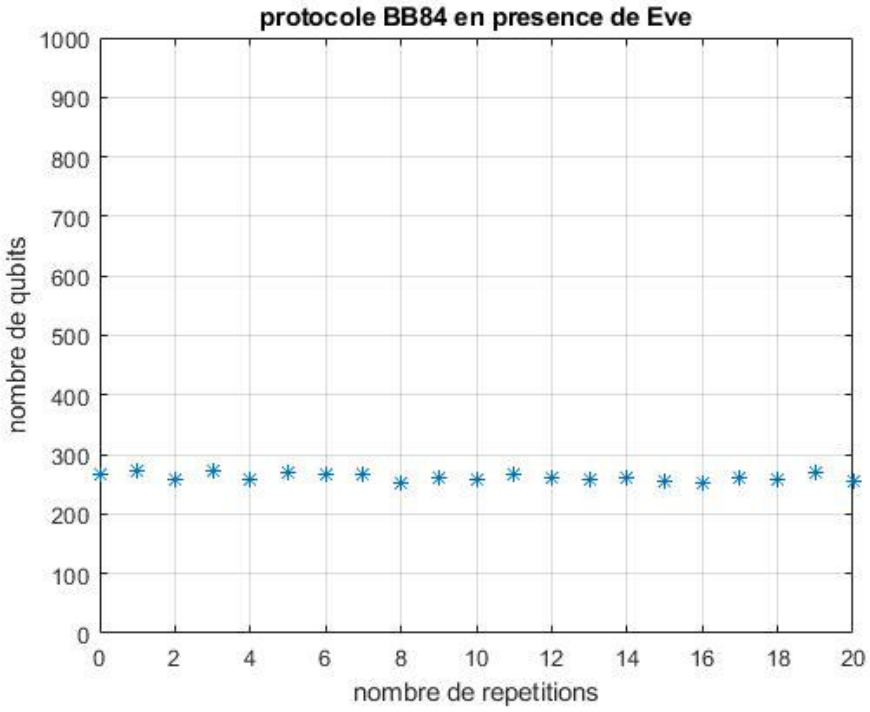


FIGURE 3.8 – Protocole BB84 en présence de Ève

Estimation du taux d'erreur quantique (QBER)

Cette perturbation permet à Bob et Alice de détecter la présence de Ève après l'étape de l'estimation du taux d'erreur. La figure (3.9) illustre le taux d'erreur calculé expérience qui est en général autour de 25% .

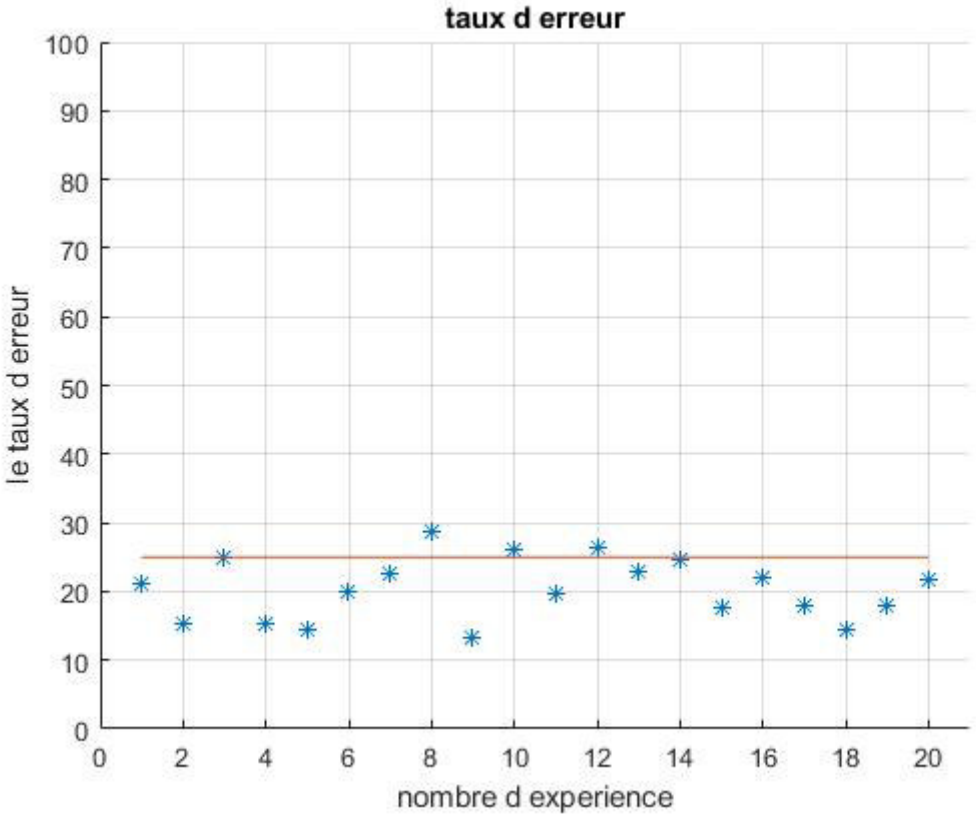


FIGURE 3.9 – Estimation du taux d'erreur

Tenant compte des résultats obtenus, dés que le **QBER** de la présence de Ève est inférieur à **25 %**, Eve n'a aucun moyen d'intercepter la clé sans que son intervention soit détectable. Si ce le QBER dépasse **25 %**, elle pourra prendre de l'information par type d'attaque **« interception - Renvoi »** et passer inaperçue. Donc **QBER** $<$ **25 %** assure la sécurisation totale de la communication .

Exemple de perturbation entre les deux clés

La figure (3.10) illustre une partie de la clé tamisé de Alice et de Bob ou' on remarque qu'il y a des bits qui n'ont pas la même valeur, ces perturbations sont la conséquence de l'espionnage de Ève.

	1	2	3	4	5	6	7	8
1	1	0	0	0	1	0	1	0
2	1	0	0	0	0	0	0	0

FIGURE 3.10 – Exemple de perturbation entre les deux clés

Finalement la figure (3.11) dévoile l'évolution de la taille de la clé après chaque étape du protocole BB84, et la figure (3.12) montre une partie de la clé finale que possède Alice et Bob après la confirmation qu'il n'y a pas d'espion. « longueur de la clé » .

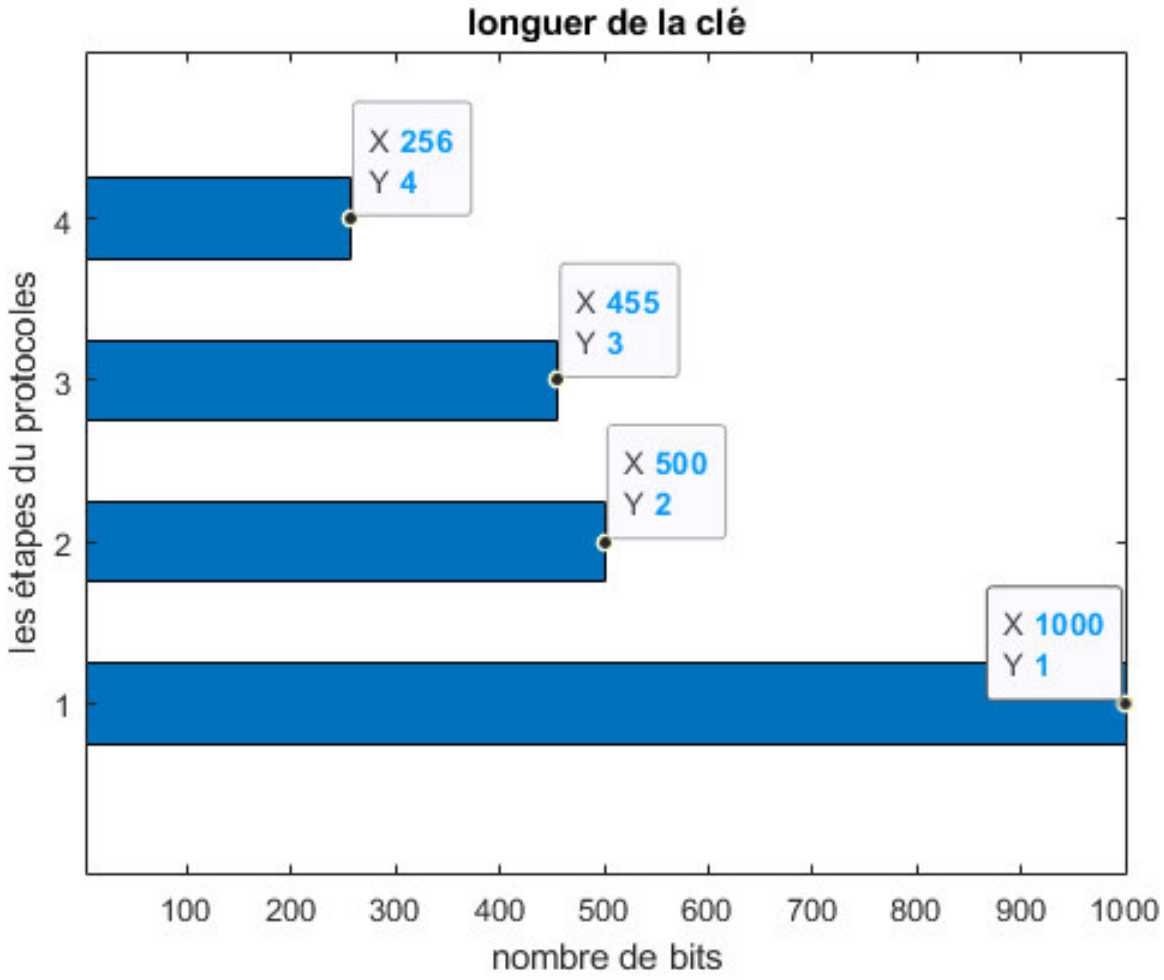


FIGURE 3.11 – Évolution de la taille de la clé

La clé finale de Bob et Alice

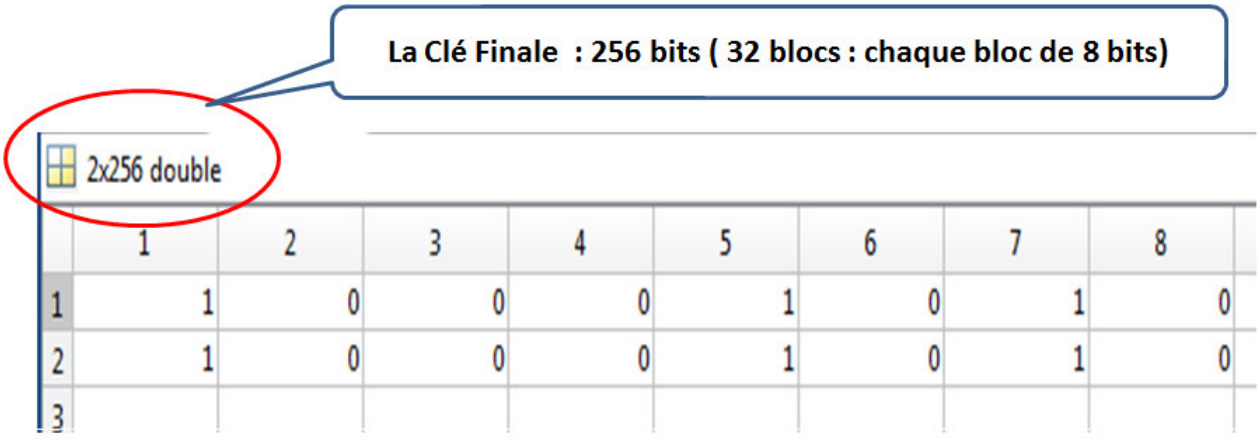


FIGURE 3.12 – La clé finale de Bob et Alice

3.6 Simulation du chiffre de Vernam (One-Time-Pad)

Après l'extraction de la clé quantique , on utilise cette clé pour le chiffrement et le déchiffrement de message par le chiffre de Vernam. nous utilisons Matlab pour simuler ce cryptosystème, la figure [3.13] montre la structure finale du procédure.

3.6.1 Explication du chiffrement de vernam

Le chiffre de Vernam Un tel chiffre parfait existe

c'est Gilbert Vernam, ingénieur au laboratoire de recherche de la compagnie "American Téléphone & Telegraph" qui l'a inventé et publié en 1926. Il peut être décrit simplement comme un chiffre de Vigenère, mais où la clé répond aux trois impératifs suivants [39] :

- elle est aussi longue que le texte à chiffrer ;
- elle est parfaitement aléatoire ;
- elle n'est utilisée que pour chiffrer un seul message, puis est immédiatement détruite.

C'est Claude Shannon, lui aussi chercheur dans les laboratoires de AT&T, qui prouva en 1949 le fait que ce chiffre est parfaitement sûr. La seule information dont on dispose si on intercepte le message chiffré est la longueur du message clair. De plus, tout chiffre parfaitement sûr est nécessairement une variante du chiffre de Vernam.

De façon moderne, le chiffre de Vernam (on parle aussi de masque jetable, pour souligner le fait que la clé doit être à usage unique), est implémenté de la façon suivante. Le message est d'abord converti informatiquement en suites de bits, c'est-à-dire de 0 et de 1. On prend une clé (complètement aléatoire) composée elle aussi d'une suite de 0 et de 1, aussi longue que le message à chiffrer.

On prend ensuite chaque bit du message clair et de la clé, et on en fait le ou exclusif. Rappelons que cette opération, que nous noterons $\oplus$, est définie par

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$.

Cette opération (qu'on peut voir comme l'addition en base 2, mais en oubliant la retenue), vérifie notamment les propriétés suivantes : $\mathbf{x} \oplus \mathbf{x} = \mathbf{0}$ **et** $\mathbf{x} \oplus \mathbf{0} = \mathbf{x}$.

Par exemple :

Si le message clair est **101110011**, si la clé est **011101000**, alors le message chiffré et **110011011**, comme le montre le tableau suivant :

Le chiffrement									
Message clair	1	0	1	1	1	0	0	1	1
Clé	0	1	1	1	0	1	0	0	0
Message chiffré	1	1	0	0	1	1	0	1	1
Le déchiffrement									
Message chiffré	1	1	0	0	1	1	0	1	1
Clé	0	1	1	1	0	1	0	0	0
Message clair	1	0	1	1	1	0	0	1	1

TABLE 3.3 – Explication du chiffrement de vernam [39]

À la réception, celui qui reçoit le message fait la même opération à partir du message chiffré : il prend donc chaque bit du message chiffré et fait le ou exclusif avec le bit correspondant de la clé. Il retrouve le message initial, à cause des deux propriétés précédentes.

3.6.2 Les étapes de simulation

La génération et la distribution de clé

Après avoir obtenu la clé quantique, par le protocole **BB84**, on utilise cette clé pour le chiffrement et le déchiffrement de message par le chiffre de Vernam. nous utilisons Matlab pour simuler ce cryptosystème, la figure [3.12] montre la structure finale du procédure.

Le message

Notre message clair est un ensemble de caractère « **ABCDEFJHIGKLMNOPQRSUVWXYZ** » , cet ensemble est converti en une séquence binaire ou chaque caractère est codé sur six bits pour pouvoir le chiffré (**voir l'annexe** 2).

Le chiffrement

après la conversion de notre message en une séquence binaire on peut effectuer une opération de chiffrement facile et simple comme **XOR** (ou exclusive) entre la séquence binaire et la clé. Si le message a une longueur supérieure à celle de la clé alors on le découpe en blocs, tel que chaque bloc possède la même longueur de la clé, et on chiffre ces blocs par des clés consécutives dans l'ensemble de clés, chaque bloc par une clé unique pour satisfaire la deuxième condition de Vernam « **La clé est aussi longue que le message** »

Le déchiffrement

Comme le chiffre de Vernam est un cas particulier du chiffrement symétrique, en utilisant la même clé du chiffrement on fait l'opération similaire pour déchiffrer le message.

3.6.3 Résultat de simulation

Nous avons fait une simulation de chiffrement et déchiffrement d'un message et les résultats sont représentés dans la figure (3.13)

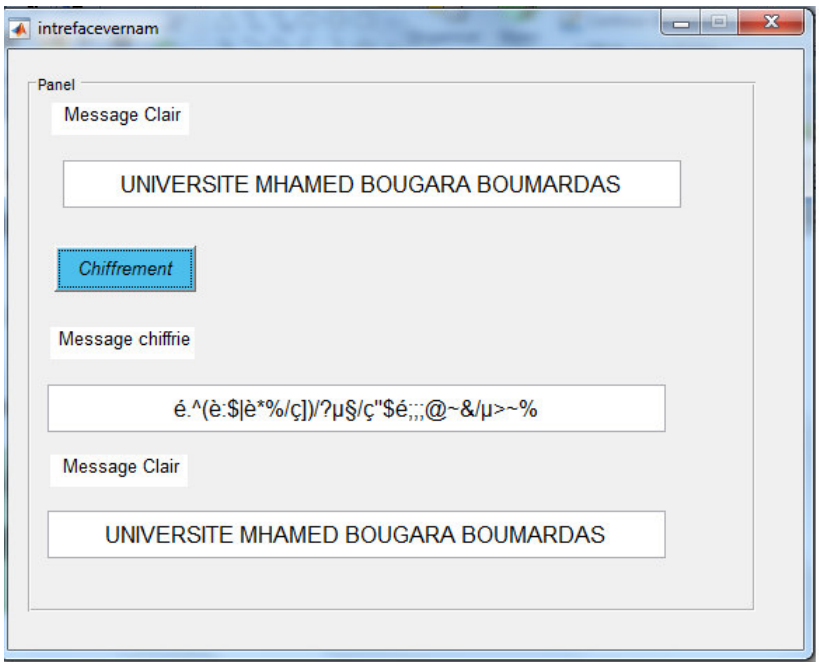


FIGURE 3.13 – Résultat de simulation de chiffrement et déchiffrement d'un message

Note importante

Pour satisfaire les autres principales conditions du masque-jetable on doit jeter la clé après chaque utilisation et utiliser une autre clé de l'ensemble de clés qui sont générées aléatoirement.

3.7 Conclusion

Dans ce chapitre nous avons procéder à l'analyse des performances et simulation du protocole BB84 avec le chiffrement de vernam sous le logiciel Matlab.

Notre apport dans ce travail se situe dans la réconciliation ,purification et amplification de clé finale qui va servir au chiffrement. En effet nous avons proposer une procédure qui est relative au calcul logique par des XOR et le calcul de parité .

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Pour conclure notre travail, la cryptographie quantique a pour objet de rendre notre communication inviolable vis à vis d'intrusion extérieure « **inter sept-resend (I-R)** » . Cela est possible grâce aux principes de la mécanique quantique sur la mesure, **le théorème de non-clonage et de l'incertitude d'Heisenberg**. Elle est utilisée pour secourir les méthodes de cryptographie mathématiques et algorithmiques qui sont de plus en plus vulnérable aux attaques des intrus. Pour la transmission par fibre optique, c'est l'élément clé pour la sécurisation de l'information. La ***QC*** est une méthode de cryptographie basée sur des procédés physiques et sur **le code de Vernam** à laquelle est adopté le protocole ***BB84***.

Tout d'abord, nous avons fait une recherche bibliographique sur le domaine de la mécanique quantique et son application dans la cryptographie symétrique par les échanges de clés, ce qui nous a permis de présenter le chapitre deux.

Ensuite, nous avons présenté le troisième chapitre sur la simulation des échanges de clés quantiques utilisant la norme ***BB84*** qui est, à notre connaissance, la plus utilisée. finalement nous avons analyse le protocole qui consiste a faire un change de clé en utilisant le protocole ***BB84*** , **la clé obtenue est utilisée pour chiffrer les message par le One-Time-Pad. Pour mettre en oeuvre ce protocole**, nous avons proposé une méthode de réconciliation étudié et simulé sous Matlab le ***BB84*** . Après l'interprétation des résultat nous avons effectuer une opération ***XOR*** entre la clé obtenue pour pouvoir l'utiliser pour le chiffrement.

En perspectives nous proposons la continuation de ce travail en remplaçant le chiffrement de Vernam par un autre tel que par exemple celui du *DES* et *AES* (*Voir annexe* 3) .

BIBLIOGRAPHIE

[1] BERRAHAL Azzeddine , HAMDANI Houssam « Échange de clés cryptographique quantique et chiffrement symétrique », Mémoire de fin d'étude, ESDAT, 2018.

[2] Lamas, Daniel, « La cryptographie », haute école de gestion de Genève, thèse de Bachelard,2015.

[3] LERVILLE, Edmond, « Les cahiers secrets de la cryptographie, Editions du Rocher », 1972.

[4] Manuel Sabban, « Sécurité en cryptographie quantique utilisant la détection homodyne d'états cohérents à faible énergie »,thèse de doctorat ,Paris, 2009.

[5] Cobourne, S. « Others Quantum Key Distribution Protocols and Applications », Surrey , England, 2011.

[6] Renaud Dumont,« Cryptographie et Sécurité informatique », Université de Liége, Belgique, 2010.

[7] Alice Lan,Benoit Vandevelde ,cours :« Panorama des algorithmes de Cryptographie », Université de Nantes,13mars2011.

[8] Kaouthar Bousselam,« Résistance des circuits cryptographiques aux attaques en faute », Micro et nanotechnologies/Microélectronique,Université Montpellier II-Sciences et Techniques du Languedoc,Français,2012.

[9] Hacini Souleyman Boumedyen, Inal Mohamed Taha, « Implémentation d'algorithmes de Cryptographie », mémoire de licence, Université Abou Bakr Belkaid, Tlemcen, 2014.

[10] Gael Thomas, « design et analyse de sécurité pour les constructions pour la cryptographie symétrique », mémoire de master, Université de limoges, juin 2015.

[11] Thomas Baignkres, Pascal Junod, Yi Lu, Jean Monnerat, Serge Vaudenay, «A classical introduction to cryptography » Springer, 2006.

[12] Nadia El Merabt, « Les concepts fondamentaux de la cryptographie », Mémoire de master, Université de Caen, France, mars 2014

[13] AZZOUZI Oussama,HADDADI Ferhat,« Plateforme de chiffrement/déchiffrement pour la sécurisation du stockage et de la transmission de l'information »,mémoire d'ingéniorat École nationale supérieur de l'informatique.2012.

[14] Nicolas Estibals,«Algorithme et arithmétiques pour l'implémentation de couplages cryptographiques », Thèse de magister,Université de Lorraine,Octobre2013.

[15] Diffie,Whitfield and Hellman,Martin, « New directions in cryptography », 1976.

[16] Gamal, T.E. « A public key cryptosystem and a signature scheme based on discrete logarithms ». Proceedings of CRYPTO on Advancesin cryptology (1984) , pp.10–18. 1984.

[17] Frédéric GROSSHAN Set Philippe GRANGIER,« La cryptographie quantique : l'incertitude quantique au service de la confidentialité », EDP Sciences,Photoniques, 2014.

[18] BROUHA . Mustapha , HMOKRANI . Karim, « Etude et simulation de la Cryptographie Quantique », Mémoire de fin d'étude, Université ABDERRAHMANE MIRA Bejaia , 2012-2013.

[19] Q. GLORIEUX, « Les principes de la mécanique quantique »,Support de cours /Université Pierre et Marie Curie 2015/2016.

[20] Douglas R.Stinson.« Cryptography Theory and Practice », 2nded.CRCPress,Inc., 2002.

[21] Lawal Muhammad Aminu,« Implementing Big Data Management on Grid Computing Environnent », International Journal Of Engineering And Computer Science,2014.

[22] Dramaix Florence, van den Broek Didier, Wens Vincent,« La Cryptographie Quantique »,Printemps des Sciences ,2003.

[23] NGUYEN Thanh Mai, « Etudier et implémenter une simulation du protocole d'échange de clef quantique BB84 », Rapport de stage de fin d'étude, Paris, Janvier 2005.

[24] Ouchao, B and ElKinani,EH,« Statistical analysis of common qubits between Alice and Bob in BB84 protocol ». Article,Journa of Contemporary Engineering Sciences.2011.

[25] Patrick J.Coles,a,Eric M.Metodiev,and Norbert Lüt kenhaus,« Numerical approachfor unstructured quantum key distribution », Article,Nature communications,2016.

[26] Hitesh Singh,D.L.GuptaA.KSingh,« Quantum keydistribution protocols », Article, IOSR Journal of Computer Engineering,2014.

[27] Rima DJELLAB, « Cryptographie quantique, Nouvelles approches », Thèse de doctorat, Batna, 2010.

[28] M.Elboukhari, « Quntum Key Distribution Protocol :A Survey », int.Journal of Universal Computer Sciences /vol.l-2010/Lss.2

[29] Shall, Sadman and Monir, Md Serajum and Rahman, Md Saifur, « Numerical modeling and simulation of quantum key distribution systems under non-ideal conditions », Article, Telecommunications and Photonics (ICTP), 2017 IEEE International Conférence on, 2017.

[30] Olivier L. « Multidimentional quantum key distrubution with single side pulse and single side band modulation multiplexing ». Thèses de doctorat. Institut de Géorgie de l'ingénieur de Technologie.December 2005.

[31] Jean-Michel ANDRÉ ,« Informatique Quantique - Comprendre le Quantum Computing pour se préparer à l'inattendu », DSI du Groupe SEB, Pilote du groupe de travail Cigref ,FÉVRIER 2020.

Annexe 1 « Programme Matlab du protocole BB84 »

```
%% polarisation des photons
N=1000;
a='-';b='/';d='\';c='|';
%% les bases
x='x';y='+';
Qber=[];m=25;tm=[];poolar=[];
for u=1:20 tm=[tm m];
%Sequence binaire aléatoire
prim_Al_key=randi([0 1],1,N);
f=randi([1 2],1,N);
for k=1:N
if(f(k)==1)
%%Alchoix : choix des bases par alice
Alchoix(k)=x;
elseif(f(k)==2)
Alchoix(k)=y;
end
end
%%%% polarisation des photons par Alice
aa=0;bb=0;cc=0;dd=0;
for k=1:N if((Alchoix(k)==y) && (prim_Al_key(k)==0))
pol(k)=a;
aa=aa+1;
elseif((Alchoix(k)==y) && (prim_Al_key(k)==1))
pol(k)=c;
cc=cc+1;
elseif ((Alchoix(k)==x) && (prim_Al_key(k)==0))
pol(k)=b;
bb=bb+1;
elseif((Alchoix(k)==x) && (prim_Al_key(k)==1))
pol(k)=d;
dd=dd+1;
end
end
```

Les nombres de polarisation chaque expérience pol0°(aa),pol45°(bb),pol90°(cc), pol135°(dd).

```
polar=[aa bb cc dd];
ff=randi([3 4],1,N);
for k=1:N if(ff(k)==3)
Bochoix(k)=y;
elseif(ff(k)==4)
```

```
%%% choix des bases par Bob
Bochoix(k)=x;
end
end
for k=1 :N
%%%% Detection des photon par Bob
if ((Bochoix(k)==y) && (pol(k)==a))
prim_Bo_key(k)=0;
elseif ((Bochoix(k)==y) && (pol(k)==c))
prim_Bo_key(k)=1;
elseif((Bochoix(k)==x) && (pol(k)==b))
prim_Bo_key(k)=0;
elseif ((Bochoix(k)==x) && (pol(k)==d))
prim_Bo_key(k)=1;
elseif((Bochoix(k)==y) && (pol(k)==b))
prim_Bo_key(k)=randi([0 1]);
elseif((Bochoix(k)==y) && (pol(k)==d)) prim_Bo_key(k)=randi([0 1]);
elseif ((Bochoix(k)==x) && (pol(k)==a))
prim_Bo_key(k)=randi([0 1]);
elseif((Bochoix(k)==x) && (pol(k)==c))
prim_Bo_key(k)=randi([0 1]);
end
end
%% Annonce des bases entre Alice et Bob
for k=1 :N
if(Alchoix(k)==Bochoix(k))
w(k)='o';
elseif(Alchoix(k)~=Bochoix(k))
w(k)='n';
end
end
for k=1 :N
if(w(k)=='o')
Al_key(k)=prim_Al_key(k);
Bo_key(k)=prim_Bo_key(k);
elseif (w(k)=='n')
Al_key(k)=3;
Bo_key(k)=3;
end
end
%%%estimation du taux d'erreur
Al_key(Al_key>1)=[];
Bo_key(Bo_key>1)=[];
```

```
NN=length(Al_key);% le Nombre des bits (la langueur de cle "Al_key")
nn=length(Al_key)/10;% en divise la langueur de cle sur 10
err=0;
for n=1 :nn
if(Al_key(n)~=Bo_key(n))
err=err+1;
end
end
%% la clé aprés l'estimation du taux d'erreur
for i=1 :NN
if(i<=nn) All_key(i)=3;
% Boo_key(i)=3;
elseif(i>nn) All_key(i)=Al_key(i);
Boo_key(i)=Bo_key(i);
end
end
%%%% purification et amplification
All_key(All_key>1)=[];
Boo_key(Boo_key>1)=[];
NNN=length(All_key);
f_key=Boo_key;
for nnn=3 :3 :NNN
% le Nombre des bits (la langueur de clé "All_key")
f_key(nnn)=-3;
end
f_key(f_key<0)=[];
ff_key=f_key; kk=1;
for kk=2 :3 :length(f_key)
if kk<length(f_key)
ff_key(kk)=xor(f_key(kk),f_key(kk+1));
kk=k+1;
else
end
end
for ii=1 :length(f_key)
if(ii<201) ff_key(ii)=f_key(ii);
else ff_key(ii)=3;
end
end
ff_key(ff_key>2)=[];
%%%%% En présence de Eve %%%%%%%%%
%% choix des bases par alice
fff=randi([5 6],1,N);
```

```
for k=1 :N if(fff(k)==5) Ev_choix(k)=x;
elseif(fff(k)==6) Ev_choix(k)=y;
end
end
%%%% Interception des photon par Eve
for k=1 :N
if ((Ev_choix(k)==y) && (pol(k)==a))
Ev_key(k)=0;
elseif ((Ev_choix(k)==y) && (pol(k)==c))
Ev_key(k)=1;
elseif((Ev_choix(k)==x) && (pol(k)==b))
Ev_key(k)=0;
elseif ((Ev_choix(k)==x) && (pol(k)==d))
Ev_key(k)=1;
elseif((Ev_choix(k)==y) && (pol(k)==b))
Ev_key(k)=randi([0 1]);
elseif((Ev_choix(k)==y) && (pol(k)==d))
Ev_key(k)=randi([0 1]);
elseif ((Ev_choix(k)==x) && (pol(k)==a))
Ev_key(k)=randi([0 1]);
elseif((Ev_choix(k)==x) && (pol(k)==c))
Ev_key(k)=randi([0 1]);
end
end
%% Resend par EVE
ffff=randi([7 8],1,N);
fffff=randi([ 9 10],1,N);
for k=1 :N if(ffff(k)==7)%|| (bb(k)==2))
s(k)=a;
elseif(ffff(k)==8) %|| (bbb(k)==4))
s(k)=c;
end if(fffff(k)==9) %|| (bbbb(k)==6))
ss(k)=d;
elseif(fffff(k)==10) %|| (bbbb(k)==8))
ss(k)=b;
end
end
for k=1 :N
if(Alchoix(k)==Ev_choix(k))
Ev_pol(k)=pol(k);
elseif((Alchoix(k)==y) && (Ev_choix(k)==x))
Ev_pol(k)=ss(k);
elseif((Alchoix(k)==x) && (Ev_choix(k)==y))
```

```
Ev_pol(k)=s(k);
end
end
%%%% Détection de photon par Bob en présence de Eve
for k=1 :N if ((Bochoix(k)==y) && (Ev_pol(k)==a))
Boev_key(k)=0;
elseif ((Bochoix(k)==y) && (Ev_pol(k)==c))
Boev_key(k)=1;
elseif((Bochoix(k)==x) && (Ev_pol(k)==b))
Boev_key(k)=0;
elseif ((Bochoix(k)==x) && (Ev_pol(k)==d))
Boev_key(k)=1;
elseif((Bochoix(k)==y) && (Ev_pol(k)==b))
Boev_key(k)=randi([0 1]);
elseif((Bochoix(k)==y) && (Ev_pol(k)==d))
Boev_key(k)=randi([0 1]);
elseif ((Bochoix(k)==x) && (Ev_pol(k)==a))
Boev_key(k)=randi([0 1]);
elseif((Bochoix(k)==x) && (Ev_pol(k)==c))
Boev_key(k)=randi([0 1]);
end
end
%%%estimation du taux d'erreur
for k=1 :N
if (w(k)=='o')% vecteur (oui)
Bo_ev_key(k)=Boev_key(k);
elseif(w(k)=='n')% vecteur (non)
Bo_ev_key(k)=4;
end
end
Bo_ev_key(Bo_ev_key>2)=[];
verr=0;
for n=1 :nn
if(Al_key(n)~=Bo_ev_key(n))
verr=verr+1;
end
end
Qber(u)=verr/nn*100;
for i=1 :NN
if(i<=nn) fin_Al(i)=3;
fin_Bo(i)=3;
elseif(i>N)
fin_Al(i)=Al_key(i);
```

```
fin_Bo(i)=Bo_ev_key(i);
end
end
% mm=catégorie ({'polar_0','polar_45','polar_90','polar_135'});
figure('Name','Estimation du taux d erreur en présence de Eve')
hold on plot(Qber,'*')
plot(tm)
% plot(MM,'r+')
title('taux d erreur') xlabel('nombre d expérience')
ylabel('le taux d erreur')
axis([0 21 0 100])
grid on hold off
figure('Name','Histogramme de la distribution uniforme de polarisation')
bar3(poolar')
title('histogramme des photons polarisés générés')
ylabel('pilarisation des photons')
xlabel('nombre d expérience')
zlabel('nombre des photons')
axis([0.5 8.5 0 5 150 350])
figure('Name','La taille de la clé quantique après chaque étape')
barh(lgth,0.5);
title('longueur de la clé')
xlabel('nombre de bits')
ylabel('les étapes du protocoles')
fff_key=[ff_key;ff_key];
choix=[pol;Alchoix;Bochoix;w;Ev_choix]
choixx=[pol;Alchoix;Ev_pol;Ev_choix]
x=0 :20; y=500*ones(size(x))+rand(size(x));
figure plot(x,y,'*') axis([0 20 0 1000])
title('protocole BB84 en absence de Eve')
xlabel('nombre de repetitions')
ylabel('nombre de qubits')
grid y=250*ones(size(x))+25*rand(size(x)); figure plot(x,y,'*')
axis([0 20 0 1000])
title('protocole BB84 en presence de Eve')
xlabel('nombre de repetitions')
ylabel('nombre de qubits') grid
```

Annexe 2 « Le codage des caractères »

Transformer une suite d'octets en une suite de caractères imprimables en **ASCII** non étendu : On transforme 3 octets (**24 bits**) en tranches de 6 bits .

Valeur		Codage	Valeur		Codage	Valeur		Codage
Décimal	Binaire		Décimal	Binaire		Décimal	Binaire	
0	000000	A	22	010110	W	44	010001	s
1	000001	B	23	010111	X	45	010010	t
2	000010	C	24	011000	Y	46	010011	u
3	000011	D	25	011001	Z	47	010100	v
4	001000	E	26	011010	a	48	010101	w
5	000101	F	27	011011	b	49	010110	x
6	000110	G	28	011100	c	50	010111	y
7	000111	H	29	011101	d	51	011000	z
8	001000	I	30	011110	e	52	110100	0
9	001001	J	31	011111	f	53	110101	1
10	001010	K	32	100000	g	54	110110	2
11	001011	L	33	100001	h	55	110111	3
12	001100	M	34	000111	i	56	111000	4
13	001101	N	35	001000	j	57	111001	5
14	001111	O	36	001001	k	58	111010	6
15	010000	P	37	001010	l	59	111011	7
16	010000	Q	38	001011	m	60	111100	8
17	010001	R	39	001100	n	61	111101	9
18	010010	S	40	001101	o	62	111110	+
19	010011	T	41	001111	p	63	1111111	/
20	010100	U	42	010000	q			
21	010101	V	43	010000	r			

Table 3.4 – Le codage des caractères

Annexe 3 « Chiffrement par bloc »

Chiffrement par bloc

Dans un algorithme de chiffrement par bloc, chaque message clair est découpé en blocs de taille fixe de même longueur et chiffré à l'aide d'une clé unique. Ces algorithmes sont en général construits sur un modèle itératif. Il utilise une fonction F qui prend une clé secrète k et un message M de n bits. La fonction F est itérée un certain nombre de fois (**nombre de tours**). Lors de chaque tour, la clé k est différente et on chiffre le message qui vient d'être obtenu de l'itération précédente. Les différentes clés k(i) qui sont utilisées sont déduites de l a clé secrète .

Les algorithmes les plus connus des systèmes cryptographique symétriques sont : le DES et l'AES .

L'algorithme DES

L'algorithme DES, Data Encryptions Standard, a été créé dans les laboratoires de la firme IBM Corp. Il est devenu le standard du NIST (***National Institute of Standards and Technology*** : ***Institue National de Standards et Technologie)*** en 1976 et a été adopté par le gouvernement Américain en 1977 (Voir figure 3.14) .

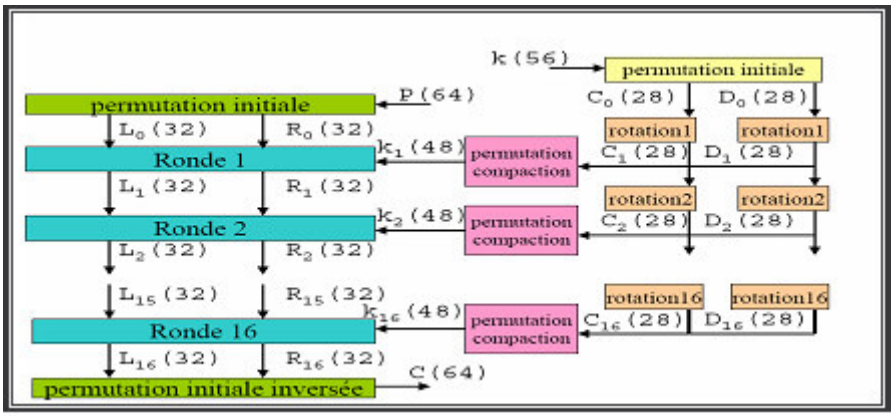


FIGURE 3.14 – Architecture globale du DES

Fonctionnement global du DES

Ce nouveau standard chiffre un bloc de 64 bits par une série de 16 rounds identiques, à l'aide d'une clé de 64 bits dont 56 bites aléatoires et 8 bits pour la détection des erreurs, soit un total de 256 clés (Voir figure 3.15).

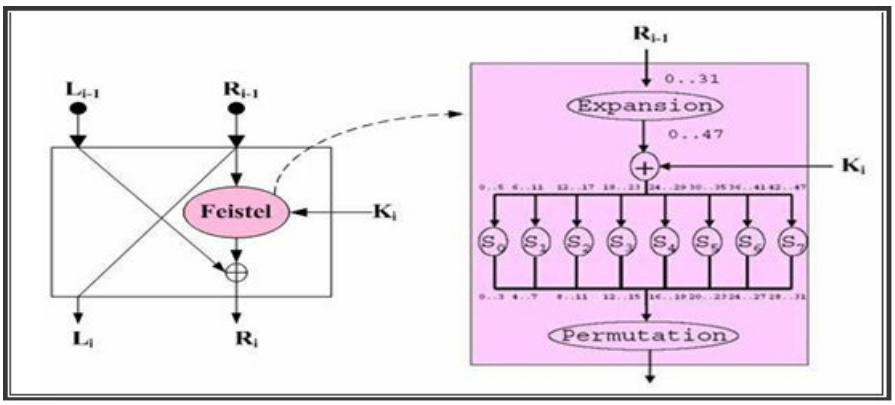


FIGURE 3.15 – Fonctionnement d'un round de chiffrement

Un round comprend une permutation, un **XOR** et une fonction **de Feistel,** cette fonction dont l'entrée est de 32 bits applique à son tour une expansion pour atteindre 48 bits suivis d'un XOR avec la clé et une substitution par groupe de six bits, pour terminer avec une permutation ; comme c'est illustré dans la figure .

La fiabilité du DES a été mise en question en 1995, à cause de ses faiblesses :

- Problème de longueur des clés.
- Problème du choix des substitutions.

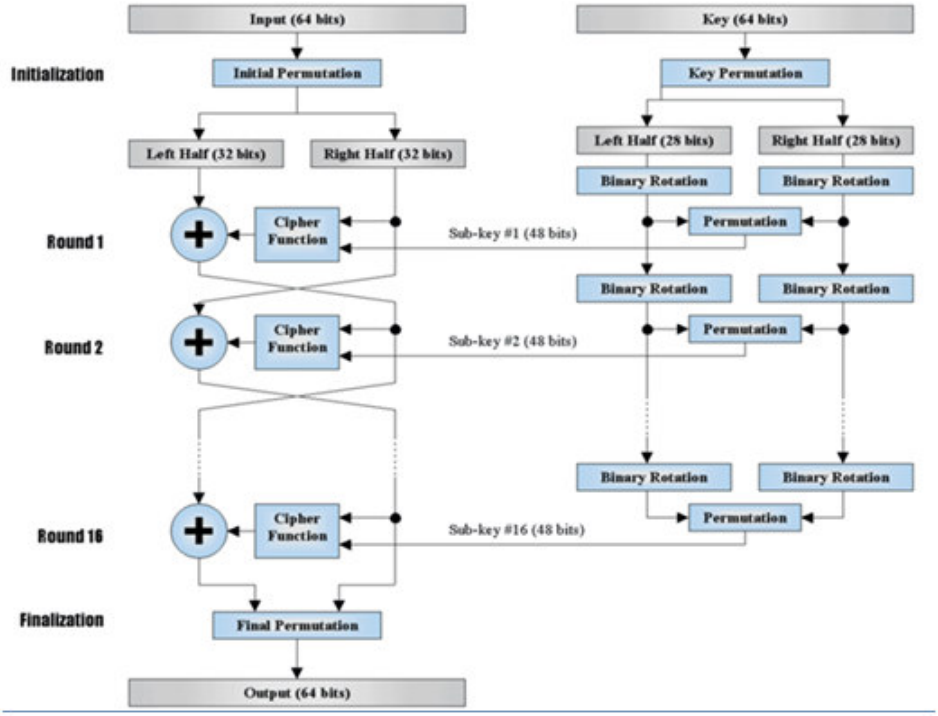


FIGURE 3.16 – Schéma de Fonctionnement.

Le triple DES

En 1999 les doutes étaient confirmés après avoir réussi à casser le DES, Le NIST a opté pour une solution intermédiaire c'était le triple DES.

Comme son nom l'indique, le triple DES est une succession des algorithmes de chiffrement ou de déchiffrement DES trois fois, les combinaisons adoptées sont :

- Le mode EDE (Encryption Decryption Encryption)

consiste à chiffrer le message clair avec une clé k1, puis le déchiffrer avec une autre clé k2, pour qu'à la fin le chiffrer avec la première clé k1, la formule est la suivante : E (k1, D (k2, E (k1, m))) (Voir figure 3.17).

Il nécessite : 3 opérations, 2 clés chaque clé est de 112 bits.

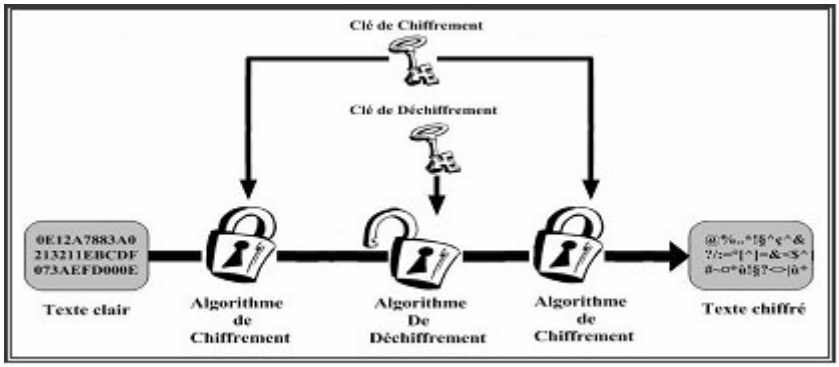


FIGURE 3.17 – Schéma du triple DES en mode EDE

- Le mode EEE (Encryption Encryption Encryption)

consiste à répéter l'algorithme de chiffrement trois fois avec trois clés différentes : E (k1, E (k2, E (k3, m))) Il nécessite : 3 opérations ,3 clés chaque clé est de 168 bits (Voir figure 3.18).

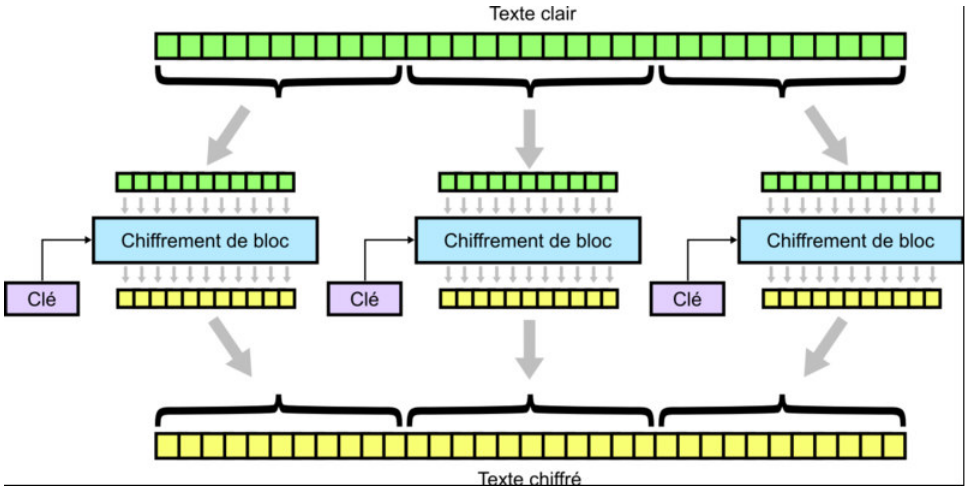


FIGURE 3.18 – Schéma du triple DES en mode EEE

Avantages et inconvénients du triple DES

- Équivalent à doubler la taille effective de la clé (longueur sure).

- Très robuste et efficace contre toutes les attaques faisables connues.
- Comme son nom l'indique il triple les opérations au mépris du temps d'exécution.

Utilisation DES

Quelques applications

- Communication secrète (SSL de https, SSH, GSM,...) .
- Authentification d'agents (Login, OTP, GSM,...).
- Signature de contrats électroniques.
- Paiement par carte à puce.
- Paiement en ligne.
- Paiement hors-ligne .
- TV payante .
- Vote électronique.

Le protocole des cartes bancaires

Le protocole d'utilisation d'une carte bancaire doit permettre d'authentifier les transactions. L'authentification concerne deux préoccupations :
  – prouver que la carte utilisée a bien été émise par l'autorité légitime ;
  – prouver qu'une carte légitimement émise n'est pas utilisée frauduleusement.

Chaque carte bancaire comporte un code secret PIN, garantissant que seul la personne autorisée peut utiliser la carte. Il s'agit en général d'un nombre de 4 chiffres .

Fabrication de la carte

1. Un PIN est généré aléatoirement
2. Le PIN, et d'autres informations (numéro de compte par exemple) est chiffré par un DES triple, en utilisant une clé de travail
3. Une fonction à sens unique extrait certains bits du bloc chiffré obtenu pour construire le PIN offset
4. Le PIN Offset est enregistré dans la mémoire de la carte, avec le numéro de compte.
5. Le PIN est posté au possesseur de la carte et effacé de la mémoire.
6. Une fois la carte émise, seul le possesseur légitime de la carte connaît le PIN.

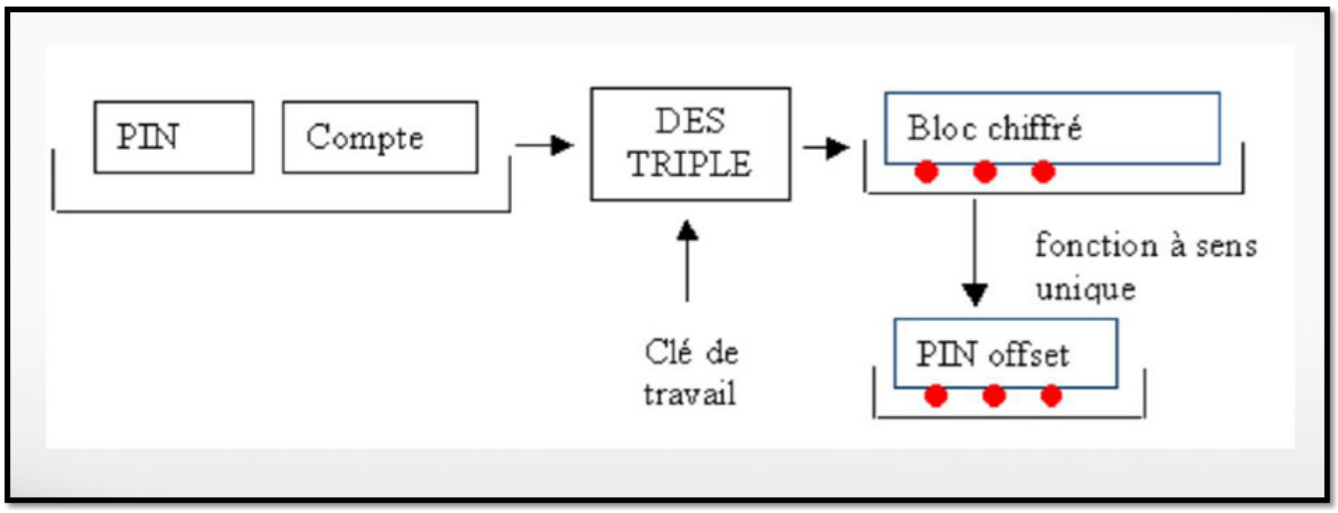


FIGURE 3.19 – Fabrication de la carte

Validation d'une transaction

1. La validation d'une transaction peut se faire localement à partir d'un terminal. Elle se limite alors à un protocole d'identification du propriétaire.
   – Le possesseur de la carte tape son PIN sur le terminal ;
   – le terminal chiffre le PIN et le numéro de compte par un DES triple, en utilisant la clé de travail et extrait le PIN offset ;
   – le PIN offset est comparé avec celui enregistré sur la carte.
2. Le terminal doit avoir accès à la clé de travail qui a servi à chiffrer le PIN.
3. Une clé maître (TMK) est générée et installée pour chaque terminal. Le système central télécharge les clés de travail, chiffrées avec la TMK correspondante.

Retrait d'espèces dans un distributeur

1. Un client insère sa carte dans le distributeur ;
2. la piste magnétique est lue ;
3. le client entre son code PIN ; il est stocké dans un buffer inaccessible ;
4. le client tape la somme qu'il veut retirer ;
5. le message contenant le PIN et la somme est chiffré avec la TMK et envoyé au serveur ;
6. Le serveur de la banque déchiffre le message et calcule le PIN Offset avec la clé de travail correspondante. Il le compare au PIN Offset stocké dans sa base de données.
7. La transaction est approuvée.

1.2 L'algorithme AES

L'AES (**Advanced Encryption Standard**) est, comme son nom l'indique, un standard de cryptage symétrique destiné à remplacer le DES (**Data Encryption Standard**) qui est devenu trop faible au regard des attaques actuelles (Voir figure 3.21) .

Historiquement, le développement de l'AES a été instigué par le NIST (**National Institue of Standards and Technology**).

Il est également approuvé par la NSA (**National Security Agency**) pour l'encryption des informations dites très sensibles.

Cet algorithme suit les spécifications suivantes :

- L'AES est un standard, donc libre d'utilisation, sans restriction d'usage ni brevet.
- C'est un algorithme de type symétrique
- C'est un algorithme de chiffrement par blocs
- Il supporte différentes combinaisons **(longueur de clé)**-(**longueur de bloc**) : 128-128, 192-128 et 256-128 bits (en fait, l'AES supporte également des tailles de blocs variables, mais cela n'est pas retenu dans le standard)

En termes décimaux, ces différentes tailles possibles signifient concrètement que :

- 3.4 x 1038 clés de 128-bit possibles .
- 6.2 x 1057 clés de 192-bit possibles.
- 1.1 x 1077 clés de 256-bit possibles **.**

Pour avoir un ordre d'idée, les clés DES ont une longueur de 56 bits (64 bits au total dont 8 pour les contrôles de parité), ce qui signifie qu'il y a approximativement 7.2 x 1016 clés différentes possibles.

Cela nous donne un ordre de 1021 fois plus de clés 128 bits pour l'AES que de clés 56 bits pour le DES. En supposant que l'on puisse construire une machine qui pourrait cracker une clé DES en une seconde (donc qui puisse calculer 255 clés par seconde), alors cela prendrait environ 149 mille milliards d'années pour cracker une clé AES .

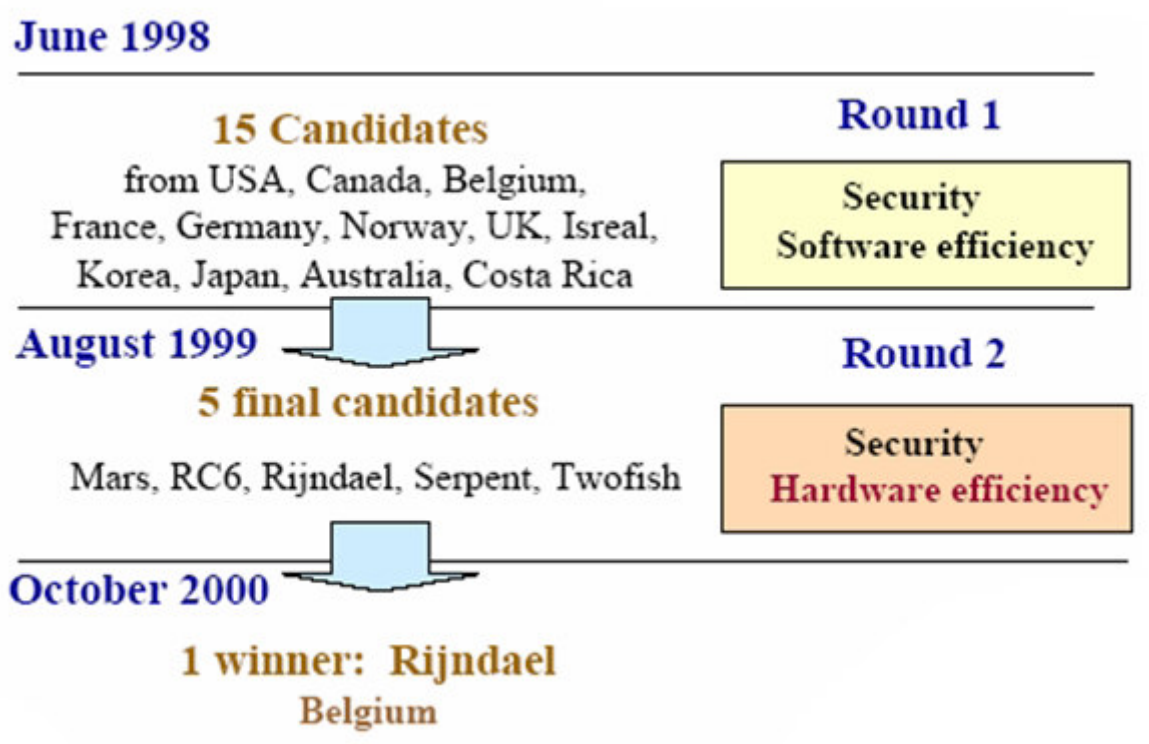


FIGURE 3.20 – Différents participants au concours AES

1.2.1 Caractéristiques et points forts de l'AES

Le choix de cet algorithme répond à de nombreux critères plus généraux dont nous pouvons citer les suivants :

- Sécurité ou l'effort nécessaire pour une éventuelle cryptanalyse.
- Puissance de calcul qui entraine une grande rapidité de traitement .
- Besoins en ressources et mémoire très faibles.
- Flexibilité d'implémentation, cela inclut une grande variété de plateformes et d'applications ainsi que des tailles de clés et de blocs supplémentaires .
- Compatibilité hardware et software, il est possible d'implémenter l'AES aussi bien sous forme logicielle que matérielle.
- Simplicité, le design de l'AES est relativement simple .

Si l'on se réfère à ces critères, on constate que l'AES est également un candidat particulièrement approprié pour les implémentations embarquées qui suivent des règles beaucoup plus strictes en matière de ressources, puissance de calcul, taille mémoire, etc.

C'est sans doute cela qui a poussé le monde de la 3G (3ème génération de mobiles) à adopter cet algorithme pour son schéma d'authentification.

Détails techniques

L'AES opère sur des blocs de 128 bits qu'il transforme en blocs cryptés de 128 bits par une séquence de N opérations ou « rounds », à partir d'une clé de 128, 192 ou 256 bits. Suivant la taille de celle-ci, le nombre de rounds diffère : respectivement 10, 12 et 14 rounds.

Le schéma suivant décrit succinctement le déroulement du chiffrement (Voir figure 3.22). :

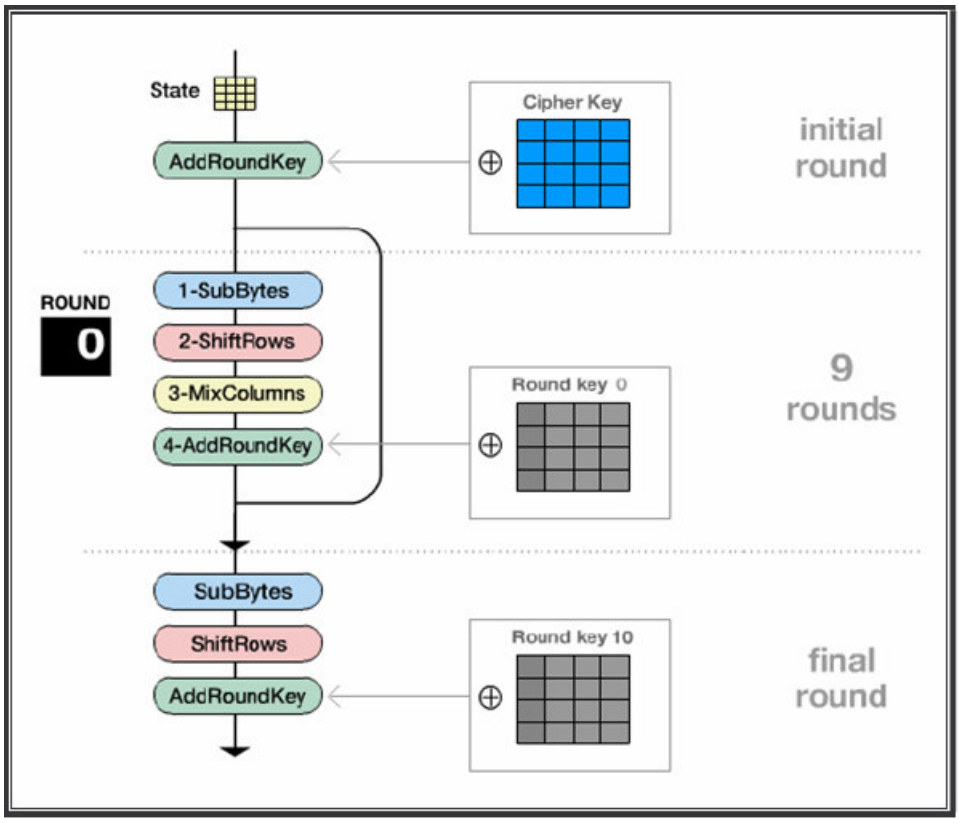


FIGURE 3.21 – le déroulement du chiffrement

SubBytes

La transformation SubBytes est une substitution non linéaire d'octet, elle se fait à l'aide d'une table que l'AES définit, appelée S-Box, qui contient 256 éléments de 8 bits (figure 3.23).

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

FIGURE 3.22 – La table S-Box

Chaque octet de la matrice state est substitué par un octet de la table de la manière suivante :

• Le premier caractère hexadécimal (4 bits de poids le plus fort) de l'élément indique une ligne de la S-Box tandis que le deuxième (4 bits de poids le plus faible) indique une colonne.

• Le byte se trouvant à l'intersection ligne-colonne dans la S-Box est celui qui doit être substitué à celui de la matrice state . La table S-Box est construite de la manière suivante :

1. Initialiser la table S-Box par les valeurs d'octets dans l'ordre croissant ligne par ligne.

La première contient {00}, {01},{02}, ... {0F} ; la deuxième contient {10}, {11}, etc. Ainsi la valeur de l'octet se trouvant dans la ligne x et la colonne y est { xy }.

2. Prendre l'inverse multiplicatif dans GF (2^8) de chaque octet de la table de substitution, l'inverse de {00} étant par convention lui même.

On considère que chaque octet de la table S-Box est constitué des 8 bits notés (b_7 , b_6 , b_5 , b_4 , b_3 , b_2 , b_1 , b_0), pour chaque bit la transformation affine dans GF (2^8) suivante est appliquée :

$b'_i = b_i \oplus b_{(i+4)} \bmod 8 \oplus b_{(i+5)} \bmod 8 \oplus b_{(i+6)} \bmod 8 \oplus b_{(i+7)} \bmod 8 \oplus c_i$

Où c est le $i^{ème}$ bit de l'octet c ayant la valeur {63} ; tel que (c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0) =(01100011) (Voir figure 3.24).

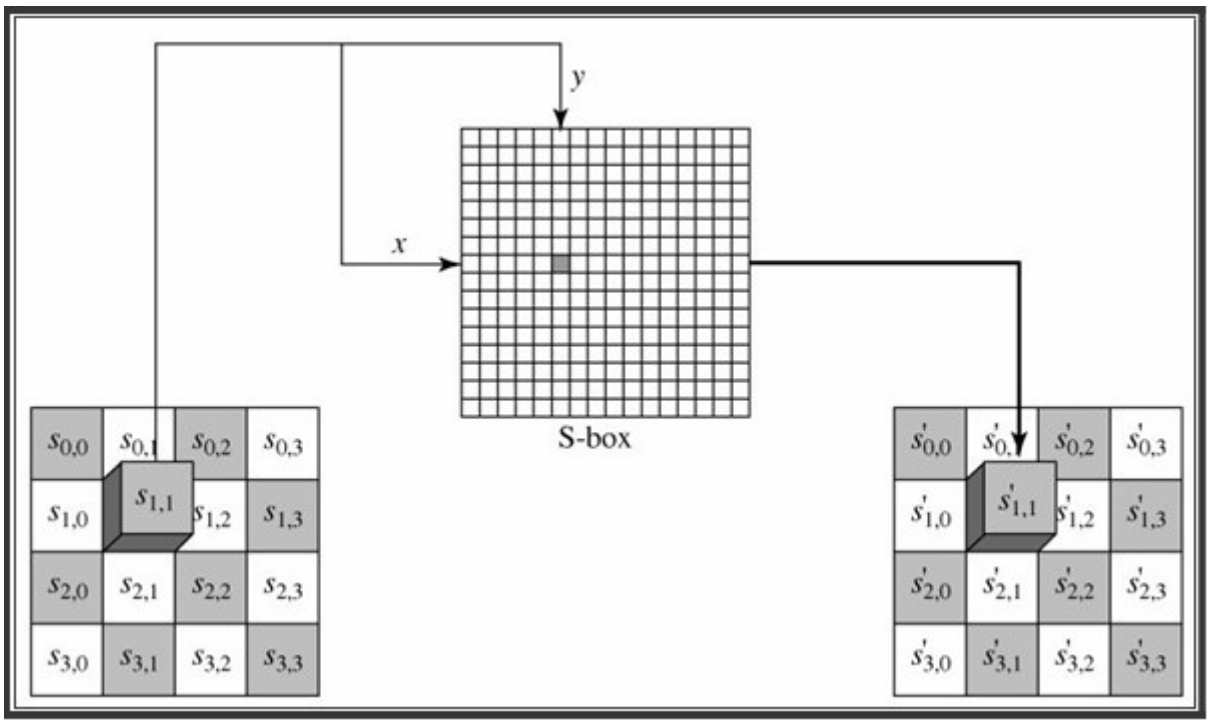


FIGURE 3.23 – Transformation SubBytes

Dans la multiplication matricielle ordinaire, chaque élément de la matrice résultante est la somme des produits des éléments d'une ligne et d'une colonne, pour l'équation (figure 3.24) les sommes sont remplacés par des XOR

ShiftRows

La transformation ShiftRows applique une permutation circulaire vers la gauche sur les trois dernières lignes du bloc (Figure 3.23). Ainsi de faibles changements dans le texte clair impliquent de grands changements dans le texte chiffré.

Les décalages ne modifient pas les valeurs des bytes, mais changent leur ordre seulement. Les décalages se font comme suit :

A La deuxième ligne reçoit une permutation d'un octet.

A La troisième ligne reçoit une permutation de deux octets.

A La quatrième reçoit une permutation de trois octets.

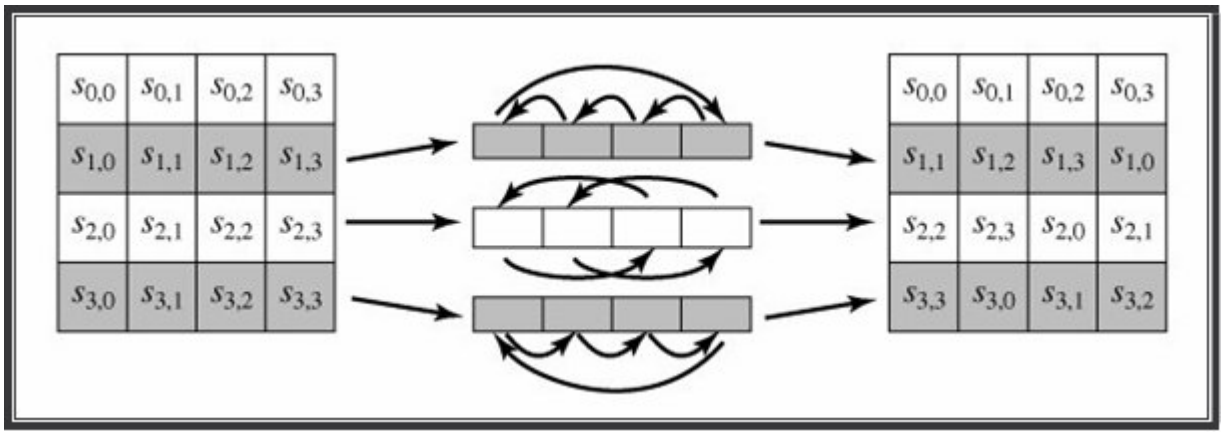


FIGURE 3.24 – Transformation ShiftRows

MixColumn

MixColumn est une fonction qui transforme chaque octet d'entrée en une combinaison linéaire d'octets d'entrée et qui peut être exprimée mathématiquement par un produit matriciel.

Le signe + entouré d'un cercle désigne l'opération de OU exclusif (XOR).

K_s est la$n^{ème}$ sous-clé calculée par un algorithme à partir de la clé principale K.

Le déchiffrement consiste à appliquer les opérations inverses, dans l'ordre inverse et avec des sous - clés également dans l'ordre inverse.

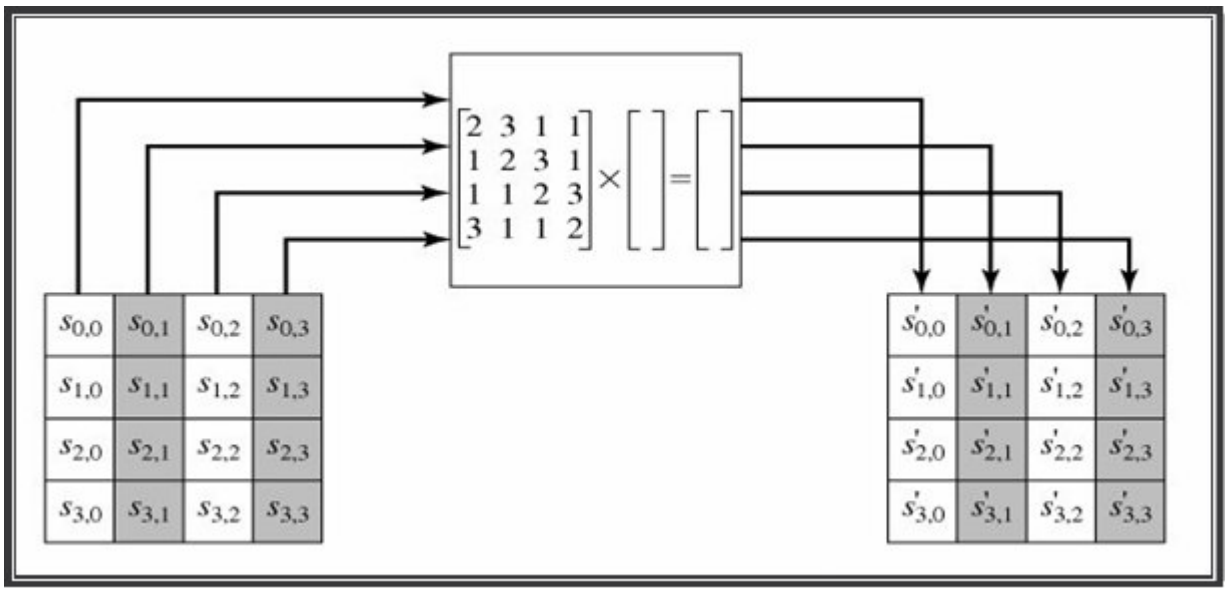


FIGURE 3.25 – Transformation MixColumns

AddRoundKey :

Lors du processus de chiffrement, à chaque Round, une clé (Round key) est extraite de la fonction génération des clés , celle ci est composée de 4 mots de 32 bits. La transformation AddRoundKey consiste à additionner modulo 2 (XOR) les colonnes de la matrice State, et les mots de la clé du Round en cours. Un exemple est donné dans la (Voir Figure 3.26).

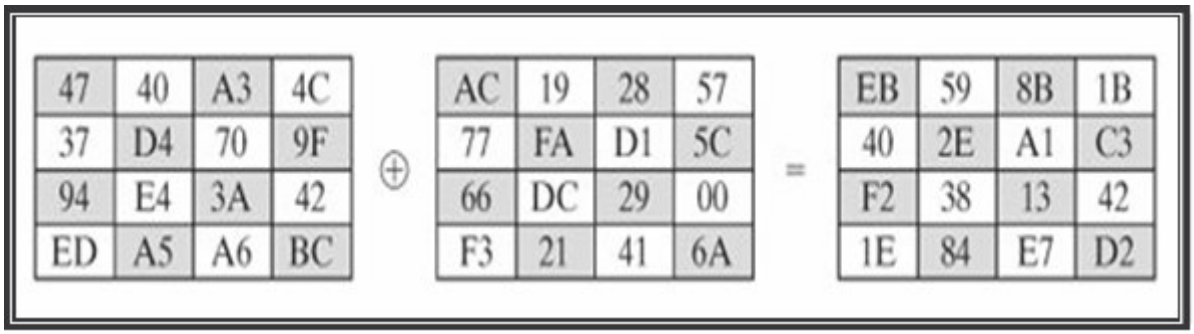


FIGURE 3.26 – Transformation AddRoundKey

En répétant ces quatre transformations 9 fois la deuxième étape est accomplie. La troi-

sième étape (Final Round) sera satisfaite en appliquant les transformations d'un Round de la deuxième étape sauf MixColumns. Une fois les trois étapes sont réalisées, le chiffrement sera terminé .

Avantages et limites

Les principaux avantages sont :

– des performances très élevées,
– la possibilité de réalisation en "Smart Card" avec peu de code,
– la possibilité de parallélisme,
– il ne comprend pas d'opérations arithmétiques : ce sont uniquement des décalages et des XOR,
– il n'utilise pas de composants d'autres cryptosystèmes,
– il n'est pas fondé sur des relations obscures entres opérations,
– le nombre de rondes peut facilement être augmenté si c'est requis,
– il ne possède pas de clés faibles,
– il est résistant à la cryptanalyse différentielle et linéaire.

Il possède pourtant quelques inconvénients et limites :

– le code et les tables sont différents pour le chiffrement et déchiffrement,
– le déchiffrement est plus difficile à implanter en "Smart Card",
– dans une réalisation matérielle, il y a peu de réutilisation des circuits de chiffrement pour effectuer le déchiffrement.

Conclusion

L'algorithme AES présente un niveau de sécurité très élevé, grâce à :

A L'utilisation de la table S-Box qui constitue une difficulté réelle pour les cryptanalystes. A L'opération MixColumns combinée avec ShiftRows fait que, après les nombreux Rounds, tous les bits de sortie dépendent de tous les bits d'entrée. Ce qui rend la cryptanalyse difficile. A L'utilisation des clés secondaires construites par extension de la clé originale, quant à elle, complique les attaques liées à la clé en cassant les symétries.

Annexe 4 « Une brève histoire de l'informatique quantique »

Un mouvement qui accélère Il ne s'agit pas ici de détailler l'histoire de l'informatique quantique[1]. Mais si l'on souhaite sortir de l'utopie, il parait néanmoins nécessaire de montrer sur quelles bases historiques les technologies quantiques s'appuient, les moments qui l'ont faite progresser, et prendre conscience de l'accélération qui s'est opérée ces 10 dernières années [38].

L'informatique quantique commence au début du XX° siècle avec les premiers travaux sur la théorie des quantas, initiée par le physicien allemand Max Planck en 1900. Cette théorie a permis de faire le lien entre la physique classique et la physique quantique. C'est en 1925 que les principes de la mécanique quantique ont ensuite été développés par Albert Einstein, Niels Bohr, Louis de Broglie, Werner Heisenberg et bien d'autres scientifiques.

1935

En 1935, Albert Einstein et deux autres physiciens, Boris Podolsky et Nathan Rosen, publient un article qui décrit une « expérience de pensée » pour démontrer que la mécanique quantique, telle que définie à l'époque, est incomplète. Pour résumer, la théorie expliquait que si l'on produit un électron et un positron[2] intriqués dans une expérience, la mesure d'une propriété de l'électron est immédiatement répercutée sur le positron qui « le sait immédiatement » même s'il est à des millions de kilomètres. Einstein spéculait qu'étant donné que ce principe quantique violait les principes de localité[3] et de réalité[4], et que la nature devant être par hypothèse réaliste et locale, la mécanique quantique devait être incomplète. C'est le paradoxe EPR[5] (Einstein-Podologie-Rose n).

Pour Einstein, cette transmission d'information « plus rapide que la lumière » était inacceptable : il devait exister des variables cachées qui « donnent l'impression » d'une communication immédiate[31].

1964

En 1964, le physicien nord-irlandais John Bell propose le principe d'une expérience qui permet de résoudre ce problème. Il formalise la question par des « inégalités[6] », dites de Bell, qui sont évaluées au cours de l'expérience. Si l'inégalité n'est pas respectée, alors le résultat de l'expérience ne peut pas être expliqué par l'existence de variables cachées, et il faut se résoudre à admettre le caractère non local de la nature qu'Einstein refusait. L'état

1. Pour cela, lire le travail complet d'Olivier Ezratty : https ://www.oezratty.net/wordpress/2018/ebook-pour comprendre- informatique-quantique/
2. Positron : antiparticule, de charge électrique positive, associée à l'électron, de charge électrique négative.
3. Localité : principe selon lequel des objets distants ne peuvent avoir une influence directe l'un sur l'autre.
4. Réalité : pour qu'une grandeur physique soit réelle, il suffit qu'il soit possible de la prédire avec certitude, sans perturber le système.
5. https ://fr.wikipedia.org/wiki/Paradoxe_EPR
6. Les inégalités de Bell sont les relations que doivent respecter les mesures sur des états intriqués dans l'hypothèse d'une théorie déterministe locale à variables cachées.

des technologies de l'époque ne permettra de réaliser cette expérience que dans les années 1980 : c'est un scientifique français, Alain Aspect [7] qui la réalisera et montrera que les inégalités de Bell sont bien violées, confirmant ainsi le caractère non local de la physique quantique, que la mécanique quantique était donc bien complète, et que, par conséquent, l'une des hypothèses de base d'Einstein était fausse. Avec cette expérience, Alain Aspect démontrera que le phénomène d'intrication théorisé par Albert Einstein, mais auquel il ne croyait pas, était valide. Nous verrons plus loin dans ce document [8] que l'intrication fait partie des éléments de base de l'informatique quantique[31].

1981

Les années 1980 sont importantes dans le monde de l'informatique quantique. En effet, en 1981 se déroule la première conférence du MIT (Massachusetts Institute of Technology) sur ce sujet. C'est lors de cette conférence qui rassemblait de nombreux scientifiques et physiciens renommés, que nait l'idée d'encoder de l'information dans les états quantiques de la matière.

Durant cette conférence, le physicien américain Richard Feynman (prix Nobel de physique en 1965) fut le premier à percevoir le potentiel de l'informatique quantique. Les ordinateurs classiques (machines de Turing) n'étant pas assez performants pour simuler les phénomènes quantiques, il suggéra, dans une phrase devenue célèbre, d'utiliser des simulateurs quantiques, plus simples et contrôlables, pour étudier d'autres systèmes quantiques.

C'est la première fois que l'on imagine la possibilité d'un ordinateur quantique ou plutôt de sa simulation ! À partir de ce moment, les travaux vont s'accélérer[31].

1 Freeman Dyson, 2 Gregory Chaitin, 3 James Crutchfield, 4 Norman Packard, 5 Panos Ligomenides, 6 Jerome Rothstein, 7 Carl Hewitt, 8 Norman Hardy, 9 Edward Fredkin, 10 Tom Toffoli, 11 Rolf Landauer, 12 John Wheeler, 13 Frederick Kantor, 14 David Leinweber, 15 Konrad Zuse, 16 Bernard Zeigler, 17 Carl Adam Petri, 18 Anatol Holt, 19 Roland Vollmar, 20 Hans Bremerman, 21 Donald Greenspan, 22 Markus Buettiker, 23 Otto Floberth, 24 Robert Lewis, 25 Robert Suaya, 26 Stan Kugell, 27 Bill Gosper, 28 Lutz Priese, 30 Paul Benioff, 31 Hans Moravec, 32 Ian Richards, 33 Marian Pour-El, 34 Danny Hillis, 35 Arthur Burks, 36 John Cocke, 37 George Michaels, 38 Richard Feynman, 39 Laurie Lingham, 40 Thiagarajan, 41 ?, 42 Gerard Vichniac, 43 Leonid Levin, 44 Lev Levitin, 45 Peter Gacs, 46 Dan Greenberger

FIGURE 3.27 – Physics of Computation Conférence - Endicott House MIT - May 6-8, 1981

7. http ://www.cnrs.fr/fr/personne/alain-aspect
8. Voir le chapitre 5.1. A la base, 3 principes quantiques

1984

Dès 1984, Charles H. Bennett, d'IBM Research, qui avait participé dans les années 70 à l'émergence de la théorie de l'information quantique, et Gilles Brassard de l'Université de Montréal, proposent le premier protocole de cryptographie quantique : BB84, mécanisme d'échange de clés quantiques [31].

1993

En 1993, un groupe international de six scientifiques, dont Charles H. Bennett, confirme les intuitions de la majorité des auteurs de science-fiction en montrant que la téléportation parfaite est « en principe » possible, mais seulement si l'original est détruit (ce qui peut quand même poser problème !)[31].

1995

En 1995, Peter Shor, chercheur en mathématiques appliquées au MIT, démontre que le calcul quantique avec des Qubits permet l'existence d'un algorithme capable de factoriser en un temps record (quelques dizaines de secondes) n'importe quel entier en un produit de deux nombres premiers [9]. En théorie - en pratique c'est une autre histoire - il était donc possible de casser les codes secrets non seulement des banques mais aussi des États et des armées en utilisant l'algorithme de Shor [10].

L'informatique quantique commence alors à intéresser au-delà de la sphère scientifique puisque l'on comprend que la sécurité des systèmes d'information peut être mise à mal. Et l'on voit très rapidement apparaitre de nouveaux acteurs se trouvant hors de la sphère scientifique[31].

1996

En 1996, David Di Vicenzo, chercheur chez IBM, définit les premiers critères permettant d'avoir un processeur quantique :

- les Qubits doivent être intégrables et réalisables en grand nombre,
- il faut disposer de portes quantiques universelles, capables de réaliser tout algorithme,
- il est nécessaire d'avoir une lecture fidèle en une fois,
- il faut pouvoir réinitialiser efficacement chaque Qubit à l'état 0.

Et la même année IBM présente le premier ordinateur quantique à 2 Qubits[31].

9. Par comparaison, en 2010 un nombre codé sur 768 bits a été factorisé par un algorithme s'exécutant une année durant sur 425 ordinateurs classiques à 4 coeurs (record non battu à ce jour !)
10. https ://interstices.info/lalgorithme-quantique-de-shor/

1997

Mais l'ordinateur quantique est extrêmement fragile car de nombreuses erreurs apparaissent dans les calculs en raison de la « décohérence quantique »[11]. Un mécanisme de correction d'erreurs est donc indispensable. En 1997 Alexeï Kitaev, professeur russo–américain de physique à l'Institut de technologie de Californie et chercheur chez Microsoft a eu l'idée de s'inspirer de la topologie, branche des mathématiques qui étudie les objets et leurs propriétés lorsque ceux-ci subissent des déformations, pour proposer une solution à ce problème[31].

2001

En 2001, les chercheurs d'IBM arrivent à factoriser le nombre 15 en utilisant l'algorithme de Shor sur leur machine quantique[31].

2011

À partir de 2011, avec l'arrivée de nouveaux acteurs issus pour la plupart du secteur numérique, tout s'accélère encore. En 2011, la société californienne D-Wave présente le premier ordinateur quantique commercial à 128 Qubits[31].

2012

En 2012, deux physiciens, David Wineland et Serge Haroche, reçoivent le prix Nobel de physique pour leurs travaux sur le contrôle et la mesure des atomes. Le premier a réussi à contrôler l'état quantique d'ions grâce à des photons, et le second a permis d'étudier le phénomène de décohérence quantique en réussissant à mesurer l'information d'un système quantique sans le détruire. Ce phénomène de décohérence quantique était un problème : plus ce temps est grand, plus il est possible d'exécuter un grand nombre de portes logiques quantiques[12]. Mais si ce temps n'est pas suffisant pour exécuter l'ensemble des opérations d'un algorithme, cela ne sert à rien. Cette barrière tombe la même année (2012) quand IBM réussit à exécuter des algorithmes quantiques de manière complète[31].

2015

En 2015 il est démontré que les algorithmes de correction d'erreur fonctionnent et sont utilisables[38].

11. Voir le chapitre 5.2. Le Qubit, unité de base de l'informatique quantique

12. Une porte logique, quantique ou électronique, est la brique de base d'un circuit, électronique ou quantique, élémentaire. En électronique, ces portes sont construites à partir de plusieurs transistors connectés de manière adéquate ; en quantique, elles opèrent sur un petit nombre de Qubits.

2016

En 2016, Microsoft annonce que l'informatique quantique devient une priorité stratégique. Et IBM rend disponible sur un Cloud public le premier ordinateur quantique. À ce jour plus de 100 000 personnes l'ont utilisé et plus de 140 articles ont été publiés à partir de travaux menés sur cette machine[31].

2017

En 2017, Atos lance la commercialisation de l'ATOS QLM (Quantum Learning Machine) permettant de simuler 30 Qubits, Rigetti lance la production de galettes de silicium destinées au calcul quantique, Intel annonce lui aussi la fabrication d'un circuit de calcul quantique à 17 Qubits. IBM réussit à simuler la structure moléculaire de l'hydrure de béryllium (BeH2) et atteint avec un ordinateur à 50 Qubits le seuil théorique de la suprématie quantique[31].

2018

En 2018 Intel dévoile à son tour un calculateur à 49 Qubits, puis Google avec Bristlecone, un processeur quantique de 72 Qubits, et Atos une version de 41 Qubits de l'ATOS QLM[31].

2019

En 2019, IBM dévoile au CES de Las Vegas le premier ordinateur quantique « compact » de 20 Qubits dénommé IBM Q System One. En octobre 2019, même si les résultats sont controversés, Google annonce avoir atteint la suprématie quantique[31].

2020

Au CES 2020, IBM a annoncé la mise en ligne d'une nouvelle machine « Raleigh » dotée de 28 Qubits. L'utopie est finie, le réel s'impose et tous les grands acteurs du numérique qui voient leur écosystème menacé par l'informatique quantique, se lancent dans la bataille[31].