REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Batna 2

Faculté des Mathématiques et d'Informatique

Département d'Informatique

THESE

En vue de l'obtention du diplôme de

Doctorat en Sciences en Informatique

Présentée par

Rima DJELLAB

Cryptographie quantique, Nouvelles approches

Soutenue publiquement le **.....** devant le jury formé de :

Pr. Azeddine BILAMI	**Président**	Prof. Université de Batna 2
Pr. Mohamed BENMOHAMMED	**Rapporteur**	Prof. Université de Constantine 2
Pr. Allaoua CHAOUI	**Examinateur**	Prof. Université de Constantine 2
Pr. Salim CHIKHI	**Examinateur**	Prof. Université de Constantine 2
Dr. Samir ZIDAT	**Examinateur**	M.C. Université de Batna 2

Remerciements

Je loue Dieu le Tout Puissant Allah pour m'avoir accordé patience et persévérance afin de faire aboutir ce travail.

Mes remerciements s'adressent ensuite au Professeur Mohammed Benmohamed de l'université de Constantine 2 pour avoir accepté de suivre ce travail de thèse qui est une continuité de celui que nous avons mené lors de notre Magistère.

J'adresse également mes remerciements au Professeur Azeddine Bilami du laboratoire LASTIC, de l'université de Batna 2 pour m'avoir fait l'honneur de présider le jury de cette thèse, mais aussi aux examinateurs ; Professeurs Allaoua Chaoui et Salim Chikhi du laboratoire MISC de l'université de Constantine 2 et Docteur Samir Zidat du laboratoire LASTIC, de l'université de Batna 2 pour avoir accepté de faire partie du jury et d'avoir consacré de leur précieux temps à la lecture et l'évaluation de mon travail de thèse.

Mes remerciements vont également à Rachid Echahed responsable de l'équipe CAPP du laboratoire LIG de l'université de Grenoble, pour m'avoir accueilli au sein de son équipe, mais également au Docteur Mehdi Mhalla, de la même équipe, pour ses précieuses discussions qui m'ont permis d'aborder le quantique d'une bien différente manière.

Je remercie particulièrement mon amie Dr Souheila Bouam de l'université de Batna 2, pour ses conseils, orientations et encouragements qui ne sauraient venir que d'une sœur et qui boostent à chaque fois qu'on en a besoin.

Je remercie aussi mon amie et collègue Leila Saadi de l'université de Batna 2, pour les moments qu'on a passé ensemble à travailler nos thèses et à s'encourager mutuellement. Qu'elle puisse trouver dans l'aboutissement de ce travail le courage et la motivation pour finaliser le sien.

Enfin, j'aimerai remercier ceux que je ne remercierai jamais assez, les membres de ma famille qui ont fait preuve à mon égard d'une patience et d'un soutien sans faille ; Qu'ils trouvent ici l'expression de ma gratitude et le témoignage de ma grande reconnaissance.

Dédicaces

A mes parents
pour leur patience et soutien inconditionnels

Résumé

Quelque soit l'environnement dans lequel se déroule une communication, la sécurité reste un pilier nécessaire pour le bon déroulement de cette communication. Afin d'assurer un certain niveau de sécurité, la confidentialité est un élément essentiel à assurer. Ce sont les techniques de cryptographie qui l'assurent. La pierre angulaire des techniques cryptographiques est la gestion de clé. Plusieurs protocoles ont été proposés pour assurer une bonne gestion de clé dans un groupe, mais leur sécurité se base sur la complexité computationnelle. La distribution de clé quantique, par abus dite cryptographie quantique, est une primitive cryptographique basée sur les lois de la mécanique quantique, permettant d'établir une clé secrète commune entre les traditionnels correspondants Alice et Bob. Dans cette thèse, nous nous intéressons à la problématique de la distribution de clé dans groupe. Nous proposons pour cela une solution basée sur la distribution de clé quantique, ce qui permet d'exploiter cette même solution dans le contexte de groupe. Nous avançons par la suite la preuve formelle de la solution proposée en utilisant le vérificateur formel PRISM.

Mots clé :

Cryptographie, Cryptographie quantique, BB84, Distribution de clé, Distribution de clé quantique, Quantique, Sécurité.

Abstract

Whatever the environment in which communication takes place, security remains a necessary pillar for the success of this communication. To ensure a certain level of security, confidentiality is essential to ensure. These are the cryptographic techniques that provide it. The cornerstone of cryptographic techniques is the key management. Several protocols have been proposed to ensure good key management in a band, but their security is based on computational complexity. The quantum key distribution, by abuse called quantum cryptography, is a cryptographic primitive based on the laws of quantum mechanics, to establish a shared secret key between the traditional corresponding Alice and Bob. In this thesis, we address the problem of group key distribution. For this we propose a solution based on quantum key distribution, which allows the use of this solution in the group context. We argue later formal proof of the proposed solution using the PRISM formal model checker.

Key Words:

Cryptography, BB84, Key distribution, Quantum cryptography, Quantum key distribution, Quantum, Security.

ملـــخص

مهما كانت البيئة التي تجري فيها الاتصالات، يظل الأمن ركنا أساسيا لنجاح هذا الاتصال. ولضمان مستوى معين من الأمن، فإن السرية ضرورية لضمان ذلك. هذا ما توفره تقنيات التشفير. حجر الزاوية في تقنيات التشفير هو الإدارة توزيع مفتاح التشفير. لذا اقترحت عدة بروتوكولات لضمان إدارة توزيع جيدة في مجموعة اتصال، ولكن أمنها يقوم على التعقيد الحسابي.
توزيع مفتاح الكم، الذي ،في إفراط لغوي، يسمى التشفير الكمي، هو بدائية تشفير تستند إلى قوانين ميكانيكا الكم، لإنشاء مفتاح سري مشترك بين أليس وبوب.
في هذه الأطروحة، نتعامل مع مشكلة توزيع مفتاح التشفير لمجموعة اتصال. لهذا نقترح حلا يستند إلى توزيع المفتاح الكمي، والذي يسمح باستخدام هذا الحل في سياق مجموعة اتصال. نجادل في وقت لاحق دليلا رسميا للحل المقترح باستخدام المدقق نموذج رسمي PRISM.

كلـمـات البـحـث:

الترميز، الترميز الكمي، BB84، توزيع مفتاح التشفير وتوزيع مفتاح التشفير الكمي، الكم ،السلامة.

Publications et contributions

Communications Nationales et Internationales

- **R.DJELLAB**, M.BENMOHAMMED "Verification of A Group Key Distribution Protocol based on QKD", International Workshop on Cryptography and its Applications - IWCA'16 -26 & 27 April 2016, U.S.T.O-MB, ORAN-ALGERIA.
- **R.DJELLAB**, M.BENMOHAMMED " Survey of Key Distribution Issue: From Classical to Quantum Solution", The 2013 International Conference on Artificial Intelligence and Information Technology (**ICA2IT'14**) Ouargla, Algeria, March 10 – 12, 2014
- **R.DJELLAB**, M.BENMOHAMMED " Survey of Key Distribution Issue: From Classical to Quantum Solution", The 2013 International Conference on Security and Management (**SAM'13**) Las Vegas, USA, July 22 – 25, 2013 (accepted)**.**
- **R.DJELLAB**, M.BENMOHAMMED " Distribution Quantique De La Clé De Chiffrement Dans Les WLANs" 2èmes Journées Doctorales en Informatique de Guelma,18 - 19 Novembre 2012
- S.SELMAN, **R.DJELLAB** " Enhanced munitias-based fingerprint authentication system using user's pattern and Shamir secret sharing Scheme" 2èmes Journées Doctorales en Informatique de Guelma,18 - 19 Novembre 2012.
- F.HEDNA, L.GUEZOULI, **R.DJELLAB** " Routage basé Clustering dans les RCSF Analyse des performances" 2èmes Journées Doctorales en Informatique de Guelma, 18 - 19 Novembre 2012.
- **R.DJELLAB** , M.BENMOHAMMED "Securing Encryption Key Distribution in WLAN via QKD", International Conference on Cyber-Enabled Distributed Computing and Knowledge Discover, CyberC 2012, Oct. 10-12, 2012, Sanya, China. Pages: 160-165. Publisher: IEEE Computer Society Washington, DC, USA ©2012, ISBN: 978-0-7695-4810-4 doi>10.1109/CyberC.2012.34
- **R.DJELLAB**, A. ABDENNEBI "Enhancing digital signature schema using fingerprint minutiae point and RSA algorithm", ICCRK'12, July 5-7, Sousse, Tunisia (accepted).
- **R.DJELLAB** , M.BENMOHAMMED "Sécurisation de la Distribution de clé de chiffrement dans les WLANs via la QKD", RSACS'11, 22-23 Juin, Oran, Algérie.
- **R.DJELLAB** , M.BENMOHAMMED "Distribution sécurisée de la clé de chiffrement dans les WLANs via la QKD", WCCCS'11, 16-17 Juin, Rabat, Maroc (Sélectionné Meilleurs Papier)
- **R.DJELLAB** , M.BENMOHAMMED "Secured Scheme for Encryption Key Distribution in WLAN Using Quantum Cryptography", ICN'11, Les Antilles (accepté).
- **R.DJELLAB** , M.BENMOHAMMED "Secured Scheme for Encryption Key Distribution in WLAN Using Quantum Cryptography", ICMOSS'10, 29-31 Mai, Tiaret, Algérie.
- **R.DJELLAB** , M.BENMOHAMMED "Multi-participants key generation in WLAN using QKD", ICAI'09, 15–17 Novembre, Bourdj Bou Ariridj, Algérie. Page: 198, ISBN:978-9947-0-2763-9

- M.BENMOHAMMED, **R.DJELLAB** "Secured Key Distribution in 802.11 via Quantum Cryptography", JGED'09, 20–21 Mai, Annaba, Algérie.
- **R.DJELLAB**, "New Scheme Of Integrating Quantum Key Distribution In 802.11i", IEEE ICMCS'09, 2–4 Avril, Ouarzazete, Maroc. Publisher: IEEE, Pages: 46-50 **ISBN:** 9781424437566 **DOI:** 10.1109/MMCS.2009.5256734

Publications

- **Rima DJELLAB**, Mohamed Benmohammed, 'A Novel Quantum Distributed Key Management Protocol for Ring-organized Group', in Press, International Journal of Information and Computer Security. (DOI:10.1504/IJICS.2017.10004329)
- Khalil Amine, **Rima Djellab**, "Industrial and Urban Applications of Eulerian and Chinese Walks," In: R. Zanjirani Farahani and E. Miandoabchi (eds.), Graph Theory for Operations Research and Management: Applications in Industrial Engineering, IGI Global, Hershey, USA. Forthcoming 2013. (ISBN: 978-146-662-661-4 DOI: 10.4018/978-1-4666-2661-4).
- **R.DJELLAB**, M. BENMOHAMMED "Secured Scheme for Encryption Key Distribution in WLAN Using Quantum Cryptography", MOMA Journal, Vol 01, Issue 01, 2012, pp 25-32 (ISSN: 2253-0665.
- **Rima DJELLAB**, Aicha ABDENNABI, " Enhancing Digital Signature Schema Using Fingerprint Minutiae Point and RSA Algorithm", Journal of Information Security Research, Vol 3, Issue 3, Sep 2012, pp 116-124 (ISSN: 0976-4143).
- **R.DJELLAB**, M. BENMOHAMMED "Enhancing 802.11i key distribution using quantum key distribution", IJARITAC Vol 2, Issue 3, Sep-Dec 2011 pp14-22. (ISSN: 0975-8089. DOI 10.5958/J.0975-8070.)
- **R.DJELLAB**, M. BENMOHAMMED "Sécurisation de la Distribution de clé de chiffrement dans les WLANs via la QKD", Vol 1, Num 1, Numéro special RESINTTICO Juin 2011, pp 70-75.

Sommaire

Table des illustrations

Table des Tableaux

Introduction générale

Communiquer a toujours été un besoin pressant chez l'homme, ce dernier a utilisé au fur et à mesure des moyens différents qui ont évolué au cours du temps afin de coder l'information qu'il voulait transmettre à ses semblables. Usant alors de la mimique et de la gestuelle, des graphes, des schémas des pictogrammes et de l'écriture les moyens de communication ont alors été de plus en plus sophistiqués. Quelque soit la forme sous laquelle était transmise l'information, il s'agissait d'une forme de code permettant, à un groupe identifié, de comprendre ladite information. Plus l'information était importante plus le code se faisait *complexe*, afin que seule une personne donnée, ou un groupe bien déterminé, puisse interpréter l'information ainsi transmise. Ce besoin de communiquer est aujourd'hui toujours présent et se fait encore plus important non seulement parce que la masse d'information qui circule aujourd'hui est conséquente, mais aussi pour son importance qui requiert un certain niveau de sécurité, ainsi e-commerce, e-health, visio-conférence, militaire…sont toutes des applications où l'information requiert justement un niveau élevé de sécurité.

La sécurité de l'information est un agencement de plusieurs piliers dont les plus importants on cite : la confidentialité, l'authentification, l'intégrité et la non-répudiation. La confidentialité, étant le fait que seul un groupe ou une entité autorisé puisse accéder à une information. Cela peut être assuré par des méthodes et des techniques, qu'elles soient mathématiques ou mécaniques, permettant de transformer l'information sous une autre forme inintelligible, et que seule l'entité pour laquelle elle était dédiée pourrait restituer la forme initiale. Ce sont les techniques de cryptographie.

La cryptographie, ou le chiffrement de l'information, ou encore la science du secret, est une branche de la cryptologie. Elle se définit comme étant la transformation d'une information sous une autre forme en utilisant un algorithme et une information supplémentaire dite clé de cryptage.

Actuellement, les techniques de cryptographie se classent en deux grandes familles en se basant sur le nombre de clés utilisées. On parlera de cryptographie symétrique, si une seule et même clé est utilisée pour le chiffrement et le déchiffrement. Par contre, si deux clés sont utilisées distinctement, l'une pour le chiffrement, l'autre pour le déchiffrement on parlera de cryptographie asymétrique.

Pour le premier cas, il faut s'assurer que les participants, traditionnellement appelés Alice et Bob, ont la même clé afin qu'ils puissent s'échanger l'information chiffrée. Le comment de l'échange de la clé symétrique entre Alice et Bob constitue en soit une problématique pour laquelle plusieurs solutions, non totalement fiables, ont été proposées. Parmi les solutions on cite l'entité tierce, qui à son tour doit être de confiance, l'échange de la clé via une autre clé, et là en revient à la case départ…et bien d'autres solutions. Cette problématique en est une qui a toujours suscité l'intérêt des chercheurs dans le domaine, jusqu'à ce que un autre type de solution, basée sur les problèmes mathématiques difficiles ait été proposé. C'est la cryptographie asymétrique. La cryptographie asymétrique se base donc sur des problèmes mathématiques complexes comme la factorisation des grands nombres pour pouvoir échanger deux clés. Une des deux clés, souvent il s'agit de celle publique, est utilisée pour le

chiffrement alors que la seconde, privée, est utilisée pour le déchiffrement. L'inconvénient d'une telle solution est qu'en théorie rien n'empêche l'opération inverse de factorisation sinon l'aspect pratique de la chose. La loi de Moore, elle, prédit une avancée en termes de puissance des ordinateurs tous les 18 mois, il convient alors de s'assurer que les solutions actuelles sont bel est bien sécurisées et que leur usage, surtout dans certains domaines dits critiques, ne représente aucun risque si une puissance de calcul assez importante, répondant à la loi de Moore, serait en possession d'un espion. A titre d'exemple, le RSA-768 (232 chiffres décimaux) a été cassé après 2 ans et demi de calcul suite aux travaux de l'équipe CACAO de l'Inria Nancy et ses partenaires suisses, japonais, hollandais et allemands. Les calculs ont nécessité une infrastructure Grid'5000 reliant 1544 machines (5000 cœurs), 5 Tera-octets pour copies de sauvegarde et espace de travail [1]. Ce qui revient à dire que la résolution de tel problème mathématique n'est difficile que pour une capacité de calcul actuelle et que si un calculateur puissant existait, il serait facilement possible de retrouver les clés de chiffrement/déchiffrement, tel est le cas du calculateur quantique qui marque un nouveau record en factorisant le nombre 56,153 avec seulement 4-qubits [2], ou encore le quantum annealer de l'entreprise canadienne D-Wave à 1000 qubits présenté en 2007 et qui a été loué à la NASA, Google et Lockheed Martin de l'armement [3].

Il est vrai que la mise sur le marché d'un ordinateur quantique n'est pas chose facile du point de vue technologique mais aussi financier. Néanmoins, une autre application des lois de la mécanique quantique s'avère moins couteuse et bien adaptée à la problématique de la distribution de clé classique, il s'agit de la distribution de clé quantique.

En se basant sur les principes de la mécanique et de la physique quantique, à savoir la superposition, l'intrication, le non-clonage et bien d'autres, Charles Bennett et Gilles Brassard ont été les premiers à proposer une solution de distribution de clé quantique [4]. Ainsi Alice code l'information sous forme d'états quantique, qu'elle envoie à Bob, qui à son tour tente de les mesurer correctement. Cela lui réussit une fois sur deux. Les états aussi bien que les valeurs d'information ne correspondant pas à des mesures correctes seront écartés, alors que celles mesurées correctement constitueront la clé quantique ainsi échangée. D'autres part, si lors des échanges, l'espion, traditionnellement appelée Eve (pour eavesdropper), s'immisce lors des échanges cela ne fera qu'augmenter le nombre d'états faussement mesurés qui seront automatiquement annulés par la suite.

La solution de Charles Bennett et Gilles Brassard, baptisée BB84, et qui au départ n'était valable que pour l'échange de clé à travers une distance de quelques centimètres, est actuellement adoptée par plusieurs firmes internationales comme MagiQ[1] qui offre des solutions adaptées tel que QPN-8505 un système de cryptographie quantique et Q-Box un système de distribution de clé quantique. En plus de MagiQ, le leader suisse IdQuantique[2] offre quant à lui des solutions de cryptographie quantique, de génération de nombres aléatoires et distribution de clé quantique.

En fait, dans le contexte de la distribution de clé quantique, il existe deux familles de protocoles de distribution de clé. Les protocoles P&M pour *Prepare and Measure* et les protocoles E-B pour *Entaglement-Based* [5]. La seconde famille de protocole nécessite l'échange de N systèmes bibartites

[1] MagiQ : organisation de recherche et de développement offrant des solutions de sécurité adaptées. Site web : http://www.magiqtech.com/Products.html

[2] IdQuantique : IDQ compagnie suisse leader dans le domaine de cryptographie quantique, la génération et distribution de clé quantique. Site web : http://www.idquantique.com

entre Alice et Bob alors que ce n'est pas le cas dans les protocoles de la première famille qui sont plus faciles à implémenter.

Nous nous somme intéressés exclusivement à la première famille de protocole de distribution de clé quantique, en l'occurrence les P&M, où Alice prépare N états et les envoie à Bob à travers un canal quantique et que Bob tente de les mesurer correctement, cas du BB84. Ceci dit, nous exploitant l'idée d'un tel protocole dans le contexte d'un groupe de participants, car rares sont les protocoles proposés dans ce sens, seuls Metwaly *et al* dans [6] et [7] proposent une solution rapprochée dans le sens où il s'agit de solution, respectivement, centralisée et décentralisée, de distribution de clé quantique dans un groupe.

La solution que nous proposons est classée dans la troisième famille de protocole de distribution de clé, à savoir les protocoles par accord. Combinant à la fois le principe des protocoles par accord et la sécurité offerte par la distribution de clé quantique, tous les membres d'un groupe participent à l'élaboration d'une clé finale en combinant par des XOr successifs des grains de clés chiffrés via des clés quantiques intermédiaires. La solution ainsi bénéficie de trois avantages non négligeables : 1- la sécurité des clés quantiques dont il n'est plus question de la prouver ; 2- l'utilisation du XOr imitant le One Time Pad qui est le seul système de chiffrement mathématiquement prouvé sûr car la clé n'est n'utilisée qu'une seule fois ; et 3- Tous les membres participent à l'élaboration de la clé, ce qui fait qu'un espion ne pourra obtenir qu'une partie de la clé, s'il réussi à casser le OTP chose qui n'est pas évidente.

Afin de valider la solution proposée, et vue la non disponibilité de simulateurs dédiés, nous avons opté pour la validation formelle, d'autant plus que la validation formelle permet une exploration exhaustive de tous les états de la solution proposée, décrite sous forme de modèle, vérifiant ainsi une/des propriétés énoncées dans un langage spécifique. Une exploration exhaustive permet ainsi de déterminer automatiquement le/les cas où le modèle ne répond pas aux propriétés, ce qui ne peut être réalisé en simulation.

Pour notre cas nous avons opté pour le vérificateur formel probabiliste PRISM. Un vérificateur formel automatique qui a déjà donné ses preuves dans le contexte de la vérification de protocoles quantiques [8] [9] [10].

Organisation de la thèse

Afin de présenter notre travail, le présent manuscrit est organisé comme suit :

Chapitre 1 : Où nous reviendront sur des notions élémentaires de la sécurité de l'information, des concepts généraux et des mécanismes permettant d'assurer la sécurité de l'information dans le monde classique, nous y exposerons également la cryptographie symétrique et asymétrique ainsi que les inconvénients relevés pour chaque type ;

Chapitre 2 : Traite de la problématique majeure de la cryptographie classique, à savoir la distribution de clé. Nous exposerons les différentes catégories de solutions proposées pour résoudre cette problématique ;

Chapitre 3 : Avant d'exposer la solution quantique, nous nous devons de comprendre d'abord les principes de base sur lesquels est fondé le monde quantique. Des principes de la physique et de la mécanique quantique, comme le qubit, la superposition, l'incertitude, l'enchevêtrement…seront

abordés dans ce chapitre. Nous y exposons également quelques applications réelles, preuves que le quantique n'est pas seulement un concept théorique ou tout au plus des expériences dans les grands laboratoires ;

Chapitre 4 : Est consacré à la solution de distribution de clé quantique qui sera explorée en détails. Les différentes étapes constituant le protocole de distribution de clé quantique le plus connu, le BB84, seront détaillées, mais également un survol d'autres protocoles également connus ;

Chapitre 5 : Constitue l'essentiel du travail, où nous expliquons la solution proposée en détail, ainsi que la preuve de sécurité de cette dernière via le vérificateur de modèle PRISM. Nous exposerons les deux propriétés du modèle et nous commenterons également les résultats ainsi obtenus montrant l'efficacité de la solution.

Partie 1 : Etat de l'art

Chapitre 1 : Sécurité de l'Information : Concepts & Mécanismes

« Être ignare, ne pas vouloir apprécier les dangers, persister à les ignorer sont des attitudes blâmables. »
Christian Queinnec

Introduction

La sécurité de l'information transmise entre deux protagonistes a depuis longtemps suscité l'intérêt, et plus d'une technique ont été inventées pour y parvenir. Aujourd'hui, avec l'avancée technologique, différents supports d'information existent, et une panoplie de structures permet le partage de l'information, cela a permis une large diffusion de l'information. Le révère de la médaille est que l'information est plus vulnérable, il y a plus de risque de perte d'authentification, d'intégrité,…Des questions importantes doivent être alors abordées : Comment pouvoir préserver la sécurité de l'information ? Comment serait-il possible de déterminer que tel ou tel autre auteur est bien celui ayant produit telle ou telle autre information ? L'information reçue est bien celle envoyée...

Des techniques ont vu le jour, allant de l'utilisation des mots de passe à la cryptographie quantique, en passant bien entendu par la cryptographie classique, les certificats et les techniques d'authentification…toutes, complémentaires les unes aux autres, et dont le but est d'assurer la sécurité de l'information, puisque à titre d'exemple, un chiffrement de l'information sans authentification ouvre une brèche sécuritaire et peut conduire à une usurpation d'identité ; d'un autre côté une authentification sans chiffrement laisse la porte grande ouverte au vol de l'information.

Bien entendu, la sécurité de l'information ne dépend pas seulement des techniques mises en place pour y aboutir, car le comportement humain y contribue largement. Un comportement inadéquat remettrait en cause les techniques de sécurité les plus sophistiquées et les rendraient parfaitement vaines [11].

Aujourd'hui, avec le développement technologique, et les différentes applications qui sont apparues, le besoin en sécurité de l'information se fait sentir de plus en plus. D'autant que notre vie quotidienne ; personnelle aussi bien qu'économique, dépend de ces applications.

Dans ce chapitre il est question d'aborder les notions élémentaires de la sécurité ainsi que les techniques utilisées pour assurer cette dernière.

1. Sécurité et le besoin de la sécurité

Le besoin de sécuriser l'information c'est fait sentir de plus en plus à force que de nouvelles applications liées au développement technologique apparaissaient : e-learning, e-commerce, e-health…sans oublier la protection de la propriété intellectuelle, la politique, l'armement …tout cette panoplie d'applications et la diversité d'information qui y est manipulée a fait qu'un groupe de participants à une communication veulent maintenir les informations qui circulent au sein du groupe à l'abri de tout intrus pouvant représenter une éventuelle menaces. Une menace qui se traduirait par un vol ou un espionnage, modification et/ou falsification de l'information,…les conséquences, humaines et/ou matérielles, peuvent être alors très lourdes.

1.1. Définitions

Avant de s'attarder sur les menaces et les techniques de sécurité, nous allons d'abord revoir les définitions de deux concepts clés, à savoir *la sécurité informatique* et *l'information.*

En fait, le concept de *sécurité informatique* est défini comme suit [12] [13]: *La protection accordée à un système d'information automatisé afin d'atteindre les objectifs applicables de préservation de l'intégrité, la disponibilité et la confidentialité des ressources du système d'information (comprenant le matériel, les logiciels, les firmware, informations/données et les télécommunications).*

Quant au concept d'*information*, il est à préciser qu'il peut prendre plusieurs formes, dépendant des besoins de la situation de communication.

La RFC 2828 définit l'*information* comme étant *tous faits et idées qui peuvent être représentés (codés) sous différentes forme de donnée* ; aussi la *donnée* est *l'information codée sous une certaine représentation physique, généralement sous forme de séquence de symboles ayant un sens ; spécialement en ce qui concerne l'information qui peut être manipulée et produite par un ordinateur.*

Cependant, dans la littérature du domaine, et suivant la définition de la sécurité informatique introduite en haut, aucune distinction n'est faite entre *information* et *donnée* [12].

On optera pour les besoins du sujet pour la définition donnée dans [14] : *L'information est toute quantité compréhensible*. Nous considérerons alors que sécuriser l'information revient, dans un premier temps, à cacher cette information, et à l'écrire sous une forme *incompréhensible* pour des entités non-autorisées.

La sécurité de l'information est ainsi élaborée contre tout changement ou lecture non-autorisé. Selon Friedrich L. Bauer [15], Il faut distinguer entre deux types de sécurité, la sécurité inconditionnelle, et la sécurité computationnelle. La sécurité inconditionnelle est liée à la puissance de calcul de l'adversaire. Dans ce cas de figure, même si l'adversaire a une puissance de calcul infinie, la sécurité de l'information devra toujours être assurée. A la différence de la sécurité computationnelle, qui elle, est basée sur certaines hypothèses concernant l'impossibilité pratique, et non théorique, de calculer certaines opérations, tout en utilisant les meilleures méthodes possibles.

1.2. Les attaques

Malgré les avancées perçues dans le domaine de la sécurité, certaines brèches subsistent encore et la sécurité doit être montée pour renforcer la vulnérabilité envers des menaces d'attaques [16] que peuvent montrer nombres des protocoles de communication. En fait, il est possible de distinguer entre deux types d'attaques [17] :

1.2.1.Attaques passives

Une attaque passive est une attaque où l'adversaire n'applique aucune modification aux informations. On distinguera entre observation et accès non autorisé:

Observation non autorisée : Où un observateur indiscret observe seulement des informations (chiffrées ou non) qui transitent sur un réseau par exemple;

Accès non autorisée : Où l'intrus lit des informations mémorisées sur un ordinateur, ou écoute un trafic sur un réseau.

1.2.2.Attaques actives

Dans une attaque de ce type par contre, l'attaquant modifie les informations, on distinguera également plus d'un cas :

Le contrôle non autorisé d'un système ; Où l'attaquant prend totalement le contrôle d'une machine ;

La modification de l'information ; que ce soit au niveau d'une machine ou lors de sa transmission ;

L'accès à des services/ressources (e.g temps de calcul, logiciel,…) auxquelles l'attaquant n'a pas le droit ;

Le refus de service (*Denial-of-services*) aux utilisateurs légitimes qui en ont normalement le droit.

La sécurité de l'information ne peut être réduite à des aspects purement techniques, liées le plus à l'informatique, qui en constituent certes une bonne partie, mais pas la totalité de la problématique. En effet, d'autres aspects rentrent en jeu, on citera les aspects juridiques, sociaux, ergonomiques, et même psychologiques et organisationnels [11]

1.3. Challenges de la sécurité

Sécuriser l'information contre les différents types d'attaques que l'on vient d'introduire, est une opération fastidieuse et non pas des moindres. En effet, il ne suffit pas de transformer l'information en une autre inintelligible, mais il s'agit de mettre en place des mécanismes associés à des buts bien définis. Et même si les buts de la sécurité sont assez clairs et précis (confidentialité, authentification,…), les mécanismes permettant de les atteindre ne le sont pas pour autant. Dans d'autre contexte il s'agit même de combiner différents mécanismes afin d'aboutir à un certain niveau de sécurité.

Autant de challenges en face des développeurs et spécialistes en matière de sécurité ; William Stallings les énumère dans [12] comme suit :

1- Les mécanismes à mettre en place pour atteindre les buts de la sécurité, peuvent être d'une complexité telle, qu'il faut un raisonnement assez subtil pour les comprendre.
2- En élaborant un mécanisme ou un algorithme de sécurité, on doit prendre en compte les éventuelles attaques. Ces dernières aboutissent juste en appréhendant le problème de sécurité d'une manière différente.
3- Les mécanismes de sécurité sont complexes, et ce n'est pas en allant du simple but défini et pris indépendamment que surgisse cette complexité. Tout au contraire, c'est en prenant en considération les différents aspects d'une éventuelle attaque que la combinaison de mécanismes donne naissance et sens à une telle complexité.
4- Ayant un bon nombre de mécanismes, il est nécessaire de décider à quel(s) niveau(x) il faudrait implémenter tel ou tel autre mécanisme.
5- Un mécanisme de sécurité n'est pas un simple algorithme ou protocole, et requiert généralement la connaissance, *a priori*, d'une information secrète (une clé de chiffrement par exemple), ce qui pose une autre problématique, celle de la création, la distribution et la sécurité même de cette information.
6- La sécurité informatique, et éventuellement celle des réseaux informatique, peut être vue comme une sorte de bataille, où les espions essayent de trouver des failles dans le système alors que les designers essayent de leur part de combler ces failles. Les espions partent en léger avantage, car il suffit de leur part de trouver une seule brèche sécuritaire, alors que les designers eux doivent retrouver et corriger toutes les failles.
7- L'utilisateur, ainsi que le gestionnaire d'un système ont une tendance naturelle de bénéficier de l'investissement sécuritaire avant même qu'une faille sécuritaire n'arrive.
8- La sécurité nécessite une gestion régulière, et même permanente, chose difficile à court-termes aujourd'hui dans les environnements surchargés.
9- La sécurité est encore trop souvent intégrée au système après sa conception au lieu d'être considérer comme partie intégrante durant l'étape même de la conception.
10- Plusieurs utilisateurs, et même des administrateurs, voient la sécurité comme obstacle à une utilisation efficace et conviviale d'un système ou d'une information.

Tant de chalenges qui forment un réel défit pour les développeurs, afin d'aboutir à un niveau de sécurité que l'on souhaiterait être parfait, mais qui peut être sémantique ou polynomiale [18]. En effet, les niveaux de sécurité varient entre :

1- Sécurité parfaite : Un schéma est dit d'une sécurité parfaite ou ayant une sécurité théorique, si un adversaire ayant une puissance de calcul infinie, et disposant du texte chiffré, ne peut avoir aucune information concernant le texte en clair.
2- Sécurité sémantique : La sécurité sémantique est telle que seul un adversaire d'une puissance de calcul polynomiale bornée puisse accéder à l'information en clair.
3- Sécurité polynomiale : Un schéma est dit de sécurité polynomiale, ou encore *indistingable*, si un adversaire ayant une fonction de chiffrement publique, ne peut deviner un bit b d'une chaine chiffrée C avec une probabilité supérieure à 0,5.

2. Mesures de sécurité

Pour assurer la sécurité de l'information, il ne suffit pas seulement d'assurer l'aspect technique, mais il faut aussi assurer l'aspect humain. En effet, il est inutile d'avoir un arsenal de techniques de sécurisation si l'élément humain n'est pas pris en charge, ce qui constitue une éventuelle menace venant de l'intérieur, chose qui peut être plus dangereuse encore.

2.1. Mesures humaines

La sécurité de l'information nécessite toute une panoplie de techniques à mettre en place pour éviter au mieux de mettre en péril tout un système d'information.

Avant de passer aux techniques proprement dites, il est à signaler que la sécurité de l'information a, aujourd'hui, un volet « humain » à prendre au plus grand degré d'importance. En effet, à quoi serviraient les meilleurs techniques si l'utilisateur même de l'information la mette en péril via des fausses manœuvres ou même des manœuvres mal intentionnées.

Dans [11] les aspects organisationnels de la sécurité sont bien détaillés. On y mentionnera alors le comportement à adopter vis-à-vis des utilisateurs inexpérimentés qui, selon Marcus J. Ranum (inventeur du par-feu), devraient être *abandonnés*. Selon Marcus J.Ranum toujours, *il serait inutile, voire nuisible, d'éduquer les utilisateurs du SI à la sécurité : son argument est que les utilisateurs incapables de maîtriser suffisamment leurs ordinateurs, notamment en termes de mesures de sécurité, sont condamnés à être expulsés du marché du travail, et qu'il ne faut rien faire pour les sauver*. [11]

D'autre part, la frontière d'un système d'information est devenue de plus en plus *perméable* avec les nouvelles technologies. La mobilité des utilisateurs du système, et donc des nœuds pouvant en constituer l'infrastructure, rajoute encore plus à la complexité de délimiter les frontières, et donc de gérer à bien la sécurité du tout. Un deuxième dispositif, est à prendre en considération afin d'assurer la sécurité de l'information, il s'agit de l'*externalisation* face à la *dépérimétrisation*[3]. Une solution proposée par la BP (***B**ritish **P**etroleum*) [11] suivant laquelle une société n'aura plus à se soucier de la sécurité des postes de travail des clients, et même d'une partie des employés, ces derniers n'ayant accès qu'à une partie de l'information, celle dont ils ont besoins.

Cela rejoint en partie la première solution (celle de Marcus J.Ranum) où l'on se souci peu des utilisateurs inexpérimentés, et ouvre également la voie à une autre, celle de la sauvegarde de l'information.

En effet, même si une partie des utilisateurs n'ont accès qu'à une partie de l'information (celle dont ils ont besoin et sur demande) un système d'information n'est pas à l'abri de quelques défaillances, il est alors intéressant, même important, de réaliser des sauvegardes régulières sur des sites éloignés. NAS (***N**etwork **A**ttached **S**torage*) et SAN (***S**torage **A**rea **N**etwork*) sont des techniques modernes permettant une duplication de données à distance, en temps réel ou à intervalles rapprochés [11]. Bien entendu, des testes sont à prévoir de temps à autre (rechargement et restauration de données

[3] Dépérimétrisation : dissolution du périmètre de sécurité.

stockées, redémarrage d'une application à distance…) cela permet en partie de vérifier la disponibilité des données.

2.2. Mesures techniques

Par le terme « mesures techniques » on fait référence aux mesures où l'homme n'intervient nullement ou presque, car c'est bien l'homme qui mettra en place ces mesures là.

Ces mesures sont mises en place dans le but d'assurer les quatre piliers de la sécurité, reconnus dans la littérature du domaine, en l'occurrence la confidentialité ; l'intégrité ; l'authentification et la non-répudiation.

- Confidentialité : il s'agit de *la capacité à garder secrètes des informations avec accès aux seules entités autorisées* [17]. Dans un autre contexte, la confidentialité peut aussi renfermer le cas où les individus contrôlent par qui, les données qui leur sont liées, peuvent être collectées, enregistrées, et par qui et à qui ces données peuvent être divulguées [12].
- Intégrité : il s'agit là de s'assurer que la donnée n'a pas changé durant sa transmission d'un point A vers un (ou plusieurs points) B. Autrement dit que l'on doit *garantir que l'information n'est aucunement modifiée excepté par une action volontaire et autorisée* [17]. L'intégrité peut aussi viser le système, dans ce cas de figure il s'agit de s'assurer que ce dernier exécute ses fonctions de façon impartiale au-delà de toute manipulation non-autorisée délibérée ou non [12].
- Authentification : il s'agit de pouvoir vérifier que la donnée provient bien de la source qui prétend l'avoir envoyer [16].
- Non répudiation : le but ici est qu'un expéditeur ne puisse nier l'envoi d'une donnée. Dans [14] la non répudiation couvre un champ plus large que le simple envoi de donnée, il s'agit de prévenir le cas où une entité dénie tout engagement ou action préalable.

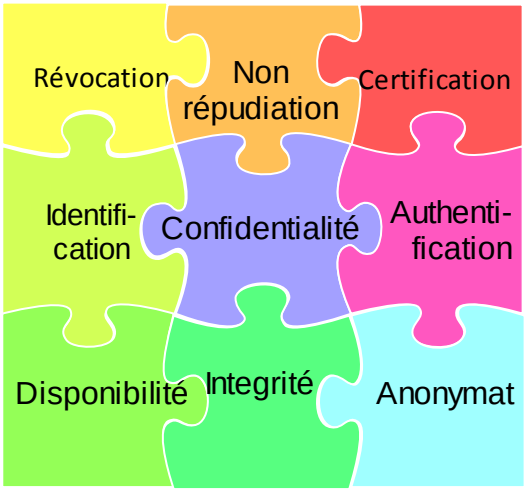


Figure 1 Principaux objectifs de la sécurité de l'information.

Plusieurs techniques ont été mises en place pour assurer ces quatre piliers et bien d'autres aussi résumés dans le tableau suivant :

Objectif	Signification
Confidentialité	L'information n'est accessible que pour le/les destinataire(s) légitime(s)
L'intégrité des données	L'information n'a pas été altérée par une partie non autorisée, ou de manière inconnue.

Identification	Corroboration de l'identité d'une entité (i.e : une personne, un ordinateur, ou une carte de crédit)
Authentification	Vérification de la source de l'information.
Signature	Moyen par lequel une information est liée à une entité.
Autorisation	Délivrer, à une autre entité, la permission officielle de faire ou être quelque chose.
Validation	un moyen d'offrir l'opportunité à une autorisation d'employer ou manœuvrer une information ou des ressources.
Control d'accès	Restriction de l'accès à des ressources, à des entités privilégiées.
Certification	Approbation d'information par une entité de confiance.
Timestamping	Enregistrant la période de la création ou l'existence d'information.
Témoignage	Vérifiant la création ou l'existence d'information par une entité autre que le créateur.
Accusé de réception	Reconnaissance que l'information a été reçue.
Confirmation	Reconnaissance que le service a été fourni.
Propriété	Un moyen de fournir à une entité le droit légal d'employer ou transférer une ressource à d'autres entités.
Anonymat	Cacher l'identité d'une entité impliquée dans un certain processus.
Non répudiation	Empêchant le démenti des engagements ou des actions précédents.
Révocation	Rétraction de certification ou d'autorisation.

Tableau 1 Objectifs de la sécurité [14].

Les techniques utilisées pour assurer les objectifs cités précédemment vont de la plus simple à la plus complexe. On parlera alors de chiffrement, signature numérique, de tatouage et d'empreinte digitale, de certificats …

La majorité de ces techniques utilisent une information particulière, en l'occurrence la « clé », à cet effet, et vue l'importance du concept, nous l'introduisons tout d'abord avant d'aborder plus loin les techniques proprement dites.

2.2.1.Clé et principe de KERCKHOFFS

Au départ, la sécurité de l'information dépendait des techniques de chiffrement utilisées pour *cacher* l'information. Ces techniques se résumaient en quelques manipulations et algorithmes, et se compliquaient au fur et à mesure qu'elles étaient découvertes. Il s'avère par la suite que la sécurité de l'information, ne dépendait pas seulement de ces dits algorithmes, qui se devaient d'être secrets, mais plutôt d'une information particulière qui contribue à la transformation de l'information dans le but de la cacher. Il s'agit de la clé.

L'importance de la clé dans le processus de chiffrement est renforcée par le principe de KERCKHOFFS. Ce dernier, introduit par Auguste KERCKOFFS en 1883 [19] [14]; stipule que la sécurité d'un système ne doit pas dépendre seulement de l'algorithme de chiffrement, mais plutôt du choix de la clé de chiffrement. En fait, KERCKHOFFS énumérait un certain nombre de conditions, qui sont au nombre

de six (06) [20] [14], et que doit remplir un système de cryptographie opérationnel. La condition portant sur l'importance de la clé faisant partie de cette énumération.

Les six conditions dans leur version originale sont comme suit :

1° Le système doit être matériellement, sinon mathématiquement, indéchiffrable ;

2° Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;

3° La clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants ;

4° Il faut qu'il soit applicable à la correspondance télégraphique ;

5° Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes ;

6° Enfin, il est nécessaire, vue les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.[4]

2.2.2.Cryptographie et Chiffrement

Avant de passer au chiffrement, il est important de rappeler l'importance de la cryptographie, qui découle directement d'une science plus large, la cryptologie. En effet, la cryptologie renferme deux grandes disciplines scientifiques : la cryptographie et la cryptanalyse.

La cryptanalyse est la science qui traite de l'opération inverse de la cryptographie. En effet, pour Friedrich L. Bauer dans [15] il s'agit de pouvoir reconstruire plus ou moins le texte en clair sans pour autant avoir accès à la clé de chiffrement. Il est possible de distinguer entre une cryptanalyse passive, où l'attaquant ne fait que « lire » les messages chiffrés sans y introduire une quelconque modification. Si par contre, il y a changement de contenu, retarder l'arrivée du message,…dans ce cas de figure il s'agit le plus d'une cryptanalyse active.

La cryptographie quant à elle est reconnue comme étant *« l'art de sécuriser»* ; Dans [14] elle est définie comme étant *l'étude des techniques mathématique liées aux aspects de la sécurité de l'information, tel que la confidentialité, l'intégrité des donnée, l'authentification des entités et de l'origine des données*; Selon Friedrich L. Bauer dans [15] *La cryptographie est la discipline qui permet d'écrire un message sous forme d'un texte chiffré, usuellement via une transformation appliquée sur un texte en clair en utilisant une clé, dans le but de protéger un secret contre des adversaires, des intercepteurs, des intrus, des indiscrets, des espions, ou simplement contre des attaquants, des adversaires ou des ennemis. La cryptographie professionnelle se charge non seulement de la protection du texte en clair mais aussi de la clé de chiffrement, plus généralement elle se charge de la protection du crypto-système en entier*.

[4] Kerckhoffs disait lui-même : « *Tout le monde est d'accord pour admettre la raison d'être des trois derniers desiderata ; on ne l'est plus, lorsqu'il s'agit des trois premiers.* »

Plus formellement, le processus de chiffrement d'un message m consiste en sa transformation en m' via une fonction de chiffrement $\mathcal{E}$, et en utilisant une clé K dite de chiffrement. L'opération inverse est le déchiffrement[5]. Elle consiste à retrouver à partir du message m' le message initial m en utilisant une fonction de déchiffrement $\mathcal{D}$ et une clé de déchiffrement K'.

On notera alors :

$$\mathcal{E}(m)_K = m'$$

$$\mathcal{D}(m')_{K'} = m$$

Il est possible que $K=K'$, on parlera alors de chiffrement symétrique. Autrement, et si $K \neq K'$, il s'agira d'un chiffrement asymétrique.

2.2.2.1. Taxonomie des algorithmes de chiffrement

Le chiffrement est considéré comme clé de voute de la sécurité de l'information. Suivant le type de la clé de chiffrement/déchiffrement, il est possible de distinguer entre deux types de chiffrement :

2.2.2.1.1. Chiffrement symétrique

Un algorithme de chiffrement symétrique, est un algorithme de chiffrement où les opérations de chiffrement et de déchiffrement se font avec la même clé. Ainsi, Alice (Alice et Bob étant les émetteur/récepteur traditionnels du processus de chiffrement dans la littérature du domaine) chiffre le message m en m' en utilisant la clé K ; Lorsque Bob reçoit le message m', il le déchiffre en utilisant la même clé K.

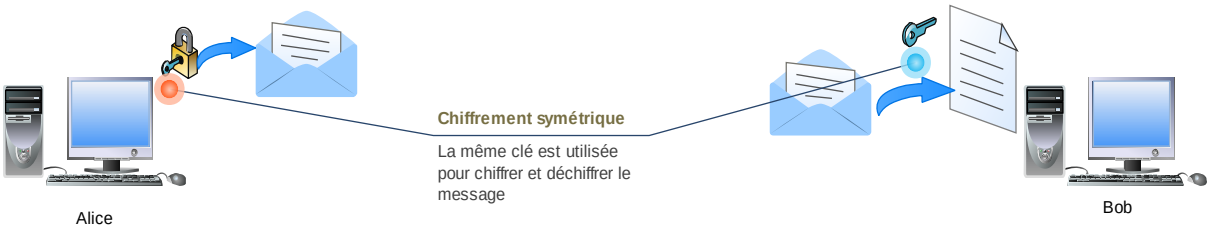


Figure 2 Chiffrement symétrique.

La confidentialité, aussi bien que l'authentification dans ce cas de figure, ne sont assurées que si la *distribution sécurisée* de la clé de chiffrement, au préalable de tout échange, est assurée. En effet, la confidentialité du transfert entre les deux protagonistes légitimes est assurée si l'on s'assure qu'une tierce entité non légitime ne détient pas, d'une manière ou d'une autre, la clé de chiffrement K et

[5] Il est à noter que le déchiffrement et le décryptage sont deux notions différentes, le déchiffrement consiste à retrouver le texte en clair à partir du texte chiffré en utilisant la clé de déchiffrement, normalement connue par le/les destinataires légitimes ; le décryptage consiste à retrouvé le texte en clair et non forcement en utilisant la clé de déchiffrement.

qu'il lui est donc impossible de déchiffrer m' . Pour ce qui est de l'authentification, cette dernière peut être assurée de par le fait de la sécurité de la distribution de la clé, car si seule Alice détient la clé K, Bob en déchiffrant le message m' via cette même clé K s'assure en même temps que c'est bien Alice qui l'a chiffré. La distribution sécurisée de la clé de chiffrement est une condition nécessaire et non suffisante pour assurer l'authentification d'un message envoyé, d'autres techniques dédiées sont utilisées pour l'assurer (e.g : signature numérique).

Pour ce qui est de l'intégrité du message envoyé, la sécurité de la clé n'est pas seule garante. Pour assurer l'intégrité d'un message envoyé, toujours en utilisant un chiffrement symétrique, il est possible d'avoir recours à un contrôle d'erreur externe ou interne.

Les figures suivantes schématisent les cas de contrôle d'erreur externe et interne :

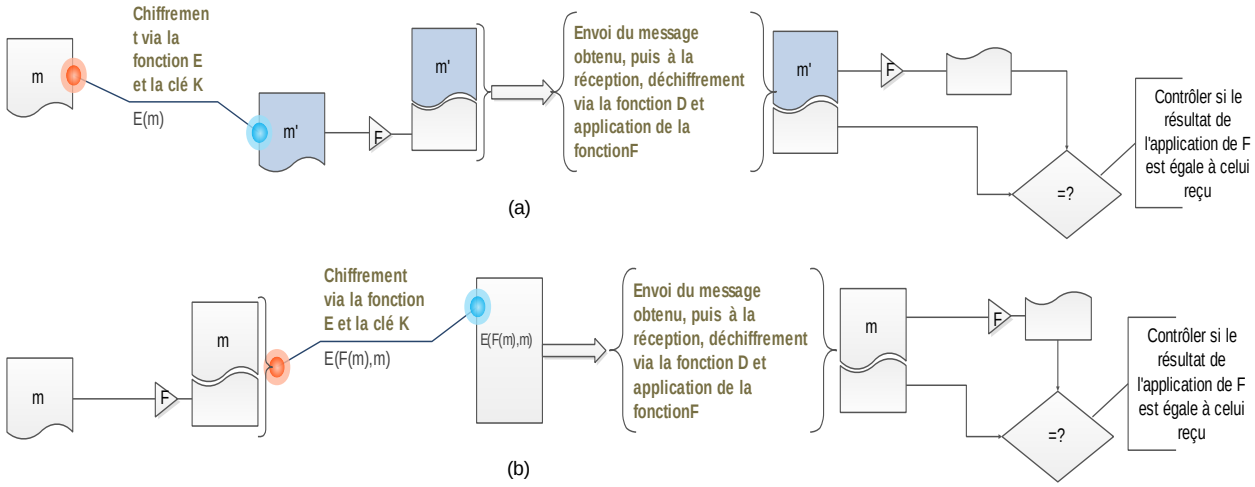


Figure 3 Contrôle d'erreur.

Dans le cas d'un contrôle d'erreur interne, *(a)* sur la figure ci-dessus, on rajoute un bloc résultant de l'application de la fonction $\mathcal{F}$ au message m. Le tout est soumis à l'algorithme de chiffrement $\mathcal{E}$ en utilisant la clé K. Une fois le message arrivé à destination, le récepteur exécute l'opération inverse, c-à-d qu'il déchiffre le message reçu ; en extrait le message initiale m, et le bloc qui lui a été rajouté ; le récepteur soumet le message mà la fonction $\mathcal{F}$, et compare le résultat au bloc extrait. Si les résultats sont identiques alors l'intégrité a probablement était préservée.

Dans le deuxième cas, *(b)* sur la figure précédente, il s'agit de la même idée, à la différence que le bloc de vérification est rajouté après le chiffrement de $m$et non avant comme c'est le cas dans (*a*). La fonction $\mathcal{F}$ est alors appliquée à $\mathcal{E}(m)$ au lieu d'être appliquée directement à m.

2.2.2.1.2. Chiffrement asymétrique

Dans le cas du chiffrement asymétrique il s'agit d'utiliser une paire de clés (publique/privée) et non plus d'utiliser une seule clé pour les opérations de chiffrement/déchiffrement. Généralement, la clé publique est utilisée pour le chiffrement alors que la clé privée est utilisée pour le déchiffrement. Ainsi, Bob distribue sa clé publique K_P à un ensemble d'entités. L'entité voulant communiquer avec Bob (soit Alice cette entité) chiffre le message $m$en utilisant la clé publique de Bob. A la réception du message chiffré, Bob utilisera sa clé privée K_S pour le déchiffrer.

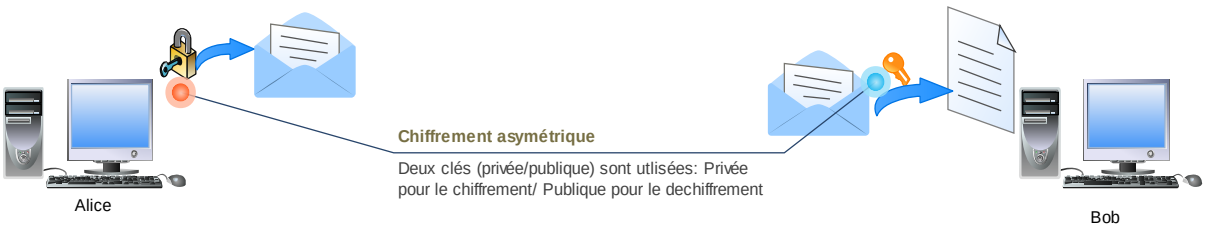


Figure 4 Chiffrement asymétrique.

Les algorithmes de chiffrement asymétrique sont basés sur la difficulté de résoudre des problèmes mathématiques. En général, il s'agit de fonction à sens unique ; une fonction à sens unique est une fonction facile à calculer dans un sens, mais difficile à calculer dans l'autre sens (avec la puissance de calcul actuelle), telle que la factorisation des grands nombres (e.g : RSA).

La confidentialité d'un message chiffré via un algorithme de chiffrement est basée sur la difficulté de résoudre ces problèmes mathématiques dans un temps acceptable en usant de la puissance de calcul actuelle.

Quant à l'authentification, il est possible d'intervertir les rôles des clés pour l'assurer. La clé privée serait utilisée pour le chiffrement alors que la clé publique servirait au déchiffrement. Ainsi Alice chiffre un message m en utilisant sa clé privée K_s et le diffuse à tous les destinataires légitimes ayant sa clé publique K_p . Bob, l'un des destinataires légitimes (ainsi que tous les autres destinataires potentiels), déchiffre le message d'Alice avec la clé publique de cette dernière, et *vérifie* par là même l'authenticité d'Alice. Alice est la seule à avoir la clé privée et donc la seule à avoir pu chiffrer le message m.

2.2.2.1.3. Hybridation

La sécurité lors de l'utilisation d'un algorithme de chiffrement symétrique est fortement liée à la distribution de la clé de chiffrement/déchiffrement qui doit être distribuée de façon sécurisée avant même d'entamer tout échange. Les algorithmes de chiffrement asymétriques quant à eux ne souffrent pas de cette problématique, du fait qu'ils peuvent servir à la génération et la distribution même des clés. Cependant, les algorithmes de chiffrement asymétriques sont gourmands en termes de temps de calcul, ce qui implique une certaine lenteur.

En pratique, il est possible de prévoir une hybridation des deux types d'algorithmes de chiffrement (voir ci-dessous Figure 5); on utilisera alors un algorithme de chiffrement asymétrique pour *distribuer* une clé au lieu de chiffrer un message. La clé ainsi distribuée est celle d'un algorithme de chiffrement symétrique.

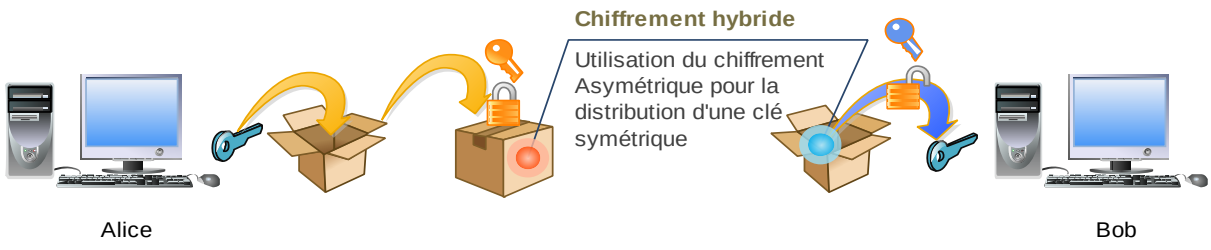


Figure 5 Hybridation: Distribution de clé symétrique via un chiffrement asymétrique.

2.2.2.2. Problématique de la distribution de clé

Il est clair que dans le cas d'un algorithme asymétrique aussi bien que symétrique, la problématique de la distribution de clé s'impose. Dans le premier cas, où la clé publique est utilisée pour le chiffrement et la clé privée pour le déchiffrement, ces clés sont générées en se basant sur des hypothèses calculatoires, comme la difficulté de factorisation des grands nombres. Cela dit, si la puissance de calcul nécessaire était disponible, la sécurité de ce genre d'algorithme serait remise en question. Dans le second cas où la même et unique clé est utilisée pour le chiffrement et le déchiffrement, la clé doit être partagée au préalable et de façon sécurisée. La question qui se pose est alors comment sécuriser cette clé.

L'hybridation (citée précédemment) est une technique utilisée pour réaliser une distribution de clé, plus ou moins, sécurisée, si l'on suppose la puissance de calcul de l'espion limitée. D'autres techniques existent pour résoudre cette problématique et que l'on évoquera plus loin.

2.2.3.La signature numérique

Tirer du concept même de la signature manuscrite, qui est censée être unique à une personne donnée, la représentant et permettant de là même à l'identifier, la signature numérique suit plus ou moins le même principe. En effet, il ne s'agit pas seulement de quelque chose d'unique au signataire et indépendant à l'information signée [14], il s'agit plutôt de techniques utilisées afin de *lier* l'identité d'une entité à une information, ce qui implique la transformation du message, et de certaines informations jugées secrètes et connues par le signataire, en un *tag* représentatif dit *signature*.

Selon la norme ISO 7498-2, la signature numérique est définie comme étant « *des données ajoutées à une unité de données, ou transformation cryptographique d'une unité de données, permettant à un destinataire de prouver la source et l'intégrité de l'unité de données …* »

Plus formellement, la signature numérique peut être définit comme suit [14], :

- Soit $\mathcal{M}$ l'ensemble des messages pouvant être signé ;
- Soit $\mathcal{S}$ l'ensemble des signatures ;
- S_A est une transformation de l'ensemble des messages $\mathcal{M}$ vers l'ensemble des signatures $\mathcal{S}$. S_A est la transformation de signature de l'entité $\mathcal{A}$. S_A est gardée secrète par l'entité $\mathcal{A}$.
- V_A est une transformation de l'ensemble $\mathcal{M} \times \mathcal{S}$ vers l'ensemble {vrai, faux} . V_A est une transformation de vérification de la signature de l'entité $\mathcal{A}$. V_A est publique et est utilisée par les autres entités pour vérifier les signatures crées par l'entité $\mathcal{A}$.
- S_A et V_A offrent un *schéma de signature numérique* pour l'entité $\mathcal{A}$.

La signature numérique permet d'éviter le problème de *personnification*[6], et donc de vérifier l'authentification d'une information signée. En effet, l'expéditeur légitime ayant une signature unique, puisqu'elle est liée à des données privées, est le seul à pouvoir la déposer sur l'information.

[6] Personnification : Une entité, non-autorisée, se fait passer pour une autre qui l'est.

Vu que la signature numérique est liée à des données privées, un expéditeur ayant signé un message ne peut nier l'avoir fait. La signature numérique permet donc d'assurer l'un des principaux buts de la sécurité de l'information, en l'occurrence la non-répudiation.

D'un point de vue pratique, l'utilisation des algorithmes de chiffrement, tel que les algorithmes de chiffrement à clé publique, où le fait même de chiffrer le message via la clé privée constitue une forme de signature, puisque seul le possesseur de cette clé est habilité à l'utiliser. Les destinataires légitimes pourront vérifier cette *signature* en déchiffrant le message via la clé publique.

Cependant, vu l'inconvénient rencontré lors de l'utilisation de tels algorithmes, en l'occurrence la lenteur induite par les calculs, il est plus judicieux de chiffrer seulement une partie du message. Cette partie est une masse plus réduite du message, mais représentative de ce dernier (aussi dite empreinte du message).

L'utilisation d'empreinte réduit considérablement le temps nécessaire au chiffrement. L'empreinte, elle, est générée via une fonction de hachage, et est chiffrée via un algorithme de chiffrement e.g : RSA. La propriété d'unicité de la signature numérique est ainsi renforcée grâce à l'utilisation des fonctions de hachage caractérisées par la propriété de *non-collision*[7], ce qui permet par conséquence d'assurer l'intégrité du message.

2.2.3.1. Taxonomies de signature numérique

Il est possible de distinguer entre deux types de signature numérique si l'on se réfère à l'entité signataire :

2.2.3.1.1. La signature directe

C'est le cas de l'utilisation des algorithmes de chiffrements à clé publique. Dans ce cas, on suppose qu'Alice, récepteur dans cette opération, ait la clé publique de Bob, elle chiffre alors le message ou son condensé, avec sa clé privée. Le message ainsi signé est chiffré via la clé publique de Bob. A la réception, Bob pourra déchiffrer le message via sa propre clé privée, mais aussi vérifier la signature d'Alice en déchiffrant le condensé avec la clé publique d'Alice.

2.2.3.1.2. La signature arbitrée

Les parties prenantes d'une communication peuvent, réciproquement, ne pas se faire confiance. Dans ce cas, il faut faire intervenir une entité tierce, *un arbitre*, en laquelle toutes les parties feraient confiance, elle validera le message signé, le datera et l'enverra à destination.

Si par contre on se réfère à la nécessité de l'utilisation du message originel durant l'étape de vérification de la signature, il est alors possible de distinguer entre [14]:

2.2.3.1.3. Signature avec appendice

Une signature avec appendice se base sur l'utilisation des fonctions de hachage et des paires de clés privée/publique. La particularité de ce type de signature est l'utilisation du message originel lors de la phase de vérification. En effet, le message initial est pris pour entrée pour le processus de

[7] Non-collision : deux entrées différentes d'une fonction de hachage n'ont pas la même sortie.

vérification, permettant ainsi de recalculer la signature et la comparer avec celle reçue, ce qui permet de la valider ou non (voir ci-dessous Figure 6).

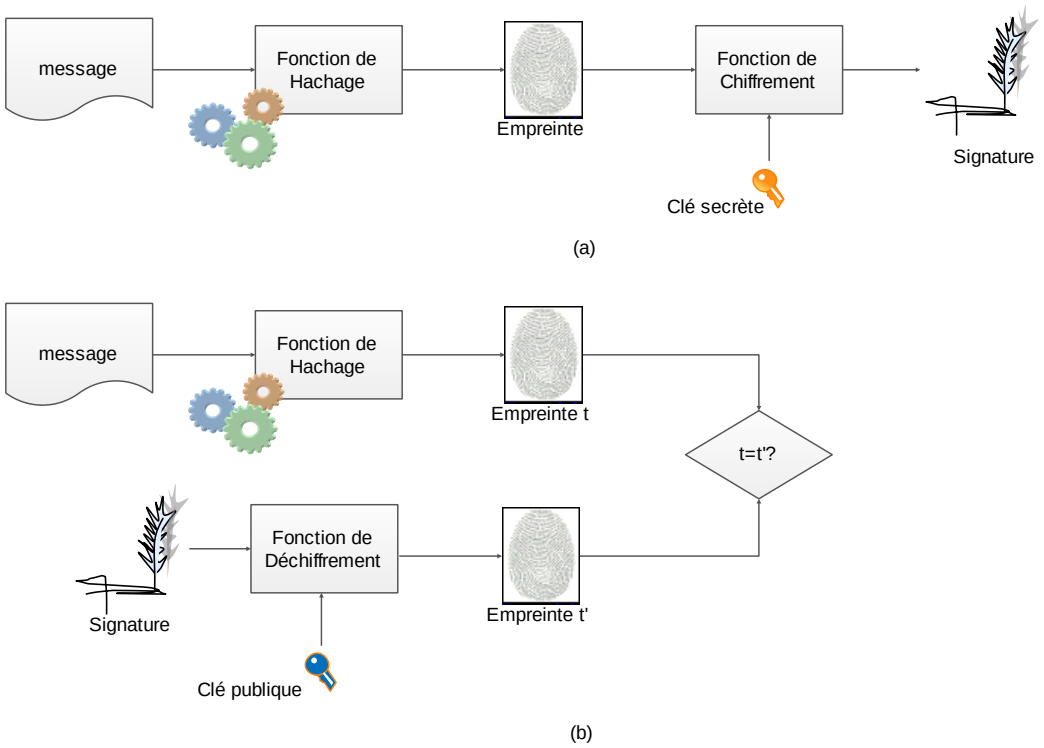


Figure 6 Illustration de la signature numérique via une fonction de hachage : (a) génération de la signature ; (b) phase de vérification de la signature.

2.2.3.1.4. *Signature avec recouvrement de message*

Dans ce cas de figure, nul besoin de connaitre, a priori, le message original. La fonction de signature est appliquée sur le message, et lors de la vérification, la signature $\mathcal{s}$ servira au recouvrement du message original m.

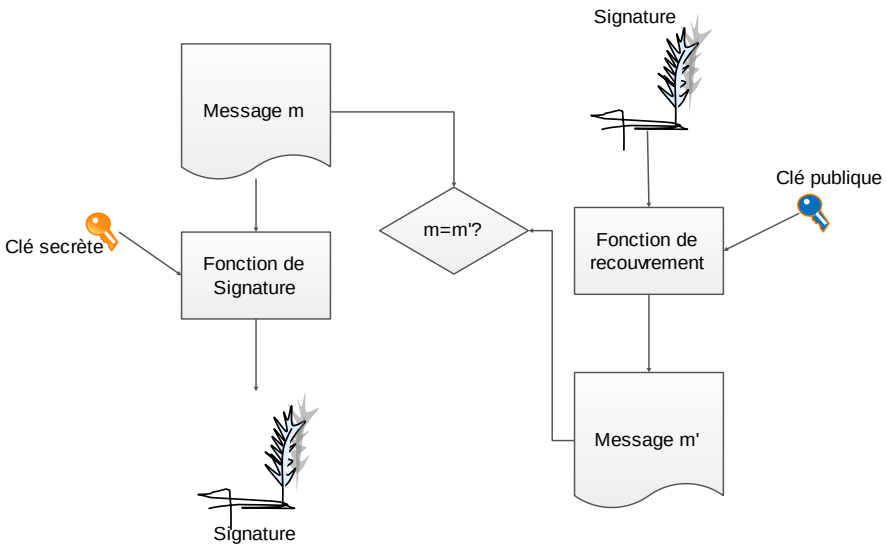


Figure 7 Signature numérique avec recouvrement.

2.2.4. Stéganographie

Cacher l'information (qu'elle soit textuelle, audio, ou vidéo), est la façon la plus intuitive d'éviter qu'une tierce partie, éventuellement malveillante, ne puisse y accéder et/ou la modifier. Où et comment *cacher* une information relève justement de la stéganographie.

La stéganographie, tout comme la cryptographie, est une technique utilisée à des fins sécuritaires, la différence entre les deux réside dans la façon de s'y prendre. En effet, alors que la cryptographie, qui consiste à transformer l'information initiale en une autre tout à fait inintelligible pour toute partie illégitime, et qui est reconnue comme étant l'art du secret, la stéganographie, elle, est l'art de dissimuler un message [21]. Dans [22] la stéganographie est la science de communiquer l'information tout en dissimulant l'existence même d'une communication.

Etymologiquement, la stéganographie, du grec *Stéganô καλύπτουν* signifiant « je couvre » et *Graphô γράφω* « j'écris » d'où ressort l'idée de couvrir un message, autrement dit, cacher un message m dans un autre message m' de façon imperceptible, de sorte même que le message m reste anodin.

En termes plus simples, la stéganographie permet de caché une information dans une autre. Cela dit, il faut savoir que toute information n'est pas nécessairement valable pour servir d'hôte et y insérer une autre information secrète. En effet, il faudrait que l'information hôte contienne assez de redondances pour être supprimer et remplacer par l'information à *cacher*.

2.2.4.1. Taxonomie des protocoles stéganographiques

En littérature du domaine [21] et [23], il est possible de distinguer entre trois types de protocoles de stéganographie : stéganographie pure, stéganographie à clé secrète, et stéganographie à clé publique :

Dans ce qui suit, nous adopterons les définitions formelles proposées par Stefan C. Katzenbeisser dans [21].

2.2.4.1.1. Stéganographie pure

On appelle stéganographie pure, un protocole de stéganographie où l'on n'a pas besoin d'un échange d'information au préalable. L'expéditeur et le récepteur doivent tous les deux avoir accès aux algorithmes d'insertion et d'extraction, sans pour autant que ces algorithmes soient rendus publics.

Plus formellement, un système de stéganographie pure se définit comme suit : C est l'ensemble des messages hôtes possibles, M l'ensemble des messages secrets, avec $|C| \geq |M|$, $E : C \times M \rightarrow C$ étant la fonction d'insertion, et $D : C \rightarrow M$ la fonction d'extraction, avec la propriété suivante : $D(E(c,m)) = m$ pour tout $m \in M$ et $c \in C$, alors le quadruplet $< C, M, D, E >$ est dit un système de stéganographie *pure*.

2.2.4.1.2. *Stéganographie à clé secrète*

La stéganographie pure se base sur la confidentialité des algorithmes d'insertion et d'extraction, ce qui va à l'encontre du deuxième principe de KERCKHOFFS (voir 2.2.1 Clé et principe de KERCKHOFFS).

Il faudrait donc que les algorithmes d'insertion et d'extraction soient publics, et que la sécurité du système de stéganographie se base sur une autre information secrète connue des protagonistes légitimes, la *stégo-clé*. La stégo-clé sera utilisée pour insérer l'information secrète dans l'information hôte et de l'en extraire.

Formellement, C est l'ensemble des messages hôtes possibles, M l'ensemble des messages secrets, avec $|C| \geq |M|$, K est l'ensemble de clés secrètes, $E_K : C \times M \times K \rightarrow C$ et $D_K : C \times K \rightarrow M$ avec la propriété suivante : $D_K(E_K(c,m,k)) = m$ pour tout $m \in M$, $c \in C$ et $k \in K$ alors le quintuple $<C, M, K, D_K, E_K>$ est dit un système de stéganographie à *clé secrète*.

Il est tout à fait évident qu'un protocole de stéganographie à clé secrète souffre des mêmes lacunes qu'un algorithme de chiffrement symétrique. En effet, la problématique de distribution de clé (la stégo-clé dans ce cas) refait surface, et les protagonistes doivent auparavant se mettre d'accord sur une clé et se partager cette information de façon sécurisée.

2.2.4.1.3. Stéganographie à clé publique

De la même manière que le chiffrement asymétrique, un protocole de stéganographie à clé publique requiert une paire de clé, l'une publique et l'autre privée. Dans un tel schéma, la clé publique sera utilisée pour chiffrer l'information et obtenir ainsi un message aléatoire qui sera par la suite insérer dans l'information hôte en utilisant la fonction d'insertion. A l'arrivée, le récepteur *soupçonne* l'existence d'une information secrète qui lui est destinée, et essayera d'extraire le message aléatoire, en utilisant la fonction d'extraction, puis le déchiffre en utilisant sa clé privée.

Ainsi, et sachant que l'algorithme de chiffrement aussi bien que la fonction d'insertion sont publiques, un espion peut essayer de d'extraire l'information secrète, mais il n'obtiendra cette dernière que sous sa forme aléatoire et ne peut être lue par l'espion que si ce dernier détienne la clé privée du récepteur.

A noter qu'il est possible d'exécuter un protocole de stéganographie pure tout en se basant sur l'utilisation des clés publiques. Alice et Bob peuvent partager ainsi une clé de session en l'envoyant sur un canal subliminal.

L'idée du protocole est proposée par S. Craver dans [23] et les étapes sont les suivantes :

1. Alice génère une paire de clé privée/publique (utilisant l'algorithme RSA par exemple).
2. Alice injecte la clé dans un canal subliminal visible par Bob aussi bien qu'Eve, et envoie *l'objet-stago* à Bob. Ni Bob, ni l'espion Eve ne peuvent déterminer si le canal contient une clé ou simplement un bruit.
3. Bob soupçonne qu'il s'agit là d'une clé publique, et l'utilise pour un message d'acquittement utilisant une clé secrète aléatoire. Il renvoie le tout à Alice dans l'objet-stego.
4. Alice soupçonne l'existence d'une clé de chiffrement dans l'objet-stego ; Elle le déchiffre avec sa clé privée et la récupère.

2.2.4.2. Quelques techniques de stéganographie

L'utilisation de la stéganographie remonte à l'antiquité. Hérodote (486-425 av J-C) utilisait son esclave le plus fidèle pour transmettre un message incitant à la révolte contre la Perse. Le message

était tatoué sur le crâne rasé de l'esclave. Une fois la chevelure redevenue normale, l'esclave était envoyé à destination et avait l'ordre de se raser le crâne une fois arrivé [21]. Plus encore, *Enée* le tacticien (5eme siècle av J-C), auteur de *La poliorcétique*, introduit dans le chapitre 31 les techniques de transmission des messages, entre autres, cacher le message dans les semelles du messager ou dans les boucles d'oreilles des femmes pour le transmettre [24].

D'autres techniques ont vu le jour en allant des tablettes de cire, à l'ancre invisible (vinaigre, lait…) en passant même par la stéganographie linguistique où il suffisait de lire une ligne et sauter une autre, ou alors de lire le premier mot de chaque vers d'un poème pour reconstruire un message caché.

Plus tard, au 17^{ieme} siècle, Wilkins (1614–1672) marque les lettres du message d'un point à l'encre invisible dans le message porteur (ou cover-message) [21].

La technique du *micropoint* elle, est apparue vers la première guerre mondiale. Un message, une image par exemple, était réduit à une échelle photographique minuscule, jusqu'à paraitre comme un point. Le point est alors inséré dans un texte anodin comme ponctuation [25].

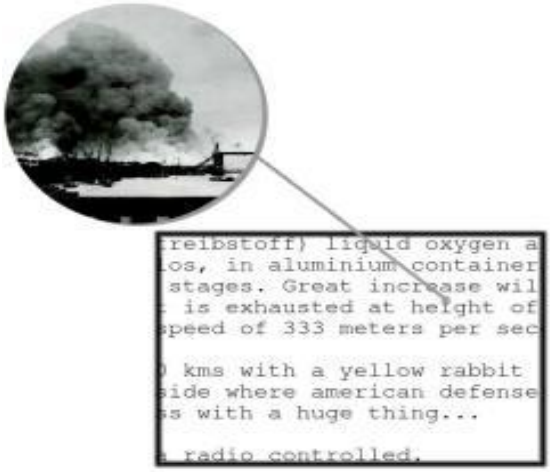


Figure 8 Technique du micropoint [25].

La technique de Semagram, du grec σημα « le signe », et γραμμα, « le texte écrit », où le texte chiffré n'est ni lettres ni chiffres, mais le sens est véhiculé par d'autres éléments faisant figure dans une image, une peinture, un graphème…et qui représentent les tirets et les points de l'alphabet Morse [15].

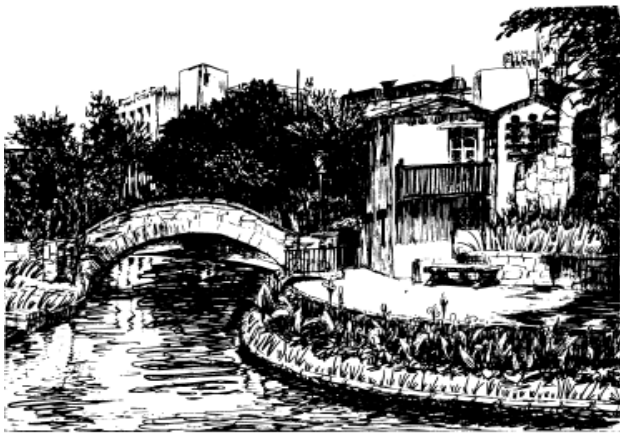


Figure 9 Technique du Semagram[8] [15]

Ce qui fait la différence entre la stéganographie et autres techniques, comme le watermarking et l'empreinte digitale, où il est toujours question de cacher une information particulière dans une autre (l'information originel), est que l'information à introduire dans le cas du watermarking, et qui est dite « *marque* », est une information dépendante de l'information initiale, alors que ce n'est pas forcement le cas dans le cas de la stéganographie.

Une deuxième différence entre la stéganographie et le watermarking, est que dans le premier cas, une information est introduite dans une autre de sorte à la cacher, c'est le principe même de la technique, alors que dans le second cas, l'information introduite n'est pas forcement cachée, elle peut très bien être perceptible, à l'exemple des logos visibles sur des images.

Une autre différence réside dans le type d'attaque. Une attaque dans le cadre de la stéganographie portera sur la détection de l'information potentiellement cachée, alors qu'une attaque dans les cas de l'empreinte et/ou du tatouage portera sur la suppression de la marque même.

2.2.5.Tatouage (*watermarking*)

Apparu au 13ième siècle, le tatouage numérique (aussi dit *watermarking*) servait comme tags d'authentification pour les supports imprimés, son équivalent numérique sert plutôt à vérifier le copyright. Le plus ancien tatouage retrouvé, remonte à 1292 et retrouvé dans la ville de *Fabriano* en *Italie*, qui a joué un rôle important dans l'industrie et la fabrication du papier, domaine où les artisans usaient des tatouages pour marquer leur produit et éviter toute confusion [21].

L'utilisation de la technique du tatouage a vite pris de l'ampleur, et dispose aujourd'hui de son équivalente numérique, toujours dans le but d'assurer l'authentification de l'information hôte du tatouage.

Dans [15] Gerrit Bleumer définit le tatouage numérique comme étant une méthode qui permet d'identifier la source ou le propriétaire du copyright d'un signale digitale ou analogique, qui est inclus dans le signale même, afin de garder la trace de provenance ou afin de déterminer le propriétaire du copyright.

[8]Le message est dissimulé dans l'image à gauche du pont est "compliments of CPSA MA to our chief Col.Harold R. Shaw on his visit to San Antonio May 11, 1945", les brins d'herbe le long du fleuve et sur le mur du jardin représentent les traits et les points du l'alphabet Morse.

Il est donc possible de considérer le tatouage comme une forme de signature numérique, où l'on ajoute une information appelée marque, à l'information initiale. En générale la marque est une information de copyright qui permet d'assurer en partie l'authentification du message du point de vue vérification de l'identité.

Le processus de tatouage numérique se compose de deux phases, à savoir la phase intégration du tatouage et la phase recouvrement (ou extraction) du tatouage. L'entrée de la première phase est constitue de la donnée à tatouer, d'une clé (publique ou privée, et l'on parlera, respectivement, de tatouage à clé publique et tatouage à clé privée) et du tatouage en soit (appelé aussi marque qui est une information de copyright). Le tatouage peut être soit un nombre, une image (un logo) ou encore un texte.

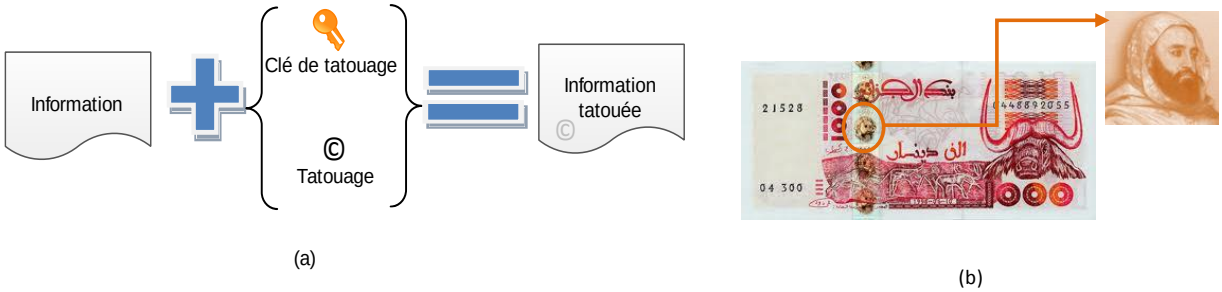


Figure 10 Tatouage numérique : (a) Schéma général de tatouage ; (b) Tatouage sur un billet d'argent

Figure 11 Tatouage monogramme retrouvé à Cambridge, Grande Bretagne[9] [21]

Comme il a été précisé dans la section précédente, le tatouage numérique, à l'image de la stéganographie, vise à cacher une information à l'intérieur d'une autre dite hôte. Cependant, à la différence de la stéganographie, et en plus des différences que nous venons d'introduire dans la section précédente réservée à cette technique (voir 2.2.4 Stéganographie), la technique de tatouage numérique se caractérise par la propriété de résistance aux modifications. Ainsi, un quelconque changement appliqué à l'information hôte, sera détectable via la distorsion décelée sur le tatouage, alors qu'il est impossible de la détecter avec la seule technique de stéganographie.

[9] Sur le monogramme les initiales: *TGE RG*: Thomas Goodrich Eliensis—Bishop of Ely, England—and Remy/Remigius Guedon, the papermaker. Un des plus anciens tatouages, retrouvé dans la région de Cambridge et utilisé à l'époque pour identifier le fabriquant de papier.

A noter qu'un tatouage idéal devrait pouvoir résister à tout rognage, compression, rotation ou toute autre distorsion, malheureusement, il n'y a pas actuellement un tel tatouage mais il existe des tatouages qui assurent une certaine résistance aux distorsions basiques.

2.2.5.1. Propriétés d'un tatouage

Un système de tatouage doit assurer un certain nombre de propriétés [21]:

a) L'imperceptibilité

La distorsion que subit l'information suite à l'insertion d'un tatouage doit être mesurée. En d'autre termes, les modifications dues à l'insertion du tatouage, doivent être en dessous d'un certain seuil de perceptibilité de sorte à ce que l'information tatouée et l'information initiale soit pratiquement identiques (e.g : même image visible ; même son audible…).

b) La redondance

En dépit de la quantité minime d'information (qu'est le tatouage) autorisée à être insérée dans l'information initiale, cette quantité devra assurer une certaine robustesse. Cette dernière peut être assurée par la duplication du tatouage dans différentes zones de l'information initiale, de sorte à pouvoir la retrouver durant la phase d'extraction en traitant n'importe quelle partie de l'information tatouée.

c) La clé

Une clé de chiffrement doit être utilisée pour protéger le tatouage contre toute manipulation ou suppression de ce dernier.

2.2.5.2. Taxonomie du tatouage

Selon la possibilité ou non de séparer la marque du message, on distingue entre un tatouage fragile et un tatouage robuste. Un tatouage fragile est un tatouage sensible à toute modification de l'information tatouée. Il permet ainsi de détecter toute tentative de modification de cette information. Un tatouage fragile, permet donc l'authentification de la donnée. Un tatouage robuste par contre, est un tatouage qui a été intégré dans l'information initiale même. Un tatouage robuste ne peut être séparé de l'information initiale qu'après modifications majeures que subit l'information tatouée, jusqu'à la rendre « inutilisable ».

Suivant la nature et le type des entrée et sortie du processus de tatouage, il est possible d'en distinguer trois types: tatouage privé, semi-privé et public [21].

2.2.5.2.1. Tatouage privé (dit nonblind watermarking)

Le tatouage privé requiert l'information initiale lors de l'étape de recouvrement. En fait, on distingue deux cas figure dans le cas d'un tatouage privé : Durant la phase d'extraction, le système de tatouage extrait la marque de l'information *déformée*, et utilise l'information initiale pour déduire l'emplacement de la marque ; dans le second cas, le système requiert une copie de la marque et déduit si *oui* ou *non* l'information *déformée* contient effectivement la marque.

2.2.5.2.2. Tatouage semi-privé (dit semiblind watermarking)

Dans le cas d'un tatouage semi-privé, nul besoin de l'information initiale, mais on y répond à la même question si oui ou non une marque existe. Le tatouage semi-privé est utilisé pour la détection de tatouage dans un contenu multimédia. Il est utilisé comme preuve de propriété de ce même contenu, mais sert aussi pour le contrôle de copie (e.g : un lecteur de disque doit pouvoir savoir si oui ou non il a le droit de lire telle ou telle information sur un DVD).

2.2.5.2.3. Tatouage public (dit blind/oblivious watermarking)

Dans ce cas de figure, bien qu'on y réponde toujours à la même question, l'information initiale, n'est pas nécessaire pour la phase de recouvrement du tatouage. Seule la clé est nécessaire pour la phase de recouvrement.

2.2.6. Empreinte numérique

L'idée de l'empreinte numérique, découle de celle de l'empreinte humaine. En effet, l'être humaine se distingue par une empreinte digitale qui le caractérise des autres êtres, par son unicité. Même de vrais jumeaux n'ont pas la même empreinte, ce qui permet de la considéré comme un bon paramètre d'identification, et d'authentification par la suite.

Historiquement, les premiers travaux autours de l'empreinte digitale remontent au 16^{ième} siècle et c'est en 1864 que Nehemiah Grew, publie le premier papier qui rapport une étude systématique sur les crêtes et les sillons des empreintes. Mayer, en 1788, a établi une description détaillée des empreintes digitales basée sur la caractérisation des crêtes, mais c'est Purkinje qui en établit la première classification basée sur la configuration des sillons, il en déduit neuf catégories [26]. Plus tard, en 1888, Sir Francis Galton, introduit les caractéristiques des *minuties* pour la comparaison des empreintes digitales [26].

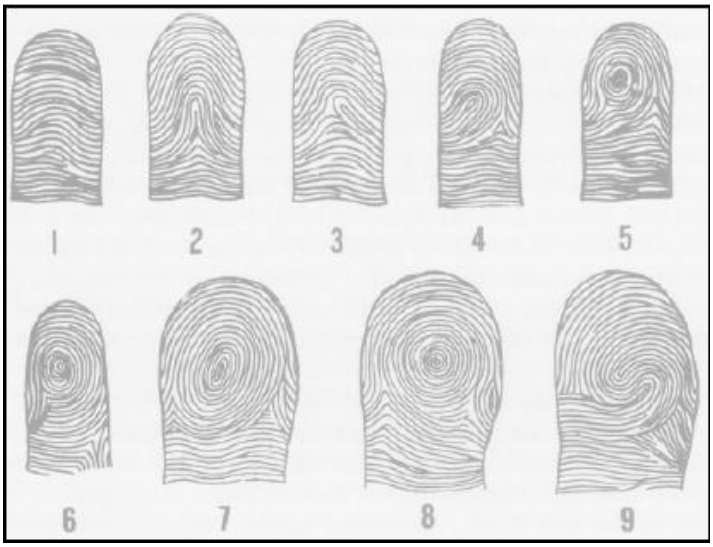


Figure 12 Types d'empreintes digitales [26].

Les caractéristiques de l'empreinte qui lui valent une telle primauté sont les suivants [26]:

1- Les crêtes et les sillons de l'épiderme ont des caractéristiques différentes pour différentes empreintes.
2- La configuration des types sont variables de façon à permettre une classification.
3- Les configurations et les détails individuels des crêtes et des sillons sont permanents et invariable.

D'autres caractéristiques biométriques peuvent faire l'objet de paramètre d'identification, comme l'iris, la démarche, la reconnaissance faciale ou vocale, la fréquence de frappe au clavier…mais il s'avère que l'empreinte, reste la caractéristique la plus facile à utiliser et la plus acceptée aussi pas l'utilisateur. Les numéros de série des produits sont aussi uniques et peuvent aussi être utilisés comme empreinte digitale [21].

Pour Alexander Barg et al. dans [15] l'empreinte numérique est une technique qui vise à prévenir la redistribution non autorisée de contenu numérique. Cela consiste à intégrer une marque χ(une empreinte digitale) dans le document dans le but d'encoder l'identité de l'utilisateur.

Autrement dit, la technique de l'empreinte numérique, aussi appelée *fingerprinting*, permet d'associer une origine à une information, cela permet de tracer les utilisateurs non autorisés qui distribuent illégalement des copies de la dite information.

L'empreinte numérique ne doit pas être facilement détectable ni amovible. Aussi l'empreinte numérique doit être conçue de sorte que sa contrefaçon soit difficile ou couteuse.

A signaler qu'un *utilisateur autorisé*, à la différence d'un utilisateur non autorisé, est un utilisateur qui peut accéder à des objets portant une empreinte précise ; Un *attaquant* est un individu qui réussit à accéder aux objets avec empreinte ; Un *traitre* est un utilisateur autorisé qui distribue illégalement des objets portant une empreinte. Un exemple est celui de la diffusion des chaines satellitaires chiffrées, ces dernières insèrent des bits d'empreinte numériques dans chaque paquet envoyé, ainsi si un utilisateur non autorisé obtient la clé de déchiffrement lui permettant de lire le trafic il sera détecté [21].

2.2.6.1. Taxonomie des techniques de fingerprinting

Une classification des techniques d'empreinte, en suivant trois critères différents, est donnée dans [21]. On distingue entre :

2.2.6.1.1. Basée objet

La première des classifications se base sur la nature de l'objet sur lequel sera apposée l'empreinte (objet hôte). Si l'objet est de nature digitale, on parlera d'empreinte *digitale*, dans ce cas de figure, l'ordinateur sera utilisé pour calculer ladite empreinte. Si par contre l'objet est de nature physique, ayant ses propres caractéristiques pouvant l'identifier et le distinguer des autres objets, on parlera d'empreinte *physique* (Iris, voix,… introduits précédemment).

2.2.6.1.2. Basée sur la sensibilité de la détection

On distinguera entre trois types de d'empreinte : empreinte parfaite, empreinte statistique, et empreinte à seuil.

Dans le premier cas, une altération qui touche l'empreinte rend l'objet non utilisable. Autrement dit, toute modification malveillante de l'empreinte se répercutera *ipso facto* sur son hôte, ce qui permettra sa détection.

Ayant un nombre suffisant d'objets mal-utilisés à analyser, dont l'utilisateur compromis a été identifier, le générateur d'empreinte peut atteindre n'importe quel niveau de confidentialité demandé.

Le dernier type, en l'occurrence l'empreinte à seuil, accepte un certain nombre d'utilisations illégales de l'objet. Ce nombre est dit seuil. Une fois le seuil atteint les copies illégales sont identifiées. Ainsi, les copies ne sont pas détectées tant que leur nombre ne dépasse pas le seuil. Autrement dit, les copies sont tracées.

2.2.6.1.3. Basée sur la méthode de fingerprinting

La suppression, l'addition et la modification, sont aussi considérés comme des critères de classification des empreintes. Si le schéma de *fingerprinting* consiste à reconnaitre et enregistrer des parties de l'objet même ; il s'agira de reconnaissance.

Le processus de *fingerprinting* peut renfermer soit une suppression d'une partie de l'objet ou un rajout, on parlera alors respectivement de schéma d'empreinte par ajout et par suppression. Si l'adition ne se fait que dans une partie on parlera alors de modification.

2.2.7. Certificat numérique

Toujours dans le but d'assurer la sécurité de l'information, on peut avoir recours à un autre dispositif à savoir les certificats. Les certificats sont en fait utilisés dans le cadre de la gestion des clés. Il s'agit d'une *structure de données signée par une entité considérée (par un ensemble d'autres entités) comme étant garante de son contenu. La signature de la structure de donnée lie les informations de sorte qu'il est impossible de les altérer sans que cela ne soit perceptible.* [15]

2.2.7.1. Forme générale d'un certificat

Le standard X.509 décrit la forme générale du certificat numérique. Le standard X.509 a été initialement lancé en 1988 et fut révisé au fur et à mesure ; une version a été lancée en 1993 puis une troisième a vu le jour en 1995 pour être révisée en 2000 [12].

Le standard se base sur l'utilisation de la cryptographie à clé publique et la signature numérique, il comprend les informations suivantes [17] [12] (Voir ci-dessous Figure 13) :

1. Sujet (*Subject*) : Nom distingue, Clé publique ;
2. Emetteur (*Issuer*) : Nom distingue, Signature ;
3. Période de validité : date de début, date de fin, les dates définissent la période durant laquelle le certificat est encore valide ;
4. Informations administratives :
    a. Version : Qui permet de distinguer entre les différentes versions, celle par défaut étant 1 ;
    b. Numéro de série : Qui est un entier unique au sein de l'entité CA (***C****ertificat* ***A****uthority*), et permet d'associer, sans ambiguïté, le certificat à l'entité CA ;

5. Identificateur de l'algorithme de signature : Algorithme utilisé pour signer le certificat.
6. Information de la clé publique du sujet : Champ contenant la clé publique du sujet ainsi que l'identifiant de l'algorithme pour lequel cette clé sera utilisée.
7. Identificateur de l'émetteur : Champ optionnel contenant une chaine de bits identifiant de façon unique l'émetteur du certificat.
8. Identificateur du sujet : Champ optionnel contenant une chaine de bits identifiant de façon unique le sujet.
9. Extensions : Un ensemble d'un ou plusieurs champs rajoutés dans la version 3 [12].
10. Signature : Contient le '*hache*', des autres champs, chiffré avec la clé privée de l'entité CA. Contient également l'identificateur de l'algorithme utilisé pour la signature.

A signaler qu'un sujet peut avoir plusieurs certificats, l'information « Nom distingué » permet alors de fournir une identité du sujet dans un contexte spécifique. L'information « Nom distingué » renferme les champs suivants : Common Name, Organization|Company, Organizational Unit, City/locality, et Country [12].

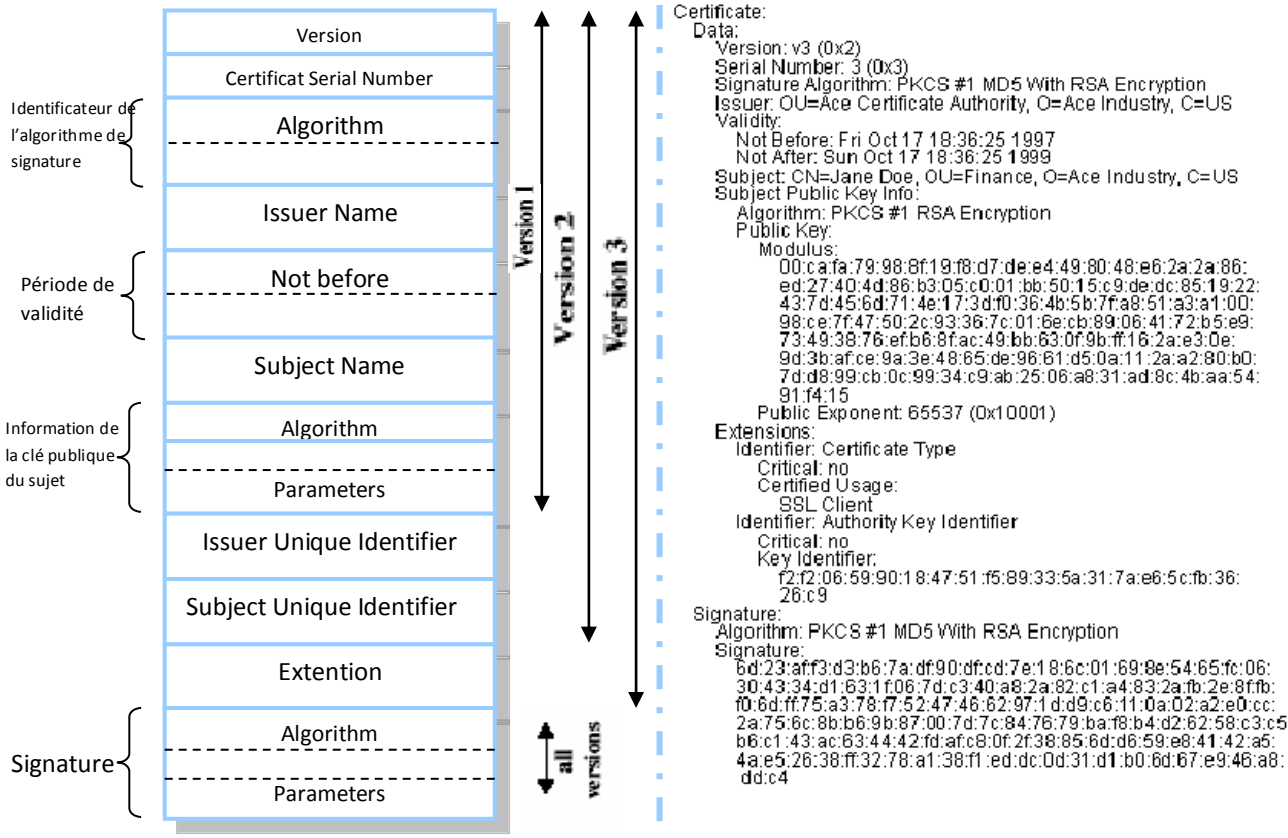


Figure 13 Structure standard d'un certificat exemple du X.509.

2.2.7.2. Principe de fonctionnement

Le certificat est créé dans un premier temps par l'entité CA suite à la demande de l'entité, supposant Alice, après que celle-ci lui ait fourni des informations concernant son identité, ainsi que sa clé publique. Enfin, l'autorité se doit de signer le certificat via sa propre clé privé et un algorithme tel que le RSA. Une fois créé, le certificat est délivré par le serveur de certificat, et est remis à l'entité qui détient la clé secrète associée, donc à Alice dans ce cas.

Ainsi, pour qu'une entité, soit Bob, puisse communiquer avec une autre entité, soit Alice, il lui suffit de récupérer son certificat, dans lequel il pourra trouver la clé publique d'Alice mais aussi d'autres informations tel que la date de validité de cette clé publique, …. Bob pourra également vérifier l'authenticité de la clé d'Alice via la signature que lui aurait apportée l'entité CA, du fait que la clé publique de la CA est mise à la disposition de tous les utilisateurs du système.

Pour pouvoir crée le certificat, il est à signaler que l'on peut distinguer entre deux cas possibles :

- Soit que c'est l'autorité de certificat qui va crée la paire de clé (privée-publique), l'associe à une entité ***A***, envoie une copie de la clé privée à cette même entité ***A*** via un canal sûr, puis créer le certificat en question.

- Ou bien, c'est l'entité en question qui crée sa propre paire de clé (privée-publique), envoie une copie de la clé publique à l'entité de certificat de façon à préserver son authenticité, via un canal sûr par exemple, pour que celle-ci puisse enfin créer le certificat.

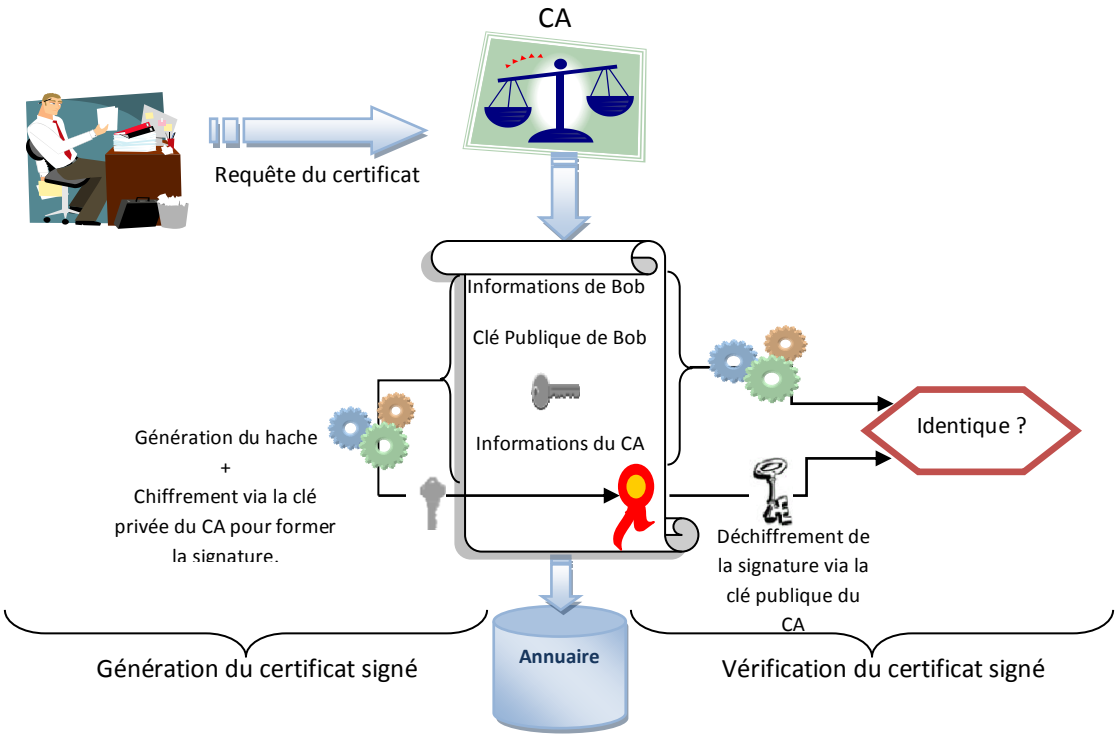


Figure 14 Principe du certificat numérique.

Si l'on suppose qu'Alice et Bob détiennent des certificats délivrés par deux CAs, respectivement CA_1 et CA_2. Dans ce cas de figure Alice à besoin de connaitre de manière sûre la clé publique du CA_2 sinon, le certificat de Bob n'est d'aucune utilité, puisqu'Alice ne peut pas vérifier la signature qu'il renferme. Cependant, si les deux CAs ont échangé de manière sûre leurs clés publiques respectives, alors ont peut s'attendre à un tel scenario :

1- Alice obtient le certificat de CA_2 signé par CA_1. Et puisqu'elle connait la clé publique de CA_1 elle pourra récupérer la clé publique de CA_2 du certificat qui lui a été délivré par CA_1 et par là même la vérifier.

2- Alice peut maintenant récupérer le certificat de Bob qui lui est délivré par CA_2, le vérifier en utilisant la copie de clé publique de CA_2 et y récupérer de façon sûre la clé publique de Bob.

On dira qu'Alice a utilisé une chaîne de certificats pour obtenir celui de Bob. Suivant la notation du standard X.509 ceci est noté comme suit [12]:

$$CA_1 << CA_2 >> CA_2 << Bob >>$$

Bob pourra, également, en utilisant une chaîne de certificats, obtenir le certificat d'Alice :

$$CA_2 << CA_1 >> CA_1 << Alice >>$$

On notera que ce schéma n'est pas limité et la chaîne de certificats peut avoir une longueur quelconque.

2.2.7.3. Révocation des certificats

Chaque certificat a une période de validité (voir section 2.2.7.1), cependant, des circonstances peuvent arriver et font qu'un certificat n'est plus valide même avant son expiration. Pour cela chaque CA doit maintenir une liste (dite ***CRL C**ertificat **R**evocation **L**ist*) contenant tous les certificats ayant été révoqués avant expiration. Un certificat peut être révoqué pour plusieurs raison, dans [12] on en cite les suivantes :

1- La clé publique de l'entité est compromise ;
2- L'entité n'est plus certifiée auprès de la CA ;
3- Le certificat de la CA est compromis.

Une liste de révocation doit contenir : le nom de l'émetteur (*Issuer name*), la date de sa création, la date à laquelle est prévue la publication de la prochaine liste et une entrée pour chaque certificat révoqué, contenant son numéro de série, ainsi que la date de révocation.

2.2.7.4. Les Infrastructures à clé publique (PKI)

Le problème lors de l'utilisation d'une clé privée est qu'il faut pouvoir gérer à la fois la génération et la distribution de cette clé. Quant à l'utilisation de clé publique, il est le plus question d'authentification. En effet, une entité devrait pouvoir vérifier l'authenticité d'une clé correspondante à une autre entité, avant de pouvoir l'utiliser pour un quelconque échange. C'est là qu'interviennent les infrastructures à clé publique PKI (***P**ublic **K**ey **I**nfrastructure*).

Dans la RFC 2822 (***I**nternet **S**ecurity **G**lossary*) une PKI est définie comme étant *un ensemble de matériel, logiciel, personnel, politique, et procédures, nécessaires pour la création, gestion, stockage, distribution et révocation des certificats basés sur la cryptographie asymétrique* [12].

Dans les PKI il s'agit de [27]:

- Une autorité de certification dont le rôle est de vérifier les certificats ;
- Une entité d'enregistrement qui vérifie au préalable les certificats pour l'autorité de certification ;
- Un ou plusieurs répertoires où sont stockés les certificats ;
- Un système de gestion de certificats.

Le groupe de travail PKIX (***P**ublic **K**ey **I**nfrastructure **X**.509*) à l'IETF (***I**nternet **E**ngineering **T**ask **F**orce*), monté en 1995 [15], avait défini un modèle d'élaboration d'une architecture basée sur la notion de certificat en se basant sur le standard X.509.

Les éléments suivants sont ceux constituants un modèle PKIX [12]:

1- *End Entity* : terme désignant l'utilisateur final, un serveur, un routeur ou toute autre entité qui peut être identifiée dans le champ *sujet* du certificat.
2- *Autorité de certificat* (*CA*): L'entité délivrant les certificats mais aussi les listes de révocation (CRL). L'enité CA peut aussi avoir à sa charge une variété de tâches d'administration qu'elle délègue généralement à une ou plusieurs entités d'enregistrement.
3- *Entité d'enregistrement RA* (***R**egistration **A**uthority*): élément optionnel, pouvant décharger le CA de certaines fonctions. L'entité d'enregistrement est en général liée à l'entité *End Entity*.
4- *CRL issuer* : élément optionnel que l'entité CA peut déléguer pour délivrer la liste de révocation *CRL*
5- *Dépôt (Repository)* : terme générique désignant toute méthode de stockage des certificats et des listes de révocation, et permettant à toute entité de pouvoir le récupérer.

Conclusion

L'engouement qu'a suscité l'utilisation des réseaux, et les différentes applications qui s'en dégagent : e-banking; e-commerce ; …, a multiplié les risques d'espionnage de l'information. La cryptographie est la solution qui permet d'assurer l'un des buts majeur de la sécurité de l'information, en l'occurrence la confidentialité.

La sécurité de l'information ne peut être assurée seulement en assurant la confidentialité des données, d'autres briques doivent être combinées, on parlera alors d'authentification ; d'intégrité ; de non répudiation…Chacun de ces buts peut être assuré via une technique précise, c'est ce dont il a été question dans ce chapitre où nous avons revu les différentes techniques pouvant assurer ces buts, allant du chiffrement aux empreintes numériques, en passant par le tatouage numérique et la signature numérique.

Quand il s'agit de chiffrement, ce qui assure principalement la confidentialité de l'information, la clé à utiliser peut être la même pour le chiffrement et le déchiffrement, on parlera alors de chiffrement symétrique. Alors que si deux clés différentes sont utilisées pour le chiffrement et le déchiffrement, on parlera de chiffrement asymétrique.

Quoi qu'il en soit, chiffrement symétrique ou asymétrique, les participants doivent bénéficier de manière sûre de la ou les clés à utiliser. Qu'il s'agisse de deux participants, ou d'un groupe de participants, faire parvenir la clé aux participants légitimes est en soi une problématique à part entière. Différentes stratégies de distribution, de partage et de gestion de clé ont été proposées afin de résoudre au mieux cette problématique. Ce sont ces stratégies et protocoles que nous allons revoir en détails dans le chapitre suivant.

Chapitre 2 Gestion des clés

« Si tu révèles ton secret au vent, tu ne dois pas lui reprocher de le révéler à l'arbre. »

Khalil Gibran (Le sable et l'écume)

Introduction

Si une sécurité parfaite est impossible à assurer, il est, néanmoins, possible d'aboutir à un certain niveau de sécurité de l'information en combinant différentes techniques, allant du chiffrement à la signature numérique et le certificat numérique. Le chiffrement constitue en soi une solution pour assurer au moins la confidentialité de l'information. Toutefois, pour atteindre une confidentialité, il faut s'assurer que la clé ou les clés de chiffrement utilisées, respectivement en chiffrement symétrique ou asymétrique, soient disponibles au niveau des participants légitimes. Plus encore, si un changement a eu lieu au sein du groupe, tel un départ (ou expulsion) ou une adhésion d'un membre, un changement de clé est obligatoire afin d'assurer dans le premier cas une confidentialité future, et dans le second une confidentialité passée.

La génération de la clé, sa distribution entre les membres, son changement en cas de dynamique dans le groupe, toutes ces opérations rentrent dans le cadre de la gestion de clé qui est une primitive très importante pour assurer un volet de la sécurité de l'information qu'est la confidentialité.

La confidentialité de l'information est donc intimement liée à cette information qu'est la clé et il apparait en conséquence qu'une importance particulière doit être accordée à cette information. Pour ce, plusieurs protocoles ont été proposés afin de résoudre la problématique de gestion de clés et particulièrement sa distribution. Cela va de l'hybridation entre chiffrement asymétrique et chiffrement symétrique, à l'autorité de confiance, et bien d'autres protocoles plus élaborés, jusqu'à même des solutions basées sur les lois de la mécanique quantique.

Dans le présent chapitre, nous nous attardons sur la problématique de la gestion de clé, nous présentons les différentes solutions proposées pour la distribution de clé symétrique, asymétrique et plus encore la distribution de clé dans un groupe et non plus entre seulement deux participants.

1. Gestion des clés

Qu'il soit question de chiffrement symétrique ou asymétrique, il est toujours important d'avoir une clé (ou plus) partagée de manière sûre et sécurisée entre les participants d'une communication. Le nombre de clés utilisées dans un groupe de communication est très variable, par conséquence, une gestion des clés est préconisée. Notamment lorsque le groupe de communication est sujet à une certaine dynamique, dans ce cas, il s'avère que la clé a une importance primordiale dans le contexte de la sécurité de l'information, dans le sens où elle doit être gérer de manière à assurer aussi bien la confidentialité passée que future.

Le principe de KERCKHOFFS, introduit dans la section 2.2.1, affirme cette hypothèse.

1.1. Définitions

Suivant Alfred J. Menzer [14] :

i. *L'établissement*[10] d'une clé est un processus ou un protocole[11] via lequel un secret partagé devient disponible pour deux parties ou plus, et ce à des fins de cryptage ultérieurs. Un protocole ou un mécanisme de *transfert* de clé est une technique où une partie crée ou obtient une valeur secrète et la transfert secrètement à d'autres parties.

ii. Un protocole ou un mécanisme *d'accord* de clé est une technique via laquelle un secret partagé est dérivé, par deux parties (ou plus), comme une fonction d'informations fournies par, ou associées à, chacune d'elles (idéalement) de telle sorte qu'aucune partie ne puisse déterminer, à l'avance, la valeur résultante.

iii. Un *Schéma de pré-distribution de clé* est un schéma d'établissement de clé via lequel la clé résultante est complètement déterminée *a priori*. Par contre, un schéma *dynamique* en est un où la clé établie (clé de session dans ce cas) entre deux parties (ou plus) varie au fur et à mesure des exécutions.

iv. Une relation de « *keying* » est un état de communication via lequel des entités partagent une donnée commune afin de faciliter des techniques cryptographiques. Cette donnée peut contenir une clé publique ou secrète, des valeurs d'initialisation, ainsi que d'autres paramètres non secrets.

v. *La gestion de clé* est l'ensemble de techniques et de processus comportant l'établissement et la maintenance d'une relation de « *keying* » entre des parties autorisées. La gestion des clés renferment plusieurs techniques et procédures comportant :

   a. Initialisation des utilisateurs du système dans ce domaine ;
   b. Génération, distribution, et installation du matériel de « *keying* » ;
   c. Contrôle d'utilisation du matériel de « *keying* » ;
   d. Mise à jour, révocation, et destruction du matériel de « *keying* » ; et
   e. Stockage, sauvegarde/restauration et archivage du matériel de « *keying* » ;

[10] L'établissement d'une clé est subdivisé en : 1) transport de la clé et 2) l'accord de la clé. [14]

[11] Un protocole étant un algorithme multi-partie, définie comme une séquence d'étapes spécifiant précisément les actions à entreprendre entre deux parties ou plus, afin d'aboutir à un objectif spécifié [14].

1.2. La distribution des clés

Il est possible d'avancer la thèse suivante : « la sécurité de l'information est liée au chiffrement de cette dernière de manière sûre ». En chiffrement, qu'il soit symétrique ou asymétrique, il est question d'utiliser une « clé », qui, à son tour doit être sûre. A cet effet, une clé doit être générée et distribuée de manière sécurisée. Dans ce qui suit nous aborderons les méthodes de distribution de clé dans le contexte du chiffrement symétrique aussi bien qu'asymétrique.

1.2.1.Distribution de clé symétrique

Rappelons qu'un chiffrement symétrique nécessite la possession de la même clé par les interlocuteurs. La même clé sera utilisée pour le chiffrement et le déchiffrement de l'information. Pour ce, plusieurs solutions sont proposées. On en cite :

1.2.1.1. Distribution physique

La solution la plus intuitive, à la problématique de distribution de clé de chiffrement symétrique, serait un échange physique. Les entités qui veulent communiquer de manière sécurisée doivent se rencontrer au préalable et s'échanger la clé de chiffrement.

La solution de distribution physique de la clé de chiffrement, même simpliste, exige une confiance mutuelle des deux entités. D'autre part, si une des deux parties est kidnappée ou tuée, la sécurité du protocole est remise en question [18].

Une autre version de la distribution physique de la clé de chiffrement est d'introduire une tierce partie de confiance. La tierce partie sera alors chargée de la génération et la livraison (distribution) de la clé, manuellement, à chacune des deux parties [12].

La situation de partage de clé symétrique, peut s'avérer plus compliquer si le nombre de parties communicantes va en augmentant dans un système devant assuré un chiffrement point-à-point. En effet, si $n$étant le nombre d'entités communicantes dans un système distribué, alors $n(n-1)/2$ serait le nombre de clés à partager dans ce système, ce qui devient rapidement difficile à gérer si n est grand.

1.2.1.2. Distribution via une clé pré-partagée

Si une clé est partagée, au préalable, entre un nombre d'entités, elle pourra servir à la distribution de la nouvelle clé de chiffrement [18] [17]. Soit A et B deux entités qui se partagent une clé K_1. La partie A peut alors initier la génération d'une nouvelle clé K, qu'elle partage avec B en la chiffrant utilisant K_1.

La figure suivante montre l'idée de base d'un tel scénario.

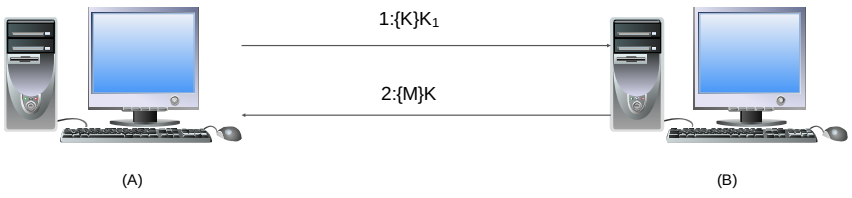


Figure 15 Distribution de clé de chiffrement symétrique via une clé pré-partagée.

Dans [17] une amélioration du schéma est proposée. Le but étant de garantir d'une part une authentification explicite de A et d'éviter une attaque par *rejeu*[12]. Le principe est d'ajouter au paquet $\{K\}_{K1}$ d'autres informations comme un numéro de séquence (N) ou une estampille (T), et l'identité de B Id_B. Le schéma est alors comme suit :

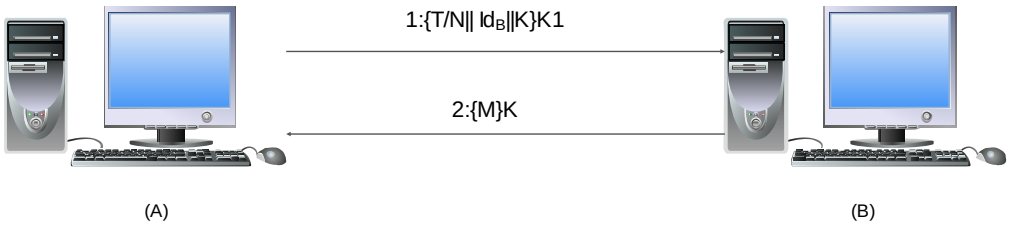


Figure 16 Schéma amélioré de la distribution de clé symétrique.

Une variante de ce scénario étant l'introduction d'une tierce partie de confiance CDC (un ***C**entre de **D**istribution de **C**lé*). Ainsi la tierce partie qu'est le CDC, qui partage une clé avec chacune des entités A et B (respectivement K_A, K_B), peut utiliser ces clés pour chiffrer et distribuer aux deux entités A et B, une clé mutuelle K_s, qu'il se chargera de générer.

La figure suivante, inspirée d'un schéma dans [12] illustre la situation qu'on vient d'introduite.

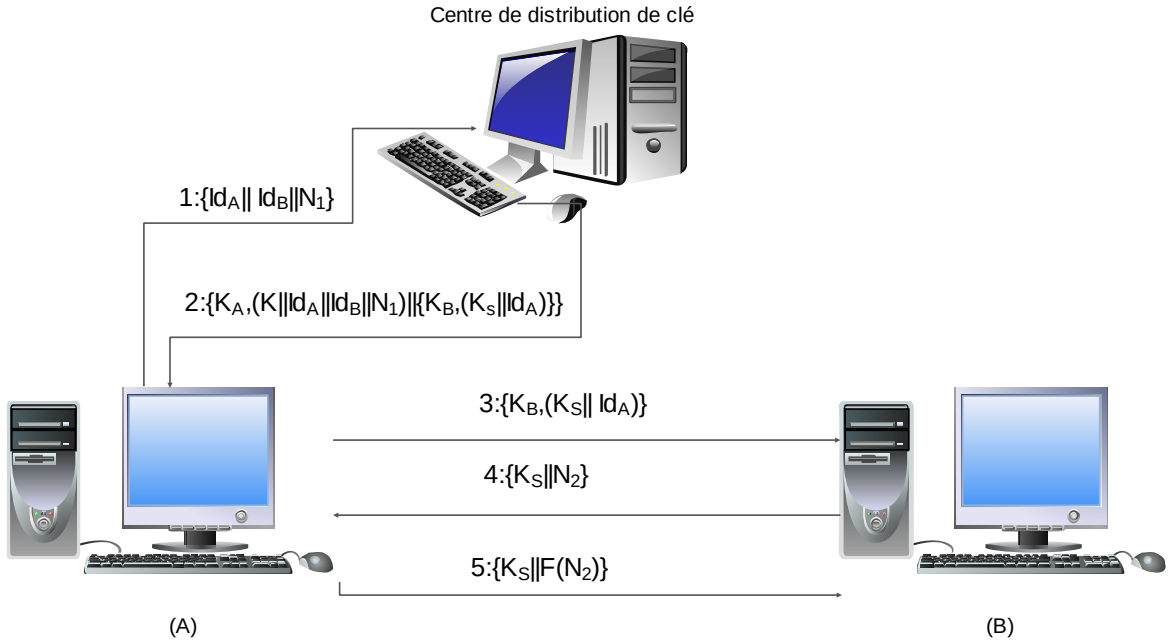


Figure 17 Distribution de clé de chiffrement symétrique via Centre de distribution de clé.

Le modèle que montre la figure précédente est appelé *pull*. L'initiateur de la communication s'adresse en premier lieu au CDC pour récupérer la clé symétrique. Ce dernier se charge de la génération de la clé. Il est aussi possible de procéder autrement. L'initiateur de la communication s'adresse à son destinataire qui à son tour va récupérer la clé symétrique auprès du CDC. Il s'agit là d'un modèle *push*.

[12] Rejeu : réutilisation des mêmes informations pour une authentification, tandis que ces dernières ne sont plus valides.

Une combinaison des deux modèles est aussi possible pour l'obtention d'un modèle dit *mixte* [17].

En fait, l'introduction d'un centre de distribution de clé, implique une hiérarchisation des clés [12]. La clé que partage le centre de distribution avec chacune des entités et qui sert à la distribution des autres clés, est une clé dite *clé master* (Sur la figure : K_A, K_B), alors que les clés qu'utilisent les entités pour communiquer entre elles, et qui sont temporaires, sont dites *clé de session* (Sur la figure précédente : K_S).

Dans Figure 17, L'entité A initie l'opération de distribution de clé, en envoyant une requête contenant l'identificateur de l'entité B, ainsi qu'un Nonce[13], au centre de distribution. Le centre répond à la requête de A par un paquet contenant une partie dédiée à B (voir étape 2-Figure 17). A transfère cette dernière à B, qui en la déchiffrant avec sa clé K_B, récupérera la clé de session K_S qui sera utilisée pour chiffrer un Nonce et l'envoyer à A. A confirmera la réception en modifiant le Nonce reçu de B et en renvoyant le résultat chiffré via K_S.

Dans un groupe plus étendu, il serait possible d'introduire plus d'un centre de distribution de clé, de sorte, que chaque centre gère la distribution de clés dans un sous-groupe [28] [12]. Ainsi, un CDC peut générer une clé de session pour des entités du sous-groupe dont il est responsable, comme il a été introduit dans le paragraphe précédent. Si une entité veut communiquer avec une entité se trouvant dans à autre sous-groupe, les CDC respectifs des sous-groupes auxquels elles appartiennent se chargent alors de la distribution d'une clé de session en passant par un CDC supérieur.

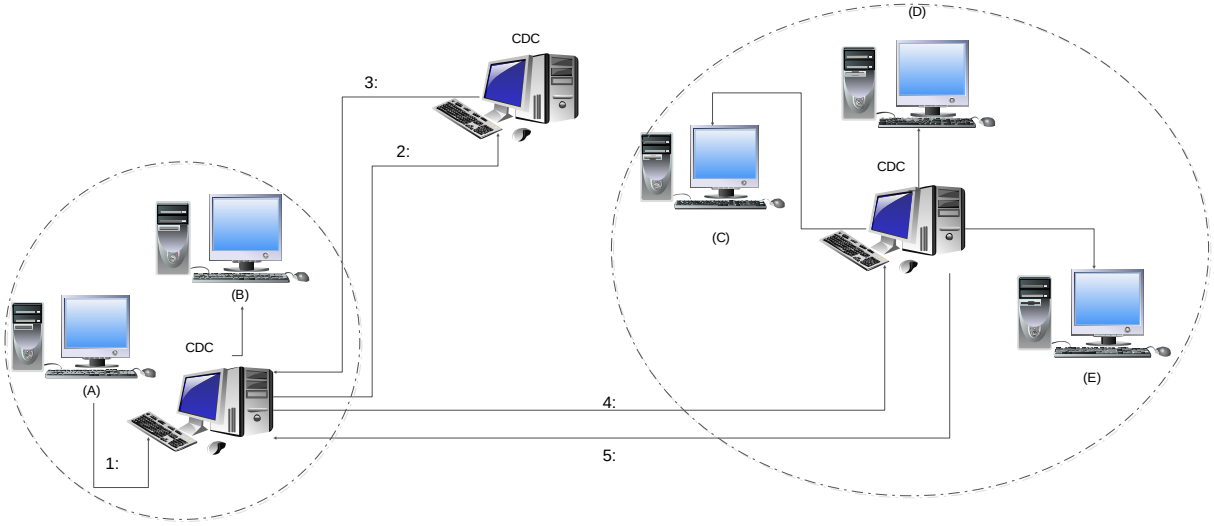


Figure 18 Hiérarchie de CDC.

Avec la mise en place d'une hiérarchie de CDC les échanges de clés sont restreints aux sous-groupes. Si un attaquant récupère une clé de session, ou même une clé master d'un sous-groupe, seule la sécurité de ce sous-groupe est remise en question, et non celle de tout le groupe.

Il est toutefois clair que le schéma de distribution de clé via une clé pré-partagée n'assure la sécurité de la distribution qu'en partie. La problématique reste toujours posée au niveau de la distribution de cette clé pré-partagée, en l'occurrence la clé master.

[13] Nonce : *Number used only once*, qui peut être un nombre aléatoire, un compteur…différent à chaque requête et utilisé une seule fois.

1.2.1.3. Distribution via une clé publique

Un autre schéma de la distribution de clé symétrique consiste à utiliser une clé publique [18] le schéma est aussi appelé *enveloppe digitale* [17], ou encore schéma hybride [12]. Ainsi si A et B veulent distribuer une clé de session, A génère une paire de clé privée/publique (en utilisant l'algorithme RSA par exemple), garde la clé privée (K_{pr}) et partage la clé publique (K_{pb}). B peut alors générer une clé de session (K), qu'il chiffre avec la clé publique de A et la lui envoie (voir page 33). Seule l'entité A peut déchiffrer le message et donc récupérer la clé de session. A et B peuvent maintenant utiliser la clé de session pour le chiffrement.

Le schéma distribution est illustré sur la figure ci-dessous.

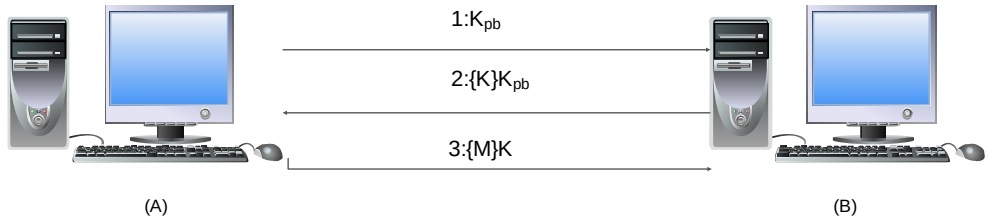


Figure 19 Distribution de clé symétrique via une clé publique.

Le schéma peut être mis en place lorsque les deux entités ne font pas confiance à une tierce entité intermédiaire. Cependant, il est a signalé ici que la sécurité d'un algorithme, tel que le RSA, est conditionnée par la puissance de calcul dont dispose un adversaire.

Un autre point faible de ce scénario étant que ce dernier ne peut résister à une attaque de type Man-in-the Middel. En effet, si un adversaire E, qui a crée sa paire de clé privée/publique (K_{prE}/ K_{pbE}), se met entre A et B et intercepte la communication à l'étape 1, il pourrait envoyer sa clé publique K_{pbE} à B. B générera la clé de session, la chiffre en utilisant K_{pbE} et l'envoie à E, ce dernier l'intercepte et récupère K en déchiffrant le message reçu de B en utilisant K_{prE}, puis E chiffre une seconde fois K en utilisant la clé publique de A et la lui envoie. Ainsi, l'espion E aura récupéré la clé de session à l'insu de A et B.

C'est dire aussi que l'authentification des entités n'est nullement assurée. Il est possible alors d'améliorer le schéma en introduisant un mécanisme de signature numérique, mais aussi d'autres informations pour éviter le rejeu. Le tout se fait à l'étape (2 :) du schéma représenté dans la figure précédente.

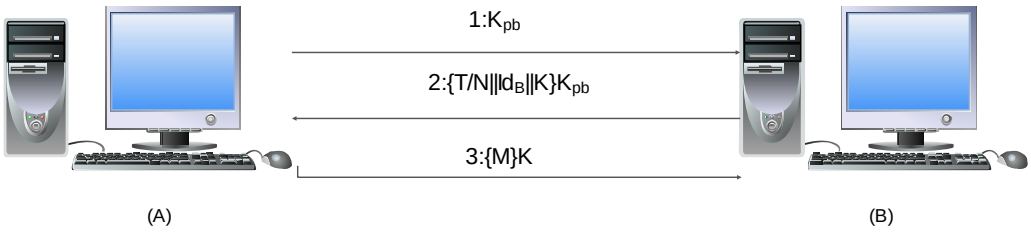


Figure 20 Schéma amélioré de la distribution de clé symétrique via une clé publique.

Ainsi, à l'étape 2 du schéma amélioré (voir figure ci-dessus), B ajoute au paquet contenant la clé, une estampille (T) ou un numéro de séquence (N), et l'identité de A. B signe le tout est le chiffré via la clé

publique de A, K_{pb}. A la réception, A déchiffre le paquet via sa clé privée K_{pr}. A vérifie l'intégrité et l'authenticité de la source ainsi que l'estampille ou le numéro de séquence. Si tout est satisfaisant A acceptera la clé secrète K qui lui a été transmise.

1.2.1.4. Distribution de clé par accord

« *Un mécanisme de mise en accord est un procédé qui permet d'établir une clé partagée entre plusieurs entités de telle sorte qu'aucune d'entre elles ne puissent établir sa valeur par avance* » [17]

En effet, dans un mécanisme par accord, chaque entité apporte une contribution pour l'élaboration d'une clé secrète qui n'est pas connue au préalable. Le protocole Diffie-Hellman en est un exemple standard.

Proposé par WHITFIELD DIFFIE et MARTIN E. HELLMAN, en 1976 [29] . Les détails du protocole sont décrits dans la RFC 2631. La sécurité de l'algorithme Diffie-Hellman repose sur la difficulté de calcul du logarithme discret sur un corps fini par rapport au calcul de l'exponentielle dans le même corps.

Nous introduisons le principe du protocole avec l'exemple suivant :

1. Soit A et B les entités devant générer en accord un secret (la clé en l'occurrence) ;
2. Soit P un nombre premier public, et W un nombre primitif[14] de Z_p^x / 1<W<P. Soit W=7 et P=11 ;
3. A génère un nombre aléatoire a / 1 < a < P, soit a=3 par exemple. B fait de même b=6 ;
4. A calcule $\alpha = W^a mod\ P = 7^3 mod\ 11 = 2$, B de son côté calcule $\beta = W^b mod\ 11 = 7^6 mod\ 11 = 4$;
5. A et B s'échange publiquement les résultats obtenus à l'étape 4 ;
6. A calcule $4^3 mod\ 11 = 9$, B de sont côté calcule $2^6 mod\ 11 = 9$;
7. A et B obtiennent le même résultat, ce qui revient à la propriété mathématique suivante :

$$[W^a]^b mod\ P = [W^b]^a mod\ P$$

Ainsi A et B auront à la fois génèrent et distribuent la même clé qui peut être utilisée maintenant comme clé symétrique.

La force du protocole réside dans le fait que la classe d'équivalence Zn / n∈ N est infinie. Un espion qui intercepte la valeur 2 (ou 4) échangée publiquement n'aura aucune information utile même sachant W et P.

Le protocole peut être étendu à plus de deux participants, mais nécessite dans ce cas de figure K -1 round si K est le nombre de participants au protocole.

Le schéma suivant reprend le protocole avec 3 participants A, B et C. Le protocole prend dans ce cas de figure 2 rounds.

On suppose dans notre exemple que P=23 et W=3. On suppose également que les trois participants A, B et C génèrent, respectivement, trois nombres aléatoires 4, 3, et 6 qu'ils gardent secrets.

[14] Soient p premier et a < p. On dira que a est un primitif de p si ∀ b∈ (1, p-1), ∃g tel que $a^g \equiv b\ mod\ p$.

Le premier round peut être lancé où chaque entité calcule $w^x mod\ P / x \in \{3,4,6\}$. Chaque entité parmi les trois envoie le résultat de cette opération à l'entité suivante. A la fin de ce premier round, les entités A, B et C auront reçues respectivement les valeurs suivantes : 16, 12, et 4.

Le deuxième round est lancé à l'étape 4. Les entités vont réaliser un calcul semblable à celui du premier round, c-à-d que chaque entité calcule : $y^x mod\ P$ avec $x \in \{2,3,9\}$ et $y \in \{4,12,16\}$.

A la fin du deuxième, et dernier, round (étape 6' voir figure ci-dessous), les trois participants A, B et C ont les valeurs respectives 2, 9 et 3.

A l'étape 7 (voir figure ci-dessous) les participants A, B et C obtiennent tous les trois la même valeur (en l'occurrence 16) suite à l'exécution de l'opération $z^x mod\ P / x \in \{3,4,6\}\ et\ z \in \{2,3,9\}$.

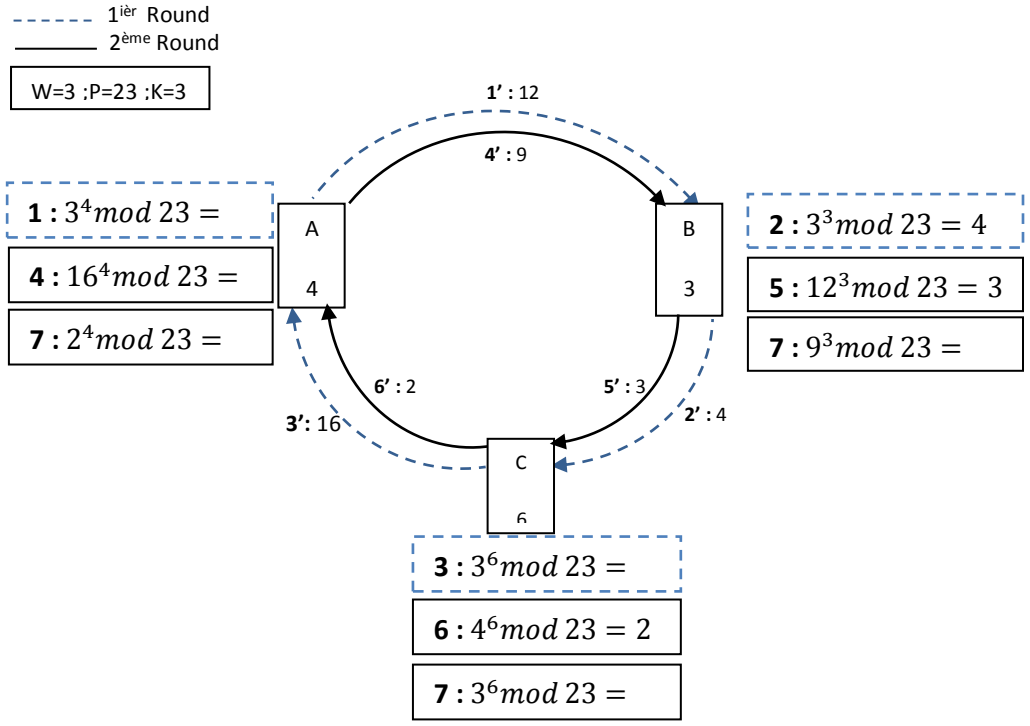


Figure 21 Diffie-Hellman avec trois participants.

1.2.1.5. Distribution de clé via le partage de secret

Deux cas se présentent :

Soit que l'on crée un secret et on le partage puis on le reconstruit. Le secret est ainsi partagé en $\mathcal{N}$ parties qui seront envoyées à $\mathcal{N}$ participants par des voies différentes. Pour reconstruire le secret, les $\mathcal{N}$ parties doivent collaborer.

La technique peut être utilisée pour la distribution d'une clé symétrique dans un groupe. Cette dernière est divisée en plusieurs parties, et chaque entité du groupe reçoit une partie du secret partagé qu'est la clé. Pour reconstruire la clé, toutes les entités, ou une partie d'entre elles, suivant la technique de distribution, doivent collaborer pour restituer la clé.

Le deuxième cas étant que le secret n'existe pas mais les entités le créer chacune de sont côté, comme il sera introduit dans la section suivante.

1.2.1.5.1. Partage de secret basée XOr

Présentée comme solution au problème de distribution de clé, le partage de secret se veut une solution élégante [30] [18].

Le schéma se base sur l'opération de XOr et nécessite l'introduction d'une tierce partie de confiance. Cette dernière génère une chaine aléatoire $\mathcal{P}$, qui sera combinée avec le secret $\mathcal{S}$ via l'opération XOr. Le résultat de l'opération étant $\mathcal{R}$.

$$R=P\oplus S$$

Équation 1 Partage se secret basé XOr.

La partie de confiance envoie $\mathcal{P}$ et $\mathcal{R}$, respectivement, aux entités A et B. Pour retrouver le secret $\mathcal{S}$, A et B doivent collaborer pour combiner leurs parties respectives ($\mathcal{P}$ et $\mathcal{R}$) par la même opération XOr.

$$S=P\oplus R$$

Équation 2 Recouvrement de secret.

Il est alors possible de considérer $\mathcal{S}$ comme la clé que cherchent à partager A et B. Après l'opération de XOr, $\mathcal{S}$ est la clé est partagée entre A et B qui peuvent l'utiliser pour le chiffrement d'autres informations.

Le schéma est sûr, dans le sens où l'adversaire, E, qui intercepte $\mathcal{P}$ ou $\mathcal{R}$, ne pourra pas resituer le secret $\mathcal{S}$. Par contre, le schéma est vulnérable devant une attaque collaborative qui permettra d'intercepter à la fois $\mathcal{P}$ et $\mathcal{R}$, et donc de restituer $\mathcal{S}$. D'autre part, le schéma requiert une collaboration des entités A et B, donc la mise en place d'un mécanisme d'authentification, et de contrôle d'intégrité des parties échangées ce qui nous ramène au point de départ.

1.2.1.5.2. Partage de secret schéma de Shamir & Blakley

En 1976, A. Shamir introduit l'idée du partage de secret dans son papier « How to share a secret ». On reprend ici l'exemple introductif de A. Shamir, inspiré des travaux de Liu [31].

Soient onze (11) scientifiques qui travaillent sur un projet secret, et qu'ils veulent sécuriser les documents du projet dans un bureau de sorte qu'il ne puisse être ouvert que si et seulement si six (06) scientifiques sont présents. Quel est alors le plus petit nombre de cadenas ? Quel est aussi le nombre plus petit de clés de cadenas que doit avoir chaque scientifique ?

La solution est qu'il faut 462 cadenas et 252 clés de cadenas par scientifiques. Il est clair qu'autant de cadenas et de clés de cadenas n'est pas chose facile à gérer. A. Shamir généralise le problème au fait qu'une information secrète est divisée en $\mathcal{N}$ parties de sorte que [31]:

1- La connaissance de k parties, ou plus, parmi les n permette de calculer facilement S ;
2- Que la connaissance de k-1 parties, ou moins, laisse S complètement indéterminée.

L'idée ainsi introduite par A. Shamir est dite schéma à seuil, que l'on note (k, n) et se base sur l'interpolation polynomiale. Plus formellement, soit k (x_1,y_1), (x_2,y_2),… (x_k,y_k) points sur un plan à deux (02) dimensions, avec des x_i distincts. Il existe alors un, et un seul polynôme q(x) de degré k-1 tel que : $q(x_i)=y_i$ pour tout i.

Pour partager l'information *D* en parties D_i on choisit un polynôme de degré k-1 tel que :

$$q(x) = a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}$$

Où $a_0=D$;

D_1 =q(1) ,…, D_i =q(i),… D_n =q(n) .

Avec n'importe quel ensemble k de D_i il est possible de recouvrir, par une interpolation de Lagrange les coefficients de q(x) puis de calculer D= q(0).

L'idée est efficace, tel que le mentionne A. Shamir dans son papier, dans le cas d'un groupe où des entités, mutuellement suspectes, avec des intérêts conflictuels doivent coopérer.

Dans le contexte de la distribution de la clé, l'information D est la clé à gérer. Une entité centrale peut alors prendre en charge les opérations de génération de D, de q(x) le polynôme de degré k-1 et la distribution de différentes parties aux entités d'un groupe. Pour utiliser la clé, k membres du groupe doivent collaborer pour la restituer.

L'idée du partage de l'information entre k entités d'un groupe a été introduite simultanément par G.R Blakley dans un papier paru la même année [32]. Se dernier propose par contre une résolution d'un système à k-1 équations.

G.R Blakley introduit la notion de parties *volatiles* qui seraient détruites en cas d'attaque. Comme une attaque peut forcer la destruction des parties volatiles, il serait intéressant de confier ses parties à des entités de confiance. G.R Blakley évoque cependant trois cas de figures où ces parties, ainsi dites non-volatiles, doivent être protégées [32] :

a. Abnégation

Il s'agit là d'un incident suite auquel la partie non-volatile n'est plus tout à fait réclamée par l'entité qui l'a confié à la partie de confiance. G.R Blacley dégage trois types d'abnégation : la **d**estruction, la **d**égradation ou la **d**éfection[15].

b. Trahison

Il s'agit là d'un incident suite auquel la partie non-volatile et totalement connue par l'adversaire. La **d**éréliction est le genre principal de trahison, où la partie de confiance dévoile la partie non-volatile, dont elle était supposée assurer la sécurité, à l'adversaire de sorte que la partie légitime ne puisse découvrir la trahison ni avant ni après l'avoir dévoilée.

[15] Défection : l'entité de confiance est compromise, elle dévoile la partie non-volatile à l'adversaire et refuse de la donner à l'entité qui la lui a confiée.

c. Combinaison d'incidents

Une abnégation qui est aussi une trahison, la défection en est l'exemple, car en plus d'être une abnégation il s'agit d'une trahison de la part de la partie de confiance.

1.2.2.Distribution de clés asymétrique

Dans [12] les différentes techniques de distribution d'une clé publique sont regroupées dans les catégories suivantes :

a- Annonce publique ;
b- Un répertoire à accès public ;
c- Une autorité à clé publique;
d- Un certificat de clé publique.

Dans ce qui suit, nous allons revoir ces différentes techniques en détails :

1.2.2.1. Annonce publique

L'idée de l'annonce publique suit exactement la philosophie du chiffrement à clé publique. Une clé publique est mise à disposition de toutes les entités d'un groupe. Toute entité A désirant communiquer avec une autre entité B, n'a qu'à chiffrer le message en utilisant la clé publique de B, qu'elle a diffusé au préalable.

Dans une telle configuration, *tout le monde peut envoyer des messages à tout le monde*. Cela peut conduire à une saturation d'un côté, d'un autre côté, un tel scénario n'est pas à l'abri d'une attaque par mascarade. Une entité X peut diffuser une clé publique K_x prétendant être A. B utilisera alors K_x pour communiquer à X tout le trafic qui devait être transmis à A. La situation est maintenue jusqu'à ce que A signale la situation à B. Il est clair que le scénario ne peut être utilisé tel qu'il est présenté et nécessite la mise en place d'un mécanisme d'authentification.

1.2.2.2. Un répertoire à accès public

Un répertoire structuré contenant l'identificateur de chaque entité, auquel est associée une clé publique est géré par une entité de confiance. Le répertoire est à accès publique. Ainsi l'entité A qui veut communiquer avec B, doit d'abord récupérer la clé publique de cette dernière depuis le répertoire.

L'entité de confiance a à sa charge l'enregistrement des couples (identité, clé publique) au niveau du répertoire et de les maintenir à jour. En effet, une entité devrait pouvoir changer sa clé publique si cette dernière a été utilisée pour le chiffrement d'un trafic assez conséquent ou si la clé privée correspondante a été compromise.

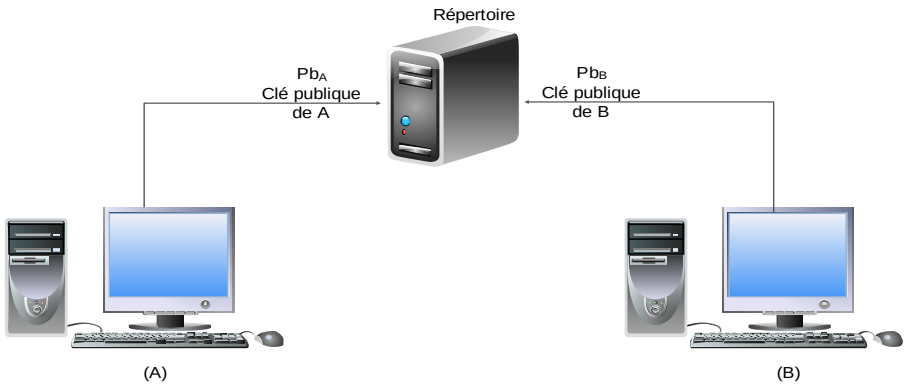


Figure 22 Distribution de clé publique via un répertoire à accès publique.

L'utilisation d'un répertoire publique est bien meilleure que la diffusion individuelle des clés publique, cela dit le scénario souffre d'un inconvénient qu'est la centralisation de la gestion du répertoire. Toute la sécurité est remise en question si l'entité de confiance est compromise.

1.2.2.3. Autorité avec clé publique

Pour renforcer le scénario précédent il est possible de munir l'entité centrale (que l'on notera C) chargée de la gestion du répertoire, d'une paire de clé privée/publique. L'entité A voulant communiquer avec B, devra envoyer une requête à C lui demandant la clé publique de B (KB_{Pb}). La requête de A sera chiffrée avec la clé publique de C (KC_{Pb}). C répondra par un message contenant à la fois la clé publique de B et d'autres informations [12]. A peut déchiffrer le message-réponse de C via la clé publique de C, récupère la clé publique de B tout en s'assurant que c'est bien C qui la lui a transmise.

Il est clair que toute entité ayant la clé publique de C, qui peut éventuellement être corrompue, peut récupérer la clé publique de B en déchiffrant le message-réponse de C. D'autre part, la sollicitation de l'entité centrale peut vite s'avérer un goulot t'étranglement si le trafic est intense. Il est alors possible d'opter pour une autre alternative, celle des certificats.

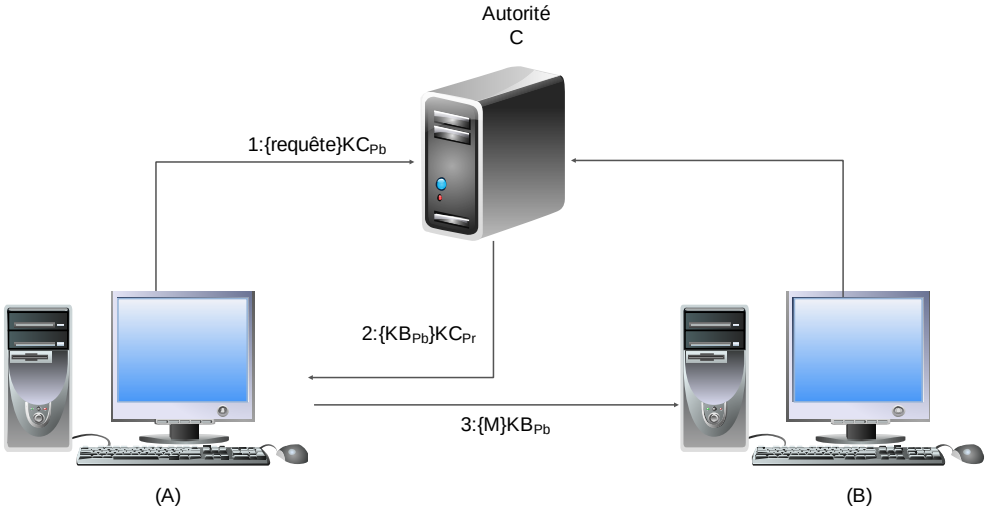


Figure 23 Distribution de clé publique via une autorité à clé publique.

1.2.2.4. Certificat 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 (voir 2.2.7.1 au chapitre précédent). Ainsi l'entité A voulant communiquer avec B récupérera son certificat où elle trouvera la clé publique de B mais pourra aussi vérifier son authenticité auprès de l'autorité de certificat.

1.2.3.Gestion et distribution de clé dans un groupe

Dans un protocole unicast sécurisé, les deux protagonistes peuvent s'authentifier respectivement et communiquer de manière sécurisée, en utilisant un algorithme de chiffrement à clé symétrique (utilisation de la même clé aussi bien pour le chiffrement que pour le déchiffrement) ou asymétrique (utilisation de deux clés différentes, l'une servant au chiffrement, l'autre au déchiffrement). Une telle solution n'est pas très efficace et ne peut être adoptée lorsqu'il s'agit de plusieurs récepteurs (ce qui est le cas d'une communication multicast), qui de plus, sont dynamiques (nouveau membre qui adhère et/ou ancien membre qui quitte ou est expulsé du groupe).

En effet, un protocole multicast n'assure pas, à lui seul, l'authentification et le contrôle d'accès des membres du groupe. De plus, l'adresse multicast du groupe est publique, ainsi toute entité désirant recevoir le flux multicast, n'a qu'à rejoindre le groupe, ce qui augmente considérablement les risques d'espionnage. Plus précisément le risque d'attaque passive, où un espion rejoint le groupe et accède au flux qui y circule sans provoquer une quelconque perturbation [33].

Aussi avec l'absence d'un contrôle d'accès, n'importe quel membre peut inonder le groupe multicast, ce qui peut provoquer une situation de déni de service. Plus encore, en l'absence d'un contrôle d'accès, un membre peut mener le groupe à une situation de mascarade et bien d'autres formes d'attaques active [33].

De ce fait, et de par cette nature dynamique d'un groupe multicast, il est impératif d'assurer, entre autres, deux conditions : La confidentialité passée et la confidentialité future.

Lorsqu'un élément nouveau rejoint le groupe, il faut s'assurer que le flux de données échangées avant son adhésion ne lui soit accessible. Autrement dit, il faut qu'il y ait changement de clé après l'adhésion d'un nouvel élément ; il s'agit là de la confidentialité passée. De même, lorsqu'un élément quitte le groupe (ou en est exclus) un changement de la clé de chiffrement doit être opéré afin que cet élément ne puisse accéder aux données échangées après son départ. C'est là, la confidentialité future.

La confidentialité étant un concept assuré par des mécanismes cryptographiques, mettant en jeu une information dont l'importance a été préalablement soulignée par le principe de KERCKHOFFS. Le principe, introduit par Auguste KERCKOFFS en 1883 [19] [14]; stipule que la sécurité d'un système de sécurisation ne doit pas dépendre *seulement* de l'algorithme de chiffrement, mais plutôt du choix de cette même information, à savoir la clé de chiffrement.

Cette dernière devant être gérée d'une manière efficace et sécurisée, le but de la gestion est alors de [34]:

1- Fournir une identification et une authentification des membres ; l'authentification permet d'éviter toute sorte d'attaque par personnification (personnification d'un membre légitime ou d'un gestionnaire de clé) ;
2- Le contrôle d'accès ; qui, lui, est réalisé afin de valider les membres d'un groupe avant même de leur concéder l'accès à la communication (en particulier l'accès à la clé) ;
3- Génération, distribution et installation des clés ; certainement il est nécessaire de changer la clé à des intervalles réguliers mais il faut également assurer l'indépendance des clés afin de ne pouvoir en déduire l'une de l'autre.

La problématique est alors de générer, partager/distribuer… de façon sécurisée la clé de chiffrement, permettant un échange sécurisé entre les membres autorisés d'un groupe. Les membres autorisés étant définis comme étant ceux ayant la clé de chiffrement, dite clé de groupe.

Afin d'aboutir à cette gestion de clé dans un groupe, plusieurs approches ont été proposées, et une autre taxonomie, qui leur est propre, s'impose. Elle se base sur la manière dont se fait la distribution de la clé, mais aussi en fonction des membres du groupe censés partager la clé (clé unique pour tout le groupe ou une clé pour chaque sous-groupe), et de leur dynamique (changement de clé suite à l'adhésion ou l'expulsion d'un membre).

Nous nous intéressons particulièrement à ces propositions, et nous reviendrons dans ce qui suit sur certaines de leurs classifications.

Comme nous l'avons introduit au préalable, et vu la nature d'un groupe multicast, un protocole de gestion de clé doit assurer la scalabilité [33], ainsi il est possible de distinguer dans la littérature plusieurs classifications des protocoles de distribution de clé extensibles:

1.2.3.1. Première classification

Dans [33] les auteurs distinguent, pour une clé commune de groupe, entre trois types de protocoles : Centralisé, Distribué, et Hiérarchique.

Dans un protocole *centralisé*, une seule entité se charge de la distribution de la clé de session au reste du groupe en la chiffrant pour chaque membre avec la clé correspondante. La mise à jour de la clé se fait à chaque fois qu'un membre quitte ou rejoint le groupe;

Le protocole *distribué* est celui où tous les membres ont le même degré de confiance, et il est à la charge des membres ayant joint le groupe en premier de générer les KEK (***K**ey **E**ncryption **K**ey*). Attribuer le même degré de confiance à tous les membres rend ce genre de protocole vulnérable à des attaques provenant de l'intérieur même du groupe.

La troisième classe est la classe des protocoles *hiérarchiques* ; sur ce point les auteurs reviennent sur deux catégories, la première se base sur une hiérarchie suivant *l'arbre des clés*, chaque membre reçoit toutes les clés KEK remontant à la racine de l'arbre ; la deuxième catégorie étant celle se basant sur *l'arbre des nœuds* où les auteurs introduisent la notion de sous-groupe pour gérer la distribution de la clé. L'adhésion d'un membre à un sous-groupe ou son départ n'affectera que le sous-groupe en question.

1.2.3.2. Deuxième classification

Dans [34] et [35] les auteurs présentent une autre répartition, toujours en trois classes, ils introduisent les nominations suivantes : Approche centralisée, décentralisées et distribuée.

Ils citent alors : les protocoles *centralisés* de gestion de clé de groupe, où la gestion est déléguée à une seule entité qui se charge à la fois du contrôle d'accès et de la distribution de clé. Le risque d'une telle approche étant que la sécurité du groupe repose sur celle de l'entité centrale. Si cette dernière est attaquée ou si elle tombe en panne, la sécurité du groupe est remise en question.

La deuxième classe étant les architectures *décentralisées*, c'est au niveau de cette classe que les auteurs introduisent pour leur part la notion de sous-groupe ; Dans ce cas de figure la gestion du groupe est départagée entre des gestionnaires de sous-groupes.

La dernière classe est celle des protocoles *distribués* de gestion de clé de groupe ; Dans cette classe, on ne fait appel à aucune entité particulière pour gérer les clés, les membres du groupe contribuent même à la génération de la clé de groupe.

Les auteurs dans [34] proposent également des facteurs d'évaluation pour chaque protocole de distribution de clé, que nous résumons dans le tableau qui suit :

Approche Centralisée	Architecture Décentralisée	Protocole distribué
▪ Besoins en termes de stockage ▪ Taille des messages ▪ Confidentialité passée et future ▪ Collusion[16]	▪ Indépendance des clés ▪ Contrôleur décentralisé ▪ Clé locale ▪ Clé Vs donnée (qui doivent êtres envoyées par des voies différentes) ▪ Confidentialité passée et future ▪ Type de communication (1-N ou N-N)	▪ Nombre de round. ▪ Nombre de message. ▪ Puissance de calcul nécessaire lors de la configuration/initialisation set up ▪ Utilisation de l'algorithme de Diffie-Hellman.

Tableau 2 Paramètres d'évaluation de l'efficacité d'un protocole de gestion de clé.

1.2.3.3. Troisième classification

Yacine Chellal et al. proposent, en 2005, une nouvelle taxonomie dans [36] où les auteurs distinguent entre deux classes de clé : clé commune par groupe TEK (***T**raffic **E**cryption **K**ey*) où tous les membres du groupe partagent la même clé, et clé indépendante par sous-groupe. Les auteurs maintiennent par contre le partitionnement en trois classes à savoir : centralisée, décentralisée et par accord.

1.2.3.3.1. Première classe

Les auteurs regroupent dans cette classe des protocoles de distribution de clé où les membres du groupe partagent la même TEK. Ils distinguent entre 3 sous-classes :

[16] Collusion : collaboration d'entités expulsées en s'échangeant leur informations individuelles pour retrouver l'accès à la clé de groupe [34].

Dans la première sous-classe, dite *centralisée*, il est toujours question d'une seule entité qui gère et distribue la clé. Suivant la technique utilisée pour distribuer la TEK, les auteurs distinguent entre 3 sous-catégories :

- ✓ Clé par paire : dans cette première sous-catégorie, une clé KEK est partagée entre le serveur de clé et chaque membre du groupe.
- ✓ Diffusion de secret : l'approche se base sur la diffusion du message au lieu d'une communication secrete point-à-point.
- ✓ Hiérarchie de clé : l'approche permet de diminuer le nombre de messages lors de l'opération de rekeying.

La deuxième sous-classe est celle des approches *décentralisées*. Pour éviter les goulots d'étranglement en lesquels peuvent se transformer les entités centrales de gestion de clé des protocoles de la classe précédente. On y distingue également des sous-catégories :

- ✓ Orientée adhésion : où le rekeying ce fait à chaque fois qu'un membre rejoint ou quitte le groupe. Ce qui est un inconvénient si le groupe est assez dynamique (trop d'adhésion et/ou de départ).
- ✓ Orientée temps (Systématique) : le rekeying ce fait à chaque intervalle de temps. Ainsi les membres quittant le groupe auront toujours l'accès au contenu sécurisé jusqu'à la fin de l'intervalle temporelle. Et les nouveaux membres n'y auront pas immédiatement accès, mais devront attendre le début d'une session pour obtenir la nouvelle clé et donc avoir accès au contenu sécurisé.

La dernière sous-classe, dégagée par les auteurs, est celle de la distribution par *accord*. Suivant la topologie virtuelle des membres du groupe, cette classe est subdivisée en :

- ✓ Coopération orientée anneau : où les membres sont disposés dans un anneau virtuel.
- ✓ Coopération orienté hiérarchie : où c'est plutôt une structuration hiérarchique qui est adoptée.
- ✓ Coopération orienté diffusion : qui se base sur la diffusion de messages secrets et distribution des calculs pour aboutir à la clé de groupe.

1.2.3.3.2. Deuxième classe

La seconde grande classe dégagée par les auteurs dans [36] est celle des TEK communes par sous-groupe. Dans cette classe il est question d'éviter les problèmes liés au phénomène 1-effects-n, qui risque d'apparaitre dans les approches à TEK commune suite à une dynamique des membres du groupe (adhésion, départ ou expulsion). En effet, avec une TEK commune, l'adhésion (ou le départ) d'un membre implique une mobilisation des tous les membres du groupe pour l'établissement d'une nouvelle clé, et ce afin de garantir la confidentialité passée (respectivement future).

Dans cette classe les auteurs regroupent deux principales sous-classes :

- ✓ Orienté dynamique de membre : où le renouvellement de la clé se fait à chaque fois qu'un membre rejoint ou quitte le groupe (ou est expulsé du groupe);
- ✓ Orienté temps : où il est question de renouveler les clés de façon périodique;

Il est clair que la classe orientée dynamique ne saurait être adaptée à des groupes où la dynamique est assez forte, où l'adhésion et le départ des membres est assez fréquent, car cela prendrai beaucoup de temps à réaliser des renouvellements de clé. Quant à la seconde classe, orientée temps, deux cas peuvent se présenter : Le premier étant qu'un membre quitte le groupe avant l'échéance, dans ce cas, si le membre est un malveillant il pourra toujours écouter le trafic car la clé ne peut être changer avant ladite période de changement, ainsi la confidentialité future n'est nullement assurée. Le deuxième cas est qu'un nouveau membre adhère au groupe avant la période de changement de clé. Le nouveau ne pourra pas accéder à la communication tant qu'il n'aura pas la nouvelle clé et devra rester ainsi en instance jusqu'à ce que le changement de clé soit opéré.

S.Gharout reprend la classification de Y.Challal dans [37] avec des exemples illustratifs de chacune des classes introduites au préalable (voir Tableau 3 Taxonomie des protocoles des protocoles de gestions de clé de groupe.).

	Centralisée		Décentralisée	Accord de clé distribuée	
TEK Commune	Paires Point-à-Point	Hiérarchie de clé		Coopération en anneau	Coopération hiérarchique
	GKMP Poovendran et al. Dunigan and Cao	LKH OFT Canett et al. ELK CFKM	SMKD IGKMP Hydra MARKS KRONOS DEP	Ingemarson et al. GDH	DH-LKH D-LKH D-OFT D-CFKM
TEK Indépendante par sous-groupe			Iolus KHIP Cipher Sequences Yang et al. Proxy Encryption SIM-KM		

Tableau 3 Taxonomie des protocoles des protocoles de gestions de clé de groupe [37].

1.2.3.4. Quatrième classification

Dans [35] une autre classification est mise en avant. On y évoque alors des critères de classification suivant comment et quand une mise à jour de la clé peut se faire. Suivant le *comment* de la mise en place du protocole de distribution les auteurs les classent en : centralisé, décentralisé, et distribué, rejoignant ainsi la classification de [34].

Puis les auteurs de [35] introduisent une autre classification suivant le critère « quand procéder à un changement de clé ». Là, ils classifient les protocoles en trois autres classes à savoir :

- ✓ Des protocoles orientés membre ; où la mise à jour de la clé se fait selon la dynamique des membres (adhésion et/ou expulsion d'un membre) ceci assure la confidentialité passée et future.
- ✓ Des protocoles orientés temps ; et là la mise à jour se fait durant des intervalles de temps réguliers.

- ✓ Des protocoles orientés message ; où la mise à jour de la clé se fait à chaque fois qu'un message est censé être envoyé.

On notera donc que quelque soit la technique utilisée pour la génération et le partage de la clé, il est impératif de garantir la confidentialité passée et la confidentialité future dans le groupe. Dans le premier cas, il s'agit d'interdire à un nouvel élément d'accéder aux données ayant circulées avant son adhésion au groupe. Pour ce qui est de la confidentialité future, il s'agit de garantir que si un élément vient à quitter le groupe, ce dernier ne pourra plus avoir accès aux données qui risquent de transiter au sein du groupe.

Conclusion

L'une des problématiques majeures de la sécurité de l'information, et plus particulièrement de son aspect confidentialité, est la distribution de clé de chiffrement. Que ce soit dans le cas du chiffrement symétrique, où une seule clé est utilisée pour le chiffrement et le déchiffrement, ou bien lors de l'utilisation du chiffrement asymétrique, où deux clés sont utilisées, l'une pour le chiffrement l'autre pour le déchiffrement, dans les deux cas de figure, les participants à une communication doivent être munie d'une/des clé(s) sûre. La question est alors de comment réaliser une telle tâche. Cette problématique a largement suscité l'intérêt des chercheurs dans le domaine de la sécurité, donnant lieu à de multiples propositions.

Lorsqu'il s'agit de groupe de communication la problématique est notamment plus accrue. Effectivement, dans un contexte de groupe il ne s'agit plus de deux participants, mais plus qui doivent s'échanger une clé sûre, leur permettant de communiquer de manière sécurisée. Dans ce sens également plusieurs protocoles ont été proposés.

En général, les protocoles de distribution de clé sont répertoriés en trois grandes catégories qui dépendent de l'engagement des parties communicantes dans le processus d'élaboration de la clé de groupe. On distingue alors entre des protocoles centralisés, selon qu'une entité centrale orchestre le processus et se charge de la gestion de la clé de groupe ; des protocoles décentralisés où l'entité centrale n'est plus, afin d'éviter la problématique du goulot d'étranglement, et enfin, la dernière catégorie est celle par accord où tous les membres participent à l'élaboration de la clé.

Toutefois, et malgré la multitude des protocoles proposés, ces derniers ont toujours des inconvénients et nul ne propose une clé totalement sûre, pis encore, on aura toujours besoin de partager une information au préalable, sinon durant le processus, ce qui peut s'avérer un risque si un quelconque espion possède la puissance de calcul nécessaire pour effectuer une opération inverse.

Pour cette raison, et d'autres, une nouvelle solution est apparue. Elle ne se base plus sur l'aspect computationnel, mais sur les principes de la mécanique et de la physique quantique assurant ainsi une clé totalement sûre. Il s'agit de la distribution de clé quantique. Mais avant d'aborder cette solution et l'explorer plus en détails, nous repassons en revue, dans le chapitre suivant, les principes de base du monde quantique.

Chapitre 3 Le quantique

If anyone thinks he knows Quantum mechanics,
He doesn't know Quantum mechanics.
Richard Feynman

Anyone who is not shoked by quantum theory,
has not understand a single word
Neils Bohr

Introduction

La puissance de calcul, pour épouser les besoins croissants des utilisateurs, ne cesse d'augmenter de manière exponentielle comme le prévoyait déjà Gordon Moore dans son papier [38]. En fait, cette même puissance de calcul est en étroite relation avec l'architecture des machines. Des architectures où le support d'information est le « bit classique », qui correspond à un unique état binaire, représenté par la valeur 1 ou 0.

Le bit traditionnel, reconnu comme support d'information classique par excellence, à un instant donné, ne peut prendre qu'une seule et unique valeur, soit le 1, soit le 0. Des valeurs qui traduisent, respectivement, le passage ou non d'un courant électrique. Le bit a servit depuis plusieurs décennies à l'élaboration de machine répondant aux critères de la machine de Van Newman, et qui n'a eu de cesse à se développer, répondant à chaque fois aux attentes de ces utilisateurs et plus précisément au besoin croissant en puissance de calcul exigée par les applications. Une croissance exponentielle prévue par la loi de Moore comme nous venons de l'introduire. Mais suivant cette même loi de Moore, nous ne sommes pas loin du phénomène du Mur, où des effets quantique peuvent alors survenir provoquant des incohérences.

A cette limite, où les lois de la physique Newtonienne ne peuvent plus expliquer des phénomènes se produisant à une échelle miniature, interviennent des lois totalement différentes de celles de Newton, ce sont les lois de la physique quantique. Ces mêmes lois prévoient l'usage d'un autre support d'information, qui lui, pourrait prendre l'une des deux valeurs que peut prendre le bit classique et bien plus encore, les deux valeurs en même temps. Il s'agit du bit quantique, dit qubit.

Afin de mieux comprendre et d'aborder la suite de notre travail, dans le présent chapitre, il est question d'introduire les principes de bases du monde quantique.

1. L'information quantique

Avant d'aborder les principes et les postulats de la mécanique quantique régissant le monde de *l'infiniment petit*, nous essayerons tout d'abord de faire une analogie avec certaines notions du monde classique, à commencer par la notion même de l'information quantique.

En fait, « information quantique » est en soit un terme générique utilisé pour désigner la théorie de traitement et de la transmission de l'information utilisant les spécificités de la mécanique quantique [39].

La mécanique quantique est reconnue comme étant la science qui permet de décrire le monde de l'infiniment petit. A cette échelle de l'infiniment petit, il est question d'atome, d'électron, et notamment de boson dont les photons, particules fondamentales considérées comme LE support d'information dans le domaine quantique.

La mécanique quantique est régit par un nombre de principes, plus ou moins déroutants, et échappant quelquefois à la logique à laquelle nous sommes habitués dans notre monde *macroscopique*. Des principes comme la superposition, l'intrication, la téléportation, le non-clonage, et bien d'autres que nous détaillerons dans la suite de ce chapitre et qui permettront de mieux comprendre plus loin le fonctionnement des protocoles de distribution de clé quantique y compris la solution proposée.

Nous reviendrons également sur quelques notions de calcul quantique ainsi que la représentation des portes logiques quantiques permettant l'élaboration de circuits quantiques, toutefois, nous ne nous étalerons pas plus sur ce volet de l'information quantique qui peut englober également l'algorithmique quantique : algorithme de recherche quantique et ses applications sur les bases de données, ou alors la transformation de Fourrier quantique et ses applications dans l'estimation de phase, ou la factorisation… (De plus amples détails sur ces sujets peuvent être trouvés dans [39]).

2. Composition de la lumière : La Dualité Onde/Particule

Comme nous venons de l'introduire, en informatique quantique, le photon, particule de lumière, est largement utilisé comme étant un support de l'information quantique. Dans cette section nous reviendrons tout d'abord sur la représentation de la lumière, mais aussi sur une particularité très importante qui caractérise la particule de lumière, en l'occurrence la dualité onde-particule.

Il est intéressant de signaler qu'une représentation de la lumière a déjà vu le jour il y a de cela bien des siècles. Chez les égyptiens plus précisément, avec la représentation d'*Aton*[17].

[17] *Aton* dieu du soleil dans l'Egypte antique. Première forme de hénothéisme au règne d'*Akhenaton*.

Figure 24 Culte d'Aton : à gauche une photo du musé *Ägyptisches Museum* de Berlin à droite *Akhenaton et Néfertiti lors du culte d'Aton Musée Égyptien du Caire* [40],

Avec la découverte des photons par A. Einstein (1879-1955), une question cruciale se posait avec insistance : « la lumière est-elle une onde ou une formation de particules ? » Il est donc intéressant de revenir sur l'étonnante composition de la lumière.

En fait, I. Newton (1642-1772) fut le premier grand savant à décréter le caractère corpusculaire de la lumière. Pour Newton la lumière est constituée d'une infinité de petites particules qui se propagent instantanément en ligne droite.

En revanche, une expérience bien connue en physique, permet de démontrer la nature ondulatoire de la lumière. Il s'agit de l'expérience des fentes des Young[18] (1773-1892) (voir Figure 25 L'expérience de Young.).

Le dispositif de l'expérience étant le suivant :

Une lumière monochromatique émise par une source, heurte une plaque opaque contenant deux fentes fines (**F1** et **F2**) éclairant un écran d'observation.

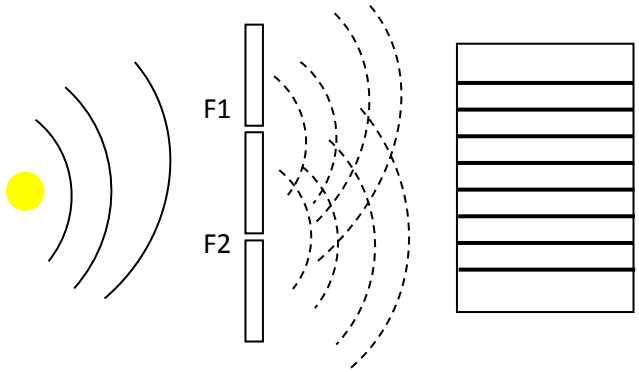


Figure 25 L'expérience de Young.

[18]L'expérience fut menée en 1801 par le pionnier de l'optique ondulatoire Thomas Young, on lui attribua son nom.

Si l'on obstrue la fente **F1**, on observera sur l'écran une répartition de l'intensité lumineuse **I2(x)** qui est la tache de diffraction de **F2** ; de même on remarque une répartition d'intensité **I1(x)** si la fente **F2** est obstruée.

Mais quand les deux fentes sont ouvertes, c'est un système de franges d'interférence qui est observé (interférence de franges obscures et claires).

Le plus étonnant est que l'intensité **I(x)** dans ce cas n'est pas la somme des deux intensités **I1(x)** et **I2(X)** produites respectivement par **F1** et **F2**.

$$I(x) \neq I_1(x)+I_2(x)$$

En fait, ni les prédictions de la théorie ondulatoire ni celles de la théorie corpusculaire ne sont vérifiées. Car même si on diminue l'intensité de la source de sorte à ce que les photons arrivent un à un sur la plaque opaque puis sur l'écran d'observation, les interférences de franges seront toujours observées sur ce dernier [41].

Plus déconcertant avec l'expérience des fentes de Yong, si un système d'observation est mis en place, afin de déterminer par quelle fente les photons passeront si l'une d'elles est obstrue, *le comportement* de la lumière redevient corpusculaire.

2.1. Dualité Onde/Particule

En fait, la lumière peut être décrite de deux manières différentes et antinomiques [42] :

- Soit on la considère comme un phénomène continu doté d'une fonction ondulatoire descriptible par sa longueur d'onde ($\lambda = cT$, c étant la vitesse de propagation de la lumière, T étant sa période exprimé en seconde $T = 1/\nu$ où ν est la fréquence d'oscillation ou de vibration exprimée en hertz);
- Soit qu'elle est considérée comme un phénomène discontinu constitué de particules dotée chacune d'une énergie ($E = h(f_i)$ où h est la constante de Planck et f_i la fréquence de rayonnement électromagnétique).

La possibilité de décrire le photon de ces deux manières, ondulatoire et corpusculaire, met en exergue le caractère contradictoire de la nature du photon, point qui a été soulevé par A. Einstein en 1909[19]. [42].

2.1.1. L'effet tunnel

Dans le domaine optique, lorsqu'une onde lumineuse arrive d'un milieu d'indice n_1 dans un milieu d'indice n_2, avec $n_2 > n_1$, il y a réflexion totale pour des angles d'incidence plus grands que l'angle i_0 donné par : $\sin i_0 = n_2/n_1$.

[19] A. Enstein avait soulevé la problématique de la nature contradictoire du photon lors d'une conférence intitulée « l'évolution de nos conceptions sur la nature et la constitution du rayonnement » présentée le 21 Septembre 1909 à Salzbourg [42].

Si le milieu d'indice n_2 est une lame suffisamment mince, la réflexion n'est pas totale, et il existe une petite probabilité de transmission car l'onde lumineuse ne s'annule pas complètement dans la lame : elle décroit exponentiellement et elle peut donc se raccorder à une onde progressive dans le milieu d'indice n_1. [39].

Et comme il a été souligné précédemment, la dualité onde-particule, ou onde-corpuscule, fait que le photon (entre autres support d'information quantique) peut être considéré comme une onde et non plus comme un corps ayant une masse et un volume. De ce fait, et en se comportant comme une onde, le qubit réussit *à traverser* un obstacle mis sur sa trajectoire. Il est possible également de faire une analogie avec les ondes sonores qui ne traversent pas en totalité un mur, mais une partie du son peut être perçue de l'autre côté du mur.

Ceci est pour dire qu'un qubit pouvant être représenté physiquement par un électron, peut traverser lui aussi un obstacle qui se trouve sur sa trajectoire et se trouver de l'autre côté du mur lorsqu'il emprunte sa propriété d'onde. C'est ce que l'on appelle l'effet tunnel, c'est comme si, de temps à autre, un tunnel s'ouvre dans l'obstacle permettant à la particule de passer.

3. Les principes et postulats de la mécanique quantique

Le support de l'information quantique étant différent du bit classique, il est important de connaitre à la fois les propriétés dudit support, mais aussi les bases qui lui confèrent ces propriétés.

3.1. Espace d'Hilbert

A tout système physique isolé est associé un espace Hilbertien (*i.e.* pas un espace de vecteurs complexes avec un produit scalaire). L'espace d'Hilbert est l'espace d'état. Un système est complètement décrit via son vecteur d'état qui est un vecteur unité dans l'espace des vecteurs d'états.

Un état est décrit par le *ket* $|\psi>$; dont la transposée est le *bra* $<\psi|$. Le produit scalaire entre vecteurs d'état $|\psi>$ et $|\phi>$ est noté $<\psi|\phi>$; et la condition de normalisation est que : $<\psi|\psi> = 1$

La notation de Dirac (1902-1984) est celle utilisée pour la description du support quantique de l'information tel que nous le précisons dans la section suivante.

L'espace d'état d'un système physique composé est le produit tensoriel des espaces d'état des systèmes physiques qui le composent.

3.2. Le Qubit

Par analogie à l'informatique dite classique, l'informatique quantique a aussi un support d'information, ce dernier étant dit le Qubit (noté également qbit). Mais à la différence du bit classique, qui ne prend que les valeurs 0 ou 1 suite à une mesure de ce dernier, le qbit lui, peut, suite à une mesure, révéler les valeurs 0 ou 1 (correspondants respectivement aux états |0> et |1>) ou, plus encore, être dans un état dit de superposition, qui est un état de combinaison linéaire des deux autres états.

Formellement, un qubit est représenté par la notation de Dirac comme suit :

$$|\psi>=\alpha|0>+\beta|1>$$ Équation 3 Notation de Dirac.

Les nombres α et β étant des nombres complexes tel que | α |^2+| β |^2=1, et où α est la probabilité d'avoir un 0 et β celle d'avoir un 1 suite à une mesure.

|0> et |1> sont des vecteurs tel que : |0>=$\begin{pmatrix}1\\0\end{pmatrix}$ et |1>=$\begin{pmatrix}0\\1\end{pmatrix}$

Physiquement parlant, plusieurs réalisations peuvent être adoptées en vue de la réalisation d'un qubit. Cela peut aller de deux polarisation différentes d'un photon, l'alignement d'un Spin dans un champ magnétique uniforme, ou deux états d'un électron qui orbite autour d'un atome unique [43] .

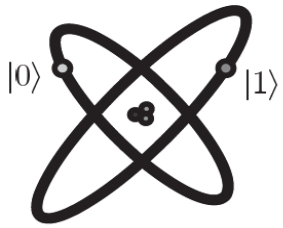


Figure 26 Représentation d'un qubit par des niveaux d'électron dans un atome. [43]

L'évolution d'un système quantique se décrit par l'opération unitaire ***U*** transformant l'état $|\psi>$ à l'instant t_1 à l'état $|\psi'>$ à l'instant t_2. [5]

$$|\psi'>=U|\psi>$$ Équation 4 Evolution d'un système quantique dans le temps.

3.2.1.Qubit multiple :

Toujours par analogie au cas classique, où une paire de bits serait dans l'un des états 00, 01, 10 ou 11, si l'on dispose de deux qubits alors il y aurait quatre états que l'on notera |00>, |01>, |10>, et |11> mais il y aurait également un état de superposition des quatre états précédents [43]:

$$|\psi>=\alpha_{00}|00>+\alpha_{01}|01>+\alpha_{10}|10>\alpha_{11}|11>$$ Équation 5 Etat de superposition d'une paire de qubits.

Où les mesures $|00>,|01>,|10>$ et $|11>$ apparaitraient, respectivement, avec les probabilités $\alpha_{00},\alpha_{01},\alpha_{10},$ et α_{11} et où $\sum_{x\in\{0,1\}^2}|\alpha_x|^2$ =1 tel que la notation {0,1}^{2} signifie une chaine binaire de langueur égale à 2.

Dans un état de superposition de deux qubits il est possible de mesurer un sous ensemble de qubits. Reprenons l'exemple introduit dans [43].

Si dans l'état de l'équation ci-dessus nous mesurons le premier qubit seul, ceci donnera la valeur 0 avec la probabilité suivante : $|\alpha_{00}|^2+|\alpha_{01}|^2$ laissant l'état de superposition en état de *post-mesure* suivant :

$$|\psi'>=\frac{\alpha_{00}|00>+\alpha_{01}|01>}{\sqrt{|\alpha_{00}|^2+|\alpha_{01}|^2}}$$ Équation 6 Etat de qubit en post mesure.

3.3. L'intrication quantique

Relevant de la science fiction pour certains, le principe de l'intrication quantique stipule que quand deux qubits sont intriqués s'ils sont séparés de quelques micromètres ou des kilomètres, donc arbitrairement éloignés, ces deux qubits, et donc ces deux systèmes, continuent à former un tout, une entité indissociable [39].

En fait, tout état qui ne peut être écrit sous forme de produit tensoriel est dit état intriqué [39].

On rappellera que si l'on a $|\varphi>$ et $|\chi>$ deux vecteurs tel que :

$|\varphi> = \alpha_1|0> + \beta_1|1>$ avec $|\alpha_1| + |\beta_1| = 1$ et

$|\chi> = \alpha_2|0> + \beta_2|1>$ avec $|\alpha_2| + |\beta_2| = 1$

Alors le produit tensoriel $|\varphi \otimes \chi>$ est comme suit :

$$|\varphi \otimes \chi> = \alpha_1\alpha_2|00> + \alpha_1\beta_2|01> + \beta_1\alpha_2|10> + \beta_1\beta_2|11>$$

Équation 7 Produit tensoriel de deux vecteurs.

Un vecteur arbitraire de la forme :

$$|\psi> = \alpha_{00}|00> + \alpha_{01}|01> + \alpha_{10}|10> + \alpha_{11}|11>$$

Pour qu'il puisse être écrit sous la forme donnée dans Équation 7 il est nécessaire que : $\alpha_{00}\alpha_{11} = \alpha_{01}\alpha_{10}$, condition qui *a priori* n'a aucun raison d'être valide [39]. $|\psi>$ n'étant pas un produit tensoriel, c'est un état intriqué.

3.4. Etat de Bell et corrélation

L'état de Bell, également appelé paire EPR (**E**instein-**P**odolsky-**R**osen) est un cas spécial des états intriqués :

$$|\boldsymbol{\phi}^+> = \frac{|\mathbf{00}> + |\mathbf{11}>}{\sqrt{\mathbf{2}}}$$

$$|\boldsymbol{\phi}^-> = \frac{|\mathbf{00}> - |\mathbf{11}>}{\sqrt{\mathbf{2}}}$$

$$|\boldsymbol{\psi}^+> = \frac{|\mathbf{01}> + |\mathbf{10}>}{\sqrt{\mathbf{2}}}$$

$$|\boldsymbol{\psi}^-> = \frac{|\mathbf{01}> - |\mathbf{10}>}{\sqrt{\mathbf{2}}}$$

Équation 8 Etats de Bell

Il est possible de démontrer que les états de Bell sont bien des états intriqués.

Sil l'on prend l'état de Bell $|\boldsymbol{\psi}^->$ (voir Équation 8) à titre d'exemple; pour que $|\boldsymbol{\psi}^->$ soit un produit tensoriel, il faudrait que l'on puisse l'écrire sous la forme donnée dans Équation 7 et que $(\alpha_1\alpha_2)(\beta_1\beta_2) = (\alpha_1\beta_2)(\beta_1\alpha_2)$

$$|\boldsymbol{\psi}^- > = \frac{|\mathbf{01} > -|\mathbf{10} >}{\sqrt{\mathbf{2}}}$$

$$|\boldsymbol{\psi}^- > = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{01} > -\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10} >$$

Donc :

$\alpha_1\alpha_2 = 0 \Rightarrow \alpha_1 = 0\ OU\ \alpha_2 = 0$

$$\alpha_1\beta_2 = {}^1\!/_{\sqrt{2}}\ ; \begin{cases} si\ \alpha_1 = 0 \Rightarrow \alpha_1\beta_2 \neq {}^1\!/_{\sqrt{2}} \therefore contradiction \\ si\ \beta_2 = 0 \Rightarrow \alpha_1\beta_2 \neq {}^1\!/_{\sqrt{2}} \therefore contradiction \end{cases}$$

$$\beta_1\alpha_2 = -{}^1\!/_{\sqrt{2}}\ ; \begin{cases} si\ \beta_1 = 0 \Rightarrow \beta_1\alpha_2 \neq -{}^1\!/_{\sqrt{2}} \therefore contradiction \\ si\ \alpha_2 = 0 \Rightarrow \beta_1\alpha_2 \neq -{}^1\!/_{\sqrt{2}} \therefore contradiction \end{cases}$$

De même si l'on prend :

$\beta_1\beta_2 = 0 \Rightarrow \beta_1 = 0\ OU\ \beta_2 = 0$

Ainsi donc :

$$(\alpha_1\alpha_2)(\beta_1\beta_2) \neq (\alpha_1\beta_2)(\beta_1\alpha_2) \therefore\ |\boldsymbol{\psi}^- >\ est\ un\ état\ intriqué$$

L'état de Bell est particulier en le fait que si, par exemple, de la mesure du premier bit de l'état $|\boldsymbol{\phi}^+ >$ resulte la valeur 0 avec la probabilité 1/2 donnant à l'état la forme *post-mesure* $|\boldsymbol{\phi}^{+'} > = |00 >$, également la mesure de la valeur 1 serait possible avec la même probabilité 1/2 avec une post-mesure $|\boldsymbol{\phi}^{+'} > = |11 >$. Il en résulte que la mesure du second qubit donnera exactement le même résultat que la mesure du premier qubit.

Il s'agit du concept de corrélation qui est à la base de la notion de téléportation quantique.

3.5. La téléportation quantique

Il s'agit d'un transfert d'information quantique d'un endroit à un autre sans que ce transfert soit associé à une propagation physique d'une particule qui porte l'information [39]. Pour réaliser une téléportation quantique d'une information d'une particule ***A***, il est nécessaire d'utiliser une paire auxiliaire de particules intriquées ***B***, ***C***. Une particule des deux paires sera gardée par Alice alors que l'autre sera au niveau de Bob. Puis une série d'opérations quantiques sera appliquée pour préparer le système.

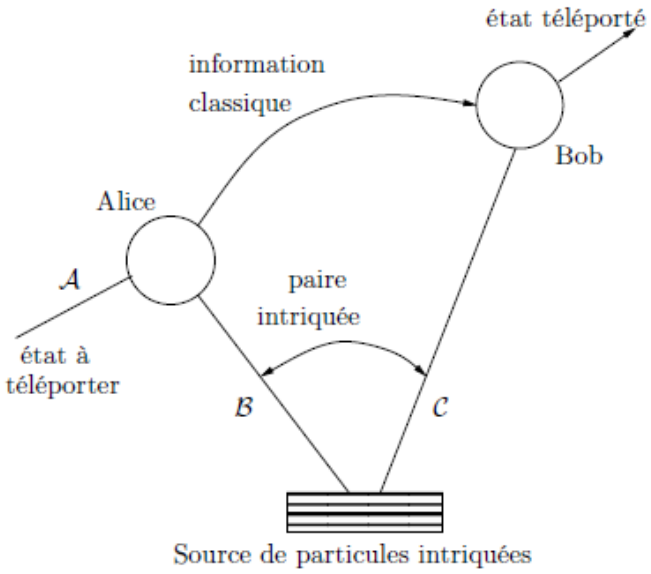


Figure 27 Téléportation quantique [39].

Nous reprenons un exemple donné dans [39] pour expliquer le processus de téléportation quantique.

Soit l'état de A donné par $|\psi> = \lambda\,|0> + \mu\,|0>$

Et l'état de la paire intriquée BC donné comme suit :

$$|\boldsymbol{\phi^{+}_{BC}}> = \frac{|\mathbf{0_B 0_C}> + |\mathbf{1_B 1_C}>}{\sqrt{\mathbf{2}}}$$

L'état des trois particules est alors :

$$|\phi^{+}_{ABC}> = (\lambda|0_A> + \mu|1_A>)\,\frac{1}{\sqrt{2}}\,(|0_B 0_C> + |1_B 1_C>)$$

$$= \frac{1}{\sqrt{2}}\,[\lambda\,|0_A>(|0_B 0_C> + |1_B 1_C>) + \mu|1_A>(|0_B 0_C> + |1_B 1_C>)]$$

$$= \frac{\lambda}{\sqrt{2}}\,[\,|0_A>(|0_B 0_C> + |1_B 1_C>)] + \frac{\mu}{\sqrt{2}}[\,|1_A>(|0_B 0_C> + |1_B 1_C>)]$$

Alice applique un CNOT[20] sur les qubits AB, le résultat est alors comme suit :

$$|\phi^{+}_{ABC}{}'> = \frac{\lambda}{\sqrt{2}}\,[\,|0_A>(|0_B 0_C> + |1_B 1_C>)] + \frac{\mu}{\sqrt{2}}[\,|1_A>(|1_B 0_C> + |0_B 1_C>)]$$

Puis elle appliquera l'opération *H* sur le qubit A ; cela donnera :

$$|\phi^{+}_{ABC}{}''> = \frac{\lambda}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}(|0_A> + |1_A>)(|0_B 0_C> + |1_B 1_C>)\right] + \frac{\mu}{\sqrt{2}}\left[(\frac{1}{\sqrt{2}}(|0_A> - |1_A>)(|1_B 0_C> + |0_B 1_C>)\right]$$

[20] L'opération logique CNOT ainsi que d'autres opérations logiques sont introduites un peu plus loin dans le chapitre.

$$= \frac{\lambda}{2} [|0_A 0_B 0_C > + |0_A 1_B 1_C > + |1_A 0_B 0_C > + |1_A 1_B 1_C >]$$

$$+ \frac{\mu}{2} [|0_A 1_B 0_C > + |0_A 0_B 1_C > - |1_A 1_B 0_C > - |1_A 0_B 1_C >]$$

$$= \frac{1}{2} \left[\lambda |0_A 0_B 0_C > + \lambda |0_A 1_B 1_C > + \lambda |1_A 0_B 0_C > + \lambda |1_A 1_B 1_C >\right]$$

$$+ \frac{1}{2} [\mu |0_A 1_B 0_C > + \mu |0_A 0_B 1_C > - \mu |1_A 1_B 0_C > - \mu |1_A 0_B 1_C >]$$

Ce qui peut s'écrire comme suit :

$$|\phi^{+}_{ABC}{}'' > = \frac{1}{2} |0_A 0_B > (\lambda |0_C > + \mu |1_C >)$$

$$+ \frac{1}{2} |0_A 1_B > (\lambda |0_C > + \mu |1_C >)$$

$$+ \frac{1}{2} |1_A 0_B > (\lambda |0_C > - \mu |1_C >)$$

$$+ \frac{1}{2} |1_A 1_B > (-\mu |0_C > + \lambda |1_C >)$$

La mesure conjointe par Alice des qubits A et B projette la paire ***AB*** sur l'un des quatre états $(|0_A 0_B >, |1_A 1_B >, |1_A 0_B >, |1_A 1_B >)$.

Si la mesure de la paire ***AB*** donne par exemple $|0_A 0_B >$ le qubit ***C*** de Bob sera dans l'état : $\lambda |0_C > + \mu |1_C >$. Alice n'aura qu'à informer Bob que le qubit lui arrivera dans le même état initiale du qubit ***A***. Si par contre elle mesure $|0_A 1_B >$ le qubit ***C*** sera dans l'état $\lambda |0_C > + \mu |1_C >$ et il faudra informer Bob qu'il doit, lui, appliquer une rotation de π autour de l'axe Ox ; ce qui revient à l'application de la matrice de Pauli $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$; s'il s'agit de $|1_A 0_B >$, alors Alice informera Bob qu'il faut appliquer une rotation π autour de l'axe Oy ; ce qui revient à l'application de la matrice de Pauli $\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ et enfin une rotation π autour de l'axe Oz ; ce qui revient à l'application de la matrice de Pauli $\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ si le résultat de la mesure de la paire ***AB*** est $|1_A 1_B >$.

On insistera sur le fait que dans le cadre de la téléportation quantique il n'y pas de transfert de matière et que l'état de ***C***, n'est connu que lorsque Alice divulgue, par voie classique, la mesure à appliquer. Cette information se fera à une vitesse *au plus* égale à celle de la lumière et donc il n'y a pas de transmission instantanée de l'information à distance [39].

3.6. La superposition quantique

Il est souvent difficile de concevoir l'état de superposition quantique de par l'absence d'une analogie dans le monde réel à un niveau microscopique, ceci dit, cela n'en diminue pas plus l'importance de ce phénomène.

Dans [43], les auteurs en donne une simplification. Le qubit étant vu comme une pièce de monnaie, peut être en position pile ou face. Mais pour une pièce de monnaie *imparfaite* qui balance il y a un état intermédiaire. Un qubit est alors en un *continuum* d'états entre |0> et |1>.

3.6.1. Chat de Schrödinger

E. Schrödinger (1887-1961) explique le concept de superposition quantique par une expérience de pensée (émise en 1935) qui lui doit même son nom [42]. Dans la littérature du domaine, l'expérience s'annonce comme suit :

Soit un chat enfermé dans une boite avec un dispositif qui libère un poison dans le cas où il détecte la désintégration d'un atome radioactif. Si l'on suppose que l'atome à une chance sur deux de se désactiver alors, le chat aurait une chance sur deux d'être vivant. Ayant la boite fermée, il serait impossible de connaitre exactement l'état du chat sans l'ouverture de la boite.

Il est donc possible, en dehors de toute mesure, et suivant les concepts de la mécanique quantique, de considérer le chat comme mort et vivant au même instant t.

3.7. Le principe d'incertitude de Heisenberg

Le principe d'incertitude de Heisenberg est une autre révolution du monde quantique qui révoque le principe de déterminisme de la mécanique classique, prouvant encore une fois, que le monde microscopique, ou de l'infiniment petit, à ses propres lois.

Elaboré en 1927 par Heisenberg (1901-1979), le principe est également appelé principe d'indéterminisme. Le principe d'incertitude est le fruit d'une longue réflexion de Heisenberg sur la non-commutativité de la multiplication de deux matrices comportant des couples de caractéristiques physique d'une particule, et en déduit que la non-commutativité est liée à l'ordre dans lequel l'observateur réalise les mesures [42].

Ainsi donc, il sera impossible d'avoir simultanément, et exactement, les valeurs de deux observables. Si l'on mesure alors la vitesse d'une particule à un instant donné, il est impossible de connaitre *exactement* sa position à ce même instant. Réciproquement, s'il s'agit de mesurer sa position en premier, il ne sera possible d'en mesure exactement la vitesse.

Dans [42] l'auteur revient sur ce principe en le simplifiant avec un exemple assez concret que nous reprenons ici. Utiliser une lampe électrique pour éclairer son chemin la nuit permet justement de voir les objets autours. Le faisceau lumineux de cette même lampe est composé de milliards de photons qui, en se réfléchissant sur chaque objet, permettent de le voir. Mais il ne nous viendrait pas à l'esprit que l'incidence des photons sur les objets éclairés les ferait bouger du fait que la masse de ces derniers est bien plus grande que celle des photons. Par contre, ceci est bien vrai dans l'infiniment petit, où la masse d'une particule donnée est tant petite que le choc avec un seul photon la ferait déplacer.

3.8. Le non-clonage

Il est tout à fait facile de réaliser une copie d'un bit classique. En plus du fait que la copie du bit aurait la même valeur que le bit originel, la copie, elle, qui ne se fait que suite à une mesure, n'entrainerai aucune perturbation ni changement de valeur du bit originel. Cependant, il est difficile de réaliser une telle opération aussi « simple » qu'elle puisse paraitre sur un qubit. En effet, mesurer un qubit entrainerai inévitablement une modification de l'état de ce dernier, car, étant dans une superposition d'états, la mesure de ce dernier lui imposerai la prise de valeur 0 ou 1, au lieu des *deux à la fois*. Une caractéristique bien propre au qubit que lui confère le principe fondamentale de non-clonage (non-clonning en Englais) ; un principe mise à jour en 1982 par Wootters William K et Zurek Wojciech H dans leur papier [44].

Il serait donc, selon la théorie du non-clonage, impossible de réaliser une copie d'un qubit, cela s'explique par le fait que pour le copier, il faudrait tout d'abord connaitre son état, et pour se faire il faut le mesurer, sauf qu'en le mesurant il perdra son état initial.

Plus formellement et en reprenant l'exemple explicatif donné dans [45] ; si l'on suppose *U* l'opération de copie d'un état quantique alors :

$$U|\psi> |.>= |\psi> |\psi>$$

Où $|\psi>$ représente l'état à copier et |.> représente l'état initiale du qubit qui servira de copie, alors nécessairement on devrait avoir :

$$U\ |0>|.>= |0>|0> \text{ et } U|1>|.>=|1>|1>.$$

Cependant : $U\ |\psi>|.>= U\ (\alpha\ |0>+ \beta\ |1>)|.>$

$$= U\ (\alpha\ |0> + U\ |1>)|.>$$

$$= \alpha\ U\ |0>|.> + \beta\ U\ |1>|.>$$

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

$$\neq |\psi> |\psi>$$

Car :

$$|\psi> |\psi>= \alpha^2\ |0> |0> + \alpha\beta\ (|0> |1> +|1> |0>) + \beta^2\ |1> |1>$$

Ce qui prouve formellement qu'il est impossible de réaliser la copie d'un état quantique.

3.9. Décohérence quantique

Si l'on considère que toute matière est composée d'atomes et de particules, et que ces derniers suivent les lois de la physique quantique, il semble « logique » d'en déduire que tout corps est régit par ces mêmes lois. Or il n'en est pas ainsi, les lois de la physique quantique ne peuvent être

applicables dans le monde macroscopique et engendrent des situations irréalistes, tel le cas du chat de Schrödinger.

En fait, la théorie de la décohérence quantique tente de trouver une réponse au problème de réduction du paquet d'ondes et de prouver que la réduction du paquet d'ondes est une conséquence de l'équation de Schrödinger et non pas en contradiction avec celle-ci. L'idée de la théorie est que tout système quantique ne doit pas être considérer comme isolé mais en interaction avec un environnement ayant un grand nombre de degrés de liberté et ce sont ces interactions qui provoquent la disparition rapide des états superposés ne laissant que les « observables » à une échelle macroscopique.

Selon G. Louis-Gavet [42], c'est bien le physicien français Serge Harouche[21] et son équipe à l'école supérieure de Paris, qui, en 2007, ont réussi à reproduire, et après quatorze ans de persévérance, le phénomène de décohérence en laboratoire. L'expérience démontre comment une mesure effectuée sur des particules déstabilise profondément celles-ci empêchant des phénomènes se produisant à cette échelle microscopique de se propager jusqu'à la notre.

La première étape de l'expérience, consistait à capturer un photon sans le détruire. Il fallait à cet effet un dispositif spécial constitué d'une boite métallique fermée et blindée aux parois épaisses appelée cavité, dotée d'un pouvoir réfléchissant et refroidie à -272,3 C^o grâce à un vernis de *niobium*[22]. Pour détecter un photon provenant des atomes du niobium, il fallait injecter dans la cavité, à des intervalles réguliers des atomes qui ne feront que *frôler* le photon. Cela ralentit pour un court instant l'électron situé sur l'orbite la plus externe et donc le décale d'un demi-tour, on lui attribue alors la valeur 1. Si par contre, l'atome sort intact de la cavité, on lui attribue la valeur 0. En suivant le même procédé, une deuxième étape a permis de capturer non pas un, mais plusieurs photons en superposition d'état, puis dans une troisième étape d'observer le phénomène de décohérence et ce dans un délai de 26 millisecondes [42].

4. Mesure d'un Qubit, Operations logiques quantiques et circuits quantiques :

Comme il a été introduit dans une section précédente, le support de l'informatique quantique est le qubit, et à l'image du bit classique, il est possible d'appliquer des opérations sur le bit quantique. Réaliser une mesure sur un bit classique revient à lui appliquer des opérations logiques via les portes logiques, de même il en est pour le qubit, à condition que les matrices utilisées en vue de telles opérations soient des matrices *unitaires*[23].

A titre d'exemple, l'opération logique NOT, transforme la valeur du 0 en 1, et inversement. Il est possible de trouver l'opération équivalente du NOT classique en mode quantique. Une opération qui transforme le |0> en |1> et inversement. Il s'agit de l'opération NOT quantique représentée par *X* et dénotant la matrice suivante :

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

[21] Prix Nobel de la physique 2012, Titulaire de la Chaire de physique quantique au Collège de France.
[22] Métal rare et supraconducteur à très basse température.
[23] Une matrice U est dite unitaire si $U^{\dagger}U = I$.

Il est possible de vérifier que l'application de l'opération *X* au vecteur d'état |0> donne |1> (respectivement l'application du *X* sur |1> donne |0>).

L'application du NOT quantique sur l'état quantique $|\psi> = \alpha|0> + \beta|1>$ qui s'écrire sous forme vectorielle $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$ serait alors le résultat d'un produit vectoriel:

$$X\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}\beta\\ \alpha\end{bmatrix}$$

Équation 9 Application du NOT quantique sur un état de superposition.

Ainsi le NOT quantique s'appliquera sur |0> et|1> simultanément. Cela va sans dire les avantages que cela procure en termes de vitesse de calcul.

Une autre opération aussi des plus importantes, l'opération d'Hadamard notée *H*. Elle se définit comme suit :

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

Équation 10 Opération d'Hadamard *H*.

L'opération d'Hadamard est une opération à un qubit, et à pour rôle de créer des superpositions d'états :

$$H\left|0>\right. = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\left|0>\right. = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix} = \frac{1}{\sqrt{2}}\left(\begin{bmatrix}1\\ 0\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\right) = \frac{1}{\sqrt{2}}(|0> + |1>)$$

$$H\left|1>\right. = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\left|1>\right. = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ -1\end{bmatrix} = \frac{1}{\sqrt{2}}\left(\begin{bmatrix}1\\ 0\end{bmatrix} + \begin{bmatrix}0\\ -1\end{bmatrix}\right) = \frac{1}{\sqrt{2}}(|0> - |1>)$$

Équation 11 Application de l'opération *H*.

La représentation en circuit de l'opération d'Hadamard est comme suit :

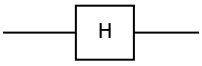


Figure 28 Circuit quantique de l'opération d'Hadamard.

Il est également possible d'appliquer des opérations sur non pas un seul qubit, mais sur un registre de *n* qubits. On distinguera alors entre deux types d'opération logiques quantiques : *locale* et *non-locale* [45]

4.1. Opération locale

Une opération est dite locale si elle prend la forme d'un produit tensoriel d'opérateurs agissant chacun sur un seul qubit.

4.2. Opération non-locale

On appellera opération non-locale une opération qui s'exprime comme produit tensoriel d'opération agissant individuellement sur chacun des qubits. La plus importantes de ces opérations est le $CNOT_{ij}$ (ou le NON contrôlé), qui, appliqué sur un registre de ***n*** qubits, permet d'inverser la valeur de l'un (le

j) en fonction de l'autre (le ***i*** qui a la valeur 1). On dira que le qubit ***i*** est la source que le qubit ***j*** est la destination.

Par exemple $CNOT_{12}$ agit sur les qubits 1 et 2 et permet d'inverser la valeur du qubit 2 si la valeur du qubit 1 est à 1. Si l'on prend un registre à deux qubits cela donnera les résultats suivants :

$$CNOT_{12}\,|00>=|00>$$

$$CNOT_{12}\,|01>=|01>$$

$$CNOT_{12}\,|10>=|11>$$

$$CNOT_{12}\,|11>=|10>$$

Équation 12 Application de l'opération CNOT sur un registre de deux qubits.

Quant à la représentation en circuit quantique de l'opération CNOT est comme suit :

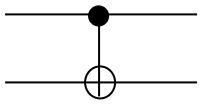


Figure 29 Circuit quantique de l'opération CNOT.

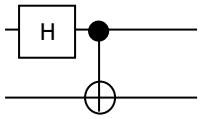


Figure 30 Circuit quantique complexe [45].

La combinaison des opérations logiques permet de créer des circuits quantiques complexes.

La Figure 30 est une représentation d'un circuit quantique complexe appliquant l'opération de Hadamard puis du CNOT sur un registre de 2 qubits.

On reprendra l'exemple donné par [45], appliquant les opérations H puis CNOT sur |00>.

$$CNOT\ H|00> = \mathrm{CNOT}\ \frac{1}{\sqrt{2}}\ (|0> +|1>)|0>$$

$$= \mathrm{CNOT}\ \frac{1}{\sqrt{2}}\ (|00> +|10>)$$

$$= \frac{1}{\sqrt{2}}\ (|00> +|11>)$$

4.3. Mesure partielle

Il est important de souligner que dans le domaine quantique, il est possible de réaliser des mesures partielles. Autrement dit, ne mesurer qu'une partie d'un système quantique, qui, peut être un état de superposition.

La règle de mesure partielle stipule que [46]:

Si l'on considère un système *AB* composé de deux parties A et B, et que l'on suppose que *AB* est dans l'état $|s>$. On applique sur la partie *A* une mesure ($|m_1>, \dots |m_N>$) où N est la dimension de *A*. Alors $|s>$ peut être écrit comme suit :

$$|s>=|m_1>\otimes|v_1>+\cdots+|m_N>\otimes|v_N>$$

Où $|v_1>, \dots |v_N>$ sont des vecteurs non normalisés de la partie *B.*

Alors :

La probabilité d'obtenir $|m_i>$ est $p_i =< v_i|v_i>$;

Si la ième mesure est obtenue, alors l'état final de la partie *A* est $|m_i>$ et l'état final de la partie B est $|v_i>/\sqrt{< v_i|v_i>}$.

Nous reprenons ici l'exemple introduit dans [46].

Soit une paire de photons dans l'état $|s>$ tel que :

$$|s>=\frac{1}{\sqrt{2}}(|00>+|11>)$$

On applique la mesure *M* sur le premier photon tel que

$$M=(|m_1>,|m_2>)=(\frac{1}{\sqrt{2}}(|0>+|1>),\frac{1}{\sqrt{2}}(|0>-|1>))$$

Quelle est la probabilité d'obtenir $|m_1>$ et quel serait alors l'état final du deuxième photon ?

Pour répondre à la question, l'état $|s>$ du système doit être réécrit sous la forme :

$$|s>=|m_1>\otimes|v_1>+|m_2>\otimes|v_2>$$

Alors :

$$|s>=\frac{1}{\sqrt{2}}(|0>+|1>)\otimes|v_1>+\frac{1}{\sqrt{2}}(|0>-|1>)\otimes|v_N>$$

Et comme tout $|v_i>$ peut être écrit sous la forme $|v_1>= a_i|0>+b_i|1>$ alors :

$$|s>=\frac{1}{\sqrt{2}}(|0>+|1>)\otimes(a_1|0>+b_1|1>)+\frac{1}{\sqrt{2}}(|0>-|1>)\otimes(a_2|0>+b_2|1>)$$

$$=\frac{1}{\sqrt{2}}[(|0>+|1>)\otimes(a_1|0>+b_1|1>)+(0>-|1>)\otimes(a_2|0>+b_2|1>)]$$

$$=\frac{1}{\sqrt{2}}[(a_1+a_2)|00>+(b_1+b_2)|01>+(a_1-a_2)|10>+(b_1-b_2)|11>]$$

On en déduit que :

$$(a_1+a_2)=1,(b_1+b_2)=0,(a_1-a_2)=0,(b_1-b_2)=1$$

Donc :

$$a_1 = a_2 = b_1 = {}^1\!/_2 \; et \; b_2 = -{}^1\!/_2$$

Ainsi :

$$|v_1> = \frac{1}{2}(|0> + |1>) et \; |v_1> = \frac{1}{2}(|0> - |1>)$$

La probabilité d'obtenir $|m_1>$ et $< v_1|v_1> = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = \frac{1}{2}$

Et l'état final du second photon est : ${}^{|v_1>}\!/_{\sqrt{< v_1|v_1}} = \left({}^1\!/_{\sqrt{2}}\right)(|0> + |1>)$

5. Applications des résultats du domaine quantique

En se basant sur les principes de la mécanique quantique, plusieurs protocoles ont vu le jour, et des applications mettant en œuvre ces mêmes principes ont été mises au point. L'une des plus importantes applications de la mécanique et la physique quantique, est la cryptographie quantique. En fait, c'est par abus de langage que l'on parle de cryptographie quantique pour désigner un domaine non loin mais plus précis, il s'agit de la distribution de clé quantique (***Q**uantum **K**ey **D**istribution* ou ***QKD***) que nous détaillerons dans le chapitre suivant.

La distribution de clé quantique est un champ fructueux exploitant les lois de la mécanique quantique que nous avons présenté dans ce chapitre et bien d'autres encore, donnant lieux à des implémentations pratiques. La distribution de clé quantique offre une solution non négligeable au problème de distribution de clé tant étudié dans le domaine classique. La distribution de clé quantique offre ainsi la possibilité aux protagonistes traditionnels, Alice et Bob de partager une clé secrète qui peut être utilisée à des fins cryptographiques. Toute tentative de l'espion Eve afin de récupérer la clé, ou une partie de la clé échangée ne peut passée inaperçue et ce grâce au principe de non-clonage et autres lois précédemment citées.

Nonobstant, la distribution de clé quantique n'est pas la seule application des résultats de la physique et mécanique quantique, d'autres applications ont vu le jour d'autres sont à prévoir dans le proches et moyen avenir.

Les réseaux quantiques sont une extension de l'idée de distribution de clé quantique, plusieurs institutions s'y sont intéressées, et ont mis en œuvre l'idée des réseaux quantique, nous citons à titre d'exemple le réseau quantique de DARPA et de SECOQC sur lesquels nous reviendrons plus loin. Il semble également que les réseaux quantiques seraient bien adéquats pour des domaines tels que le militaire, ou le gouvernemental et financier surtout avec la conjoncture mondiale actuelle, où la sécurité de l'information et des communications est primordiale. À noter qu'en 2004, une transaction monétaire fut réalisée entre Vienna City Hall et Bank Austria Creditanstalt [47].

Dans [47], l'auteur revient également sur des applications portables et des infrastructures quantiques. On y revient sur la proposition d'une idée novatrice d'un jeton contenant un secret quantique et utilisé comme moyen de liaison à un réseau quantique par exemple, ou à un réseau classique, cela offre à la fois les avantages d'une sécurité inconditionnelle, mais aussi la possibilité de détection d'intrus, de part les principes de la mécanique quantique.

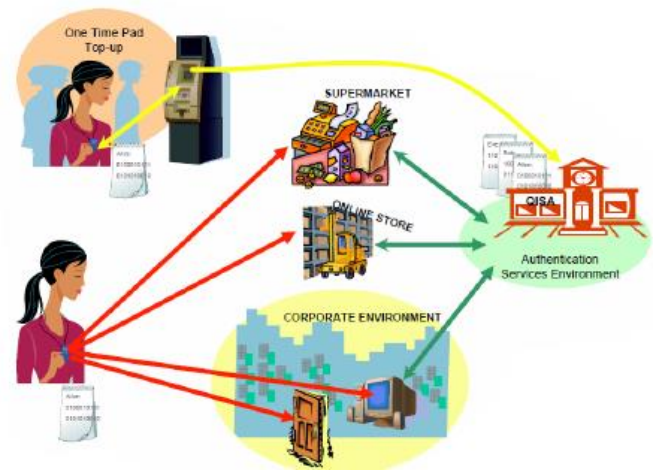


Figure 31 Authentification et application de la QKD [47].

Conclusion

La mécanique classique et la physique dite classique ou newtonienne, et ce malgré les résultats extraordinaires auxquels sont arrivées celles-ci et qui ont permis de comprendre le monde macroscopique qui nous entoure, mais aussi de pouvoir interagir avec, cette physique s'avère inapproprié pour l'étude des phénomènes ou même l'observation d'événements qui se produise dans le monde de l'infiniment petit.

En effet, comme le suggérait G. Louis-Gavet dans [42] il ne nous viendrait pas à l'esprit que les particules de lumière provenant d'une lampe peuvent déplacer les objets éclairés. Pourtant cela est vrai puisque si l'on admet que ces mêmes objets sont eux aussi constitués de particules, minuscules, le contact entre les particules de lumière et ces dernières est bien réel et provoque bien des phénomènes qui se traduisent, à notre échelle, que par la possibilité de voir ces objets là.

La physique et la mécanique quantiques sont là pour justement permettre de comprendre ce qui se passe à une si petite échelle que l'observateur humain ne peut atteindre.

Dans le présent chapitre, nous sommes revenus sur quelques principes et théories dites quantiques, qui décrivent justement le comportement des atomes et des particules, en particulier, les particules de lumière à savoir les photons. Pourquoi le photon, car de par la curiosité qu'a suscité les caractéristiques de cette particules, depuis bien des siècles déjà, elle est utilisée de nos jours comme le support de l'information quantique par excellence.

Nous avons donc présentés les principes de superposition, de non-clonage, d'intrication, de décohérence et bien d'autres permettant une meilleure compréhension des protocoles que nous présenterons dans le chapitre suivant. Il s'agira évidemment, de protocoles concrétisant ces principes là et donnant solution à une des problématiques de la sécurité informatique ; à savoir la distribution de clé.

Chapitre 4: La distribution de clé quantique

The human mind treats a new idea the way
The body treats a strange protein- it rejects it
Peter MEDAWAR

Introduction

La distribution de clé quantique est l'une des applications les plus importantes de la physique et la mécanique quantique. Appelée, par abus cryptographie quantique, il s'agit plutôt de protocole mettant en œuvre les principes de la mécanique quantique que nous avons introduit dans un chapitre précédent, afin de permettre à deux protagonistes d'établir une clé secrète commune. Une clé qui peut être utilisée à des fins de chiffrement afin de sécuriser une communication.

Dans le cadre de cette étude, nous nous sommes intéressés à la famille de protocole P&M, l'idée principale des protocoles de la distribution de clé quantique appartenant à cette famille est de coder l'information sur des états quantiques et d'échanger ces états sur un canal quantique. Alice alors prépare les états, les envoie à Bob qui, à son tour, les mesure en choisissant aléatoirement à chaque fois une base de polarisation, espérant ainsi à chaque fois de tomber sur la bonne base utilisée par Alice. Alice est Bob utilisent par la suite un canal classique qui servira à quelques échanges, entre autres, l'échange des bases utilisées. Si les deux bases, celles d'envoi et de réception utilisées respectivement par Alice et Bob sont identiques, alors Alice et Bob devraient avoir les mêmes valeurs d'information, sinon, si les valeurs sont différentes alors cela se traduit par le fait qu'un espion a dû tenter d'écouter les échanges. Il sera trahit par les principes de la mécanique quantique introduits précédemment. Dans le cas contraire, si les bases sont différentes les valeurs correspondantes seront automatiquement supprimées.

Dans le présent chapitre nous revenons sur les applications réelles de la distribution de clé quantique, 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 preuves de sécurité de ce dernier. L'accent est mis sur le BB84 non seulement parce qu'il s'agit du protocole de distribution de clé quantique le plus connu, mais aussi parce que c'est le protocole de distribution de clé quantique sur lequel se base une bonne partie de la solution proposée.

1. Distribution de clé, Partage de secret et réseau quantiques

L'engouement qu'a suscité la QKD a fait que depuis des années déjà on œuvre pour lui trouver des applications et pour bénéficier ainsi du niveau de sécurité qu'elle offre. Nous allons revoir, dans ce qui suit, quelques une de ces propositions allant de la théorie à l'application, depuis le BB84, protocole de distribution de clé quantique, jusqu'au réseau quantique en passant par le partage de secret quantique.

Proposé par Gilles Brassard et Charles Bennett en 1984 [4], le BB84, qui reprend par son acronyme les initiales de ses concepteurs, est le protocole de distribution de clé le plus connu. Il se fonde sur le principe du OTP (***O**ne **T**ime **P**ad*) tout en exploitant les principes de la mécanique quantique permettant ainsi de générer une chaine totalement aléatoire, qui peut être utilisée comme clé de chiffrement via un simple XOr.

Le développement dans le domaine de la distribution de clé quantique a connu un essor considérable et plusieurs implémentations ont vu le jour depuis la toute première implémentation de distribution de clé quantique qui consistait en un transfert de 40 cm [48]. Puis une distance accourue à 80 Km en 2013 dans une expérience mener par des chercheurs du CNRS de l'Institut d'optique Graduate School, de Télécom ParisTech, de l'INRIA et de la start-up SeQureNet [49]. Plus tard un record de 200 Km était enregistré et ce en 2014 par l'équipe de Jian-Wei Pan et Qiang Zhang de l'université Science et Technologie de Chine à Hefei [50] puis un autre record fut enregistré plus récemment par l'équipe d'Hugo Zbinden du Groupe de physique appliquée de l'UNIGE, qui est parvenue en 2015 à transmettre une clé quantique sur une distance de 307 kilomètres [51].

D'un autre côté, et d'un point de vue économique, des entreprises comme IdQuantique, MagiQ et SmartQuantum proposent sur le marché des produits matériels permettant de mettre au point des systèmes de distribution de clé quantique. Quant aux approches et prototypes exploitant la QKD dans le cadre de la sécurité des communications on notera qu'en 2003 par exemple, Gilles Brassard *et al* proposent une architecture pour implémenter un réseau de fibre optique pour la distribution de clé quantique en utilisant le multiplexage par répartition d'ondes (**WDM** pour ***W**avelength **D**ivision **M**ultiplexing*) permettant ainsi le partage d'une clé secrète et aléatoire avec chaque utilisateur du réseau tout en se basant sur le fameux protocole BB84 [52] Nous reviendrons sur le protocole BB84 plus loin dans notre étude.

Un peu plus loin de l'aspect théorique, et en parallèle à celui-ci, toujours en 2003, a vu le jour le projet européen SECOQC (***Se**cure **co**mmunication based on **Q**uantum **C**ryptography*). SECOQC est le fruit de collaboration de 41 groupes de recherche de pays de l'union européenne [53] [7]. Le but du projet était de mettre en place et tester la faisabilité d'un réseau de distribution de clé quantique, ce dernier fut déployer dans la ville de Vienne [53]. Durant la même année, DARPA et BBN Technologies (***B**olt, **B**ernek and **N**ewman Technologies*) construisent à leur tour un réseau de distribution de clé quantique fonctionnel. Le projet DARPA-BBN rassemblait au départ 6 nœuds, mais en regroupa plus tard 10 éparpillés sur les sites de l'université de Harvard, l'université de Boston et BBN [7]. La différence relevée entre ces deux projets avant-gardistes est que dans le projet DARPA la communication est contrôlée via des Switchs de fibre optique, basculant selon le partenaire de communication alors que dans le projet SECOQC, une paire de lien QKD forme un lien (QKD-link) fixe. Cependant, alors que dans le projet DARPA il s'agissait de faire une distribution de clé quantique en mode point-à-point, l'utilisation des répéteurs dans le réseau SECOQC permettait de dépasser cette

contrainte et aller vers une distribution multipoints -à-multipoints et où le contrôle de la communication classique était soumis à un agent central, responsable de la gestion de la clé générée et de sa sécurité d'un nœud à un autre [53].

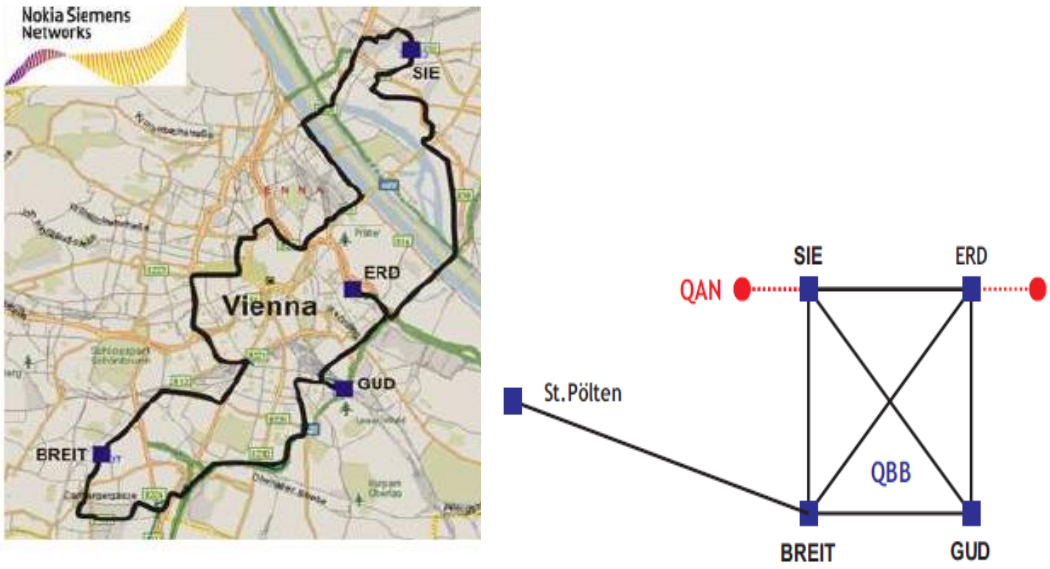


Figure 32 Réseau quantique SECOQC: à gauche:Carte de la ville de Vienne avec les station du réseau ; à droite Schéma représentatif du réseau [53].

Plus tôt, en 1999, Cleve *et al.*, dans [54] ont étudié le concept de partage de secret quantique QSS (***Q**uantum **S**ecret **S**haring*), ultérieurement, en 2000 D. Gottesman, lui, dans son papier [55]présente des résultats de recherche à propos du partage de secret quantique, et démontre, entre autres, que la taille de chaque partie doit avoir au moins la taille du secret partagé. Plus tard, en 2008, Lian-Fang *et al*. proposent un protocole de partage de secret multipartite pour une communication directe sécurisée en utilisant un seul photon [56]. Alors que la majorité des protocoles QSS sont du genre un-à-plusieurs (one-to-multiparty), dans [57] les auteurs proposent un protocole de partage de secret quantique multipartite-à-multipartite MMQSS (***M**ultiparty-to-**M**ultiparty **Q**uantum **S**ecret **S**haring*) ; les auteurs rappellent que Yan *et* Gao*,* ont proposé ultérieurement un protocole du genre, où deux groupes, respectivement de taille ***m*** et ***n*** s'échangent un secret. Les membres du groupe 1, génèrent conjointement un secret en codant chacun sa chaine secrète via des opérations unitaires, et le dernier membre du groupe 1, envoie 1/***n*** de la chaine de qubits résultante à chaque membre du groupe 2. Ce protocole fut démontré non sécurisé par Li *et al.* et Han *et al.* [57]. Les auteurs de la même source, Song *et al.* proposent alors un protocole MMQSS sécurisé en utilisant l'état GHZ (***G**reenberger-**H**orne-**Z**eilinger*) où deux groupes peuvent partager un bit du message secret en transmettant 2max (m,n) qubits. D'un autre côté, Wang *et al.* dans [58] proposent un protocole quantique de partage de message classique QSSCM (***Q**uantum **S**ecret **S**haring on **C**lassical **M**essage*) tout en utilisant un seul photon.

Il faut signaler ici que, si dans le domaine classique, la clé peut être en elle-même considérée comme étant le secret à partager [31] [32], le cas est différent quant au domaine quantique dans le sens où un protocole QSS est dédié à la sécurisation des opérations distribuées d'un calcul quantique, et qu'un QSDC est consacré à la transmission de secret directement sans établissement de clé préalable, quant aux protocole QSS-SDC (***Q**uantum **S**ecret **S**haring for **S**ecure **D**irect **C**ommunication*) ils combinent les rôles des QSS et QSDC [56].

En 2009, S.-K.Chong *et al.* proposent un protocole quantique par accord pour l'élaboration d'une clé, mais seulement entre deux entités [59]. Le protocole est basé sur des opérations unitaires, permettant aux deux participants, Alice et Bob, de faire un échange quantique suivi d'un autre sur le canal classique suite auxquels ils négocient la clé finale partagée. On signalera au passage que le protocole nécessite également l'utilisation des mémoires quantiques où seront stockés les qubits avant d'entamer la série de mesures. Ceci dit, la sécurité du protocole se base sur deux facteurs, à savoir l'utilisation de l'opération du XOr et du BB84 ; deux techniques prouvées sécurisées.

Plus récemment, en 2014, et concernant la distribution de clé proprement parlé, dans [6] les auteurs proposent une architecture multicast pour une gestion de clé centralisée utilisant la QKD et le chiffrement symétrique. Les auteurs introduisent l'idée de l'utilisation du QM_{KDC} centralisé (***Q**uantum **M**ulticast **K**ey **D**istribution **C**enter*). L'idée est que le QM_{KDC} génère pour chaque groupe multicast deux clés ; la première est une clé de groupe utilisée pour le chiffrement du trafique entre QM_{KDC} et le groupe multicast, alors que la seconde est une clé symétrique partagée entre les membres du groupe, elle sert au chiffrement du trafique entre les membres du groupe. Quand deux membres de groupes différents veulent communiquer, le QM_{KDC} est impliqué soit de façon totale (génération de clé, et envoi du message du destinataire vers la cible) ou partielle (seulement génération de la clé). Le processus est indiqué dans la Figure 33 Processus de gestion de clé dans une architecture centralisée utilisant la QKD.

En 2015, Metwaly *et al.* proposent dans [7] une architecture décentralisée de gestion de clé toujours en utilisant la QKD mais aussi un VPN (***V**irtual **P**rivate **N**etwork*). Dans cette proposition le groupe multicast est divisé en sous-groupes et à chaque sous-groupe est assigné un QM_{KC} (***M**ulticast **Q**uantum **K**ey **C**ontroller*). Chaque QM_{KC} nécessite deux canaux ; un classique et l'autre quantique, le premier est utilisé pour la transmission de message et la distribution de clé privée aux membres du sous-groupe, alors que le canal quantique est, lui, utilisé pour la transmission du signal quantique au sous-groupe concerné. La technique du QKD est utilisée dans cette proposition pour la génération de clé privée qui se fait par rotation de photon polarisé [7].

A noter que les premiers à en parler sont Cleve *et al* dans leur papier [54] puis Gottesman qui présenta des résultats à ce propos dans son papier [55].

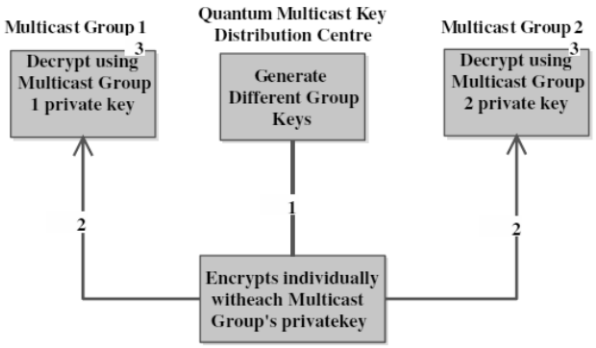


Figure 33 Processus de gestion de clé dans une architecture centralisée utilisant la QKD [6].

2. Protocoles de distribution de clé quantique

Le domaine où les principes de la mécanique quantique ont été exploités de manière très pratique est celui de la mise en place de protocoles de distribution de clé. Plusieurs protocoles ont été alors

proposés, nous reviendrons sur certains de ces protocoles dans la présente section, à commencer par le plus connu, en l'occurrence le BB84. Mais avant toute chose, il est important de signaler que l'idée de la distribution de clé quantique repose sur la transmission de quanta entre Alice et Bob sur un canal quantique. L'espionnage dans ce cas n'est autre que le fait de mesurer les quantas transmis sur le canal. Physiquement, mesurer revient inéluctablement à modifier un état, ce qui sera forcément détecté par Alice et Bob dans une phase d'échange ultérieure [48] [60].

Il faut signaler également que la distribution de clé quantique ne permet pas un échange d'information prédéfinie, mais assure la transmission d'une chaine aléatoire entre deux parties qui ne partagent initialement aucun secret mutuel.

2.1. BB84, l'idée

Comme nous l'avons précisé dans une section précédente, le BB84 est le protocole de distribution de clé le plus connu, probablement cela est dû à la simplicité de l'idée mais aussi à la sécurité qu'offre le protocole. Le protocole fut conçu par Charles Bennett et Gilles Brassard et présenté lors d'une conférence en Inde en 1984 (d'où son acronyme BB84) dans leur fameux papier « Quantum Cryptography: Public-key Distribution and Coin Tossing ».

Charles Bennett et Gille Brassard avaient exploité dans leur papier un principe très important de la physique quantique, à savoir le principe d'incertitude (voir page 84). Ils y rappellent que dans le contexte de la théorie de l'information classique, il est convenu que toute communication peut être passivement écoutée et même copiée, par tout tiers ignorant même la signification de l'information transmise. Cependant, si l'information est codée sous forme d'états quantiques non-orthogonaux, tel que les photons polarisés, le canal de transmission ne peut être écouté, ni que l'information soit copiée sans que cela n'entraine des perturbations menant à la détection de l'espion par les utilisateurs légitimes du canal de transmission [4].

A noter cependant que l'idée d'utiliser des états non-orthogonaux remonte aux travaux Stéphane Wiesner sur l'argent quantique infalsifiable qui exploite le principe de non-clonage (voir page 85) [48] [4].

Nous allons revenir dans ce qui suit sur les différentes étapes formant le BB84, avant d'enchainer sur les preuves de sécurité qui ont été avancées à propos de ce protocole.

2.2. Phases d'échange du BB84

Le déroulement du protocole de distribution de clé quantique BB84, comme décrit par ses concepteurs, passe par deux phases principales. La première consiste en un échange sur un canal quantique ; la seconde s'effectue sur un canal classique. Cette dernière étape permet aux participants à la communication de décider si oui ou non il y a eu espionnage et donc l'utilisation ou non du bit obtenu suite à l'échange quantique. Le résultat de ces échanges est une suite de bits aléatoires qui peut être utilisée ultérieurement à des fins cryptographiques comme le chiffrement via le One Time Pad.

2.2.1.Première phase : échange quantique

Durant cette première phase, les deux protagonistes, Alice et Bob, s'échangent des qubits sur le canal quantique. Concrètement les qubits peuvent être des photons polarisés. Alice choisit alors une chaine de bit aléatoire et une séquence de bases de polarisation (rectilinéaire que l'on note **+** ou diagonale que l'on note **x**). Alice prépare des photons polarisés suivant l'une des deux bases de polarisation utilisées dans le protocole puis les envoie à Bob. Chaque photon polarisé représente un bit de la séquence, suivant qu'une polarisation à 0^{o} ou à 45^{o} traduit un 0, et qu'un photon polarisé à 90^{o} ou 135^{o} traduit un 1.

A chaque envoi, Bob tente de mesurer correctement le qubit envoyé par Alice en choisissant aléatoirement l'une des deux bases de polarisation utilisées, traduisant ainsi à chaque fois les photons reçus en 0 et 1. Bob a une chance sur deux de mesurer correctement le qubit parce que toute tentative de mesurer un photon dans une polarisation qui n'est pas celle dans laquelle il a été préparé, lui fait perdre automatiquement l'information qu'il porte produisant une information tout à fait aléatoire.

2.2.2.Seconde phase : échange classique

La deuxième étape du protocole de distribution de clé quantique, tel introduit par ses concepteurs, se fait sur un canal classique susceptible d'être écouter (attaque passive), mais non sujet à une attaque active.

Alice et Bob déterminent alors quels sont les photons qui ont été reçus et ceux ayant été correctement mesurés. Un photon correctement mesuré signifie que Bob a utilisé la bonne base de polarisation pour le mesurer, évitant toute perturbation et récupérant ainsi la valeur transmise par Alice.
Nous illustrons le processus ci-dessus décrit dans la figure suivante.

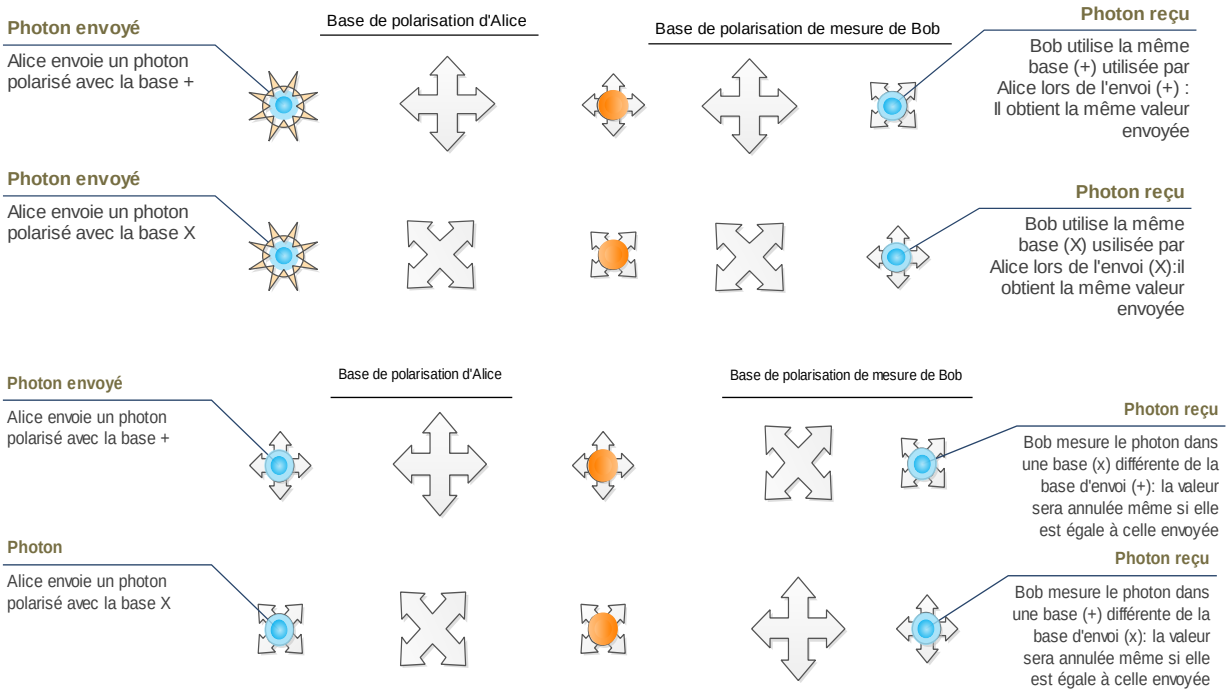


Figure 34 Echange quantique du BB84 sans éspion.

Voici un exemple explicatif de cet échange :

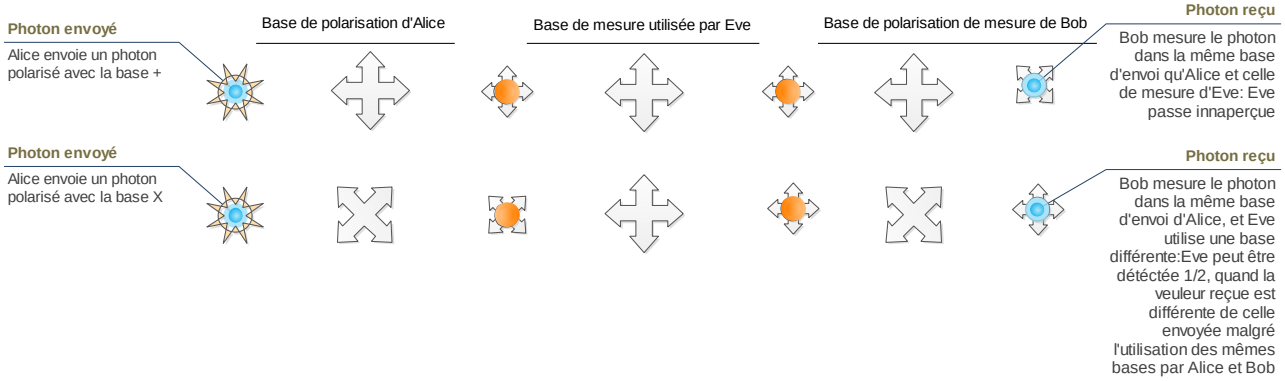
Bits Alice	1	0	0	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	0	1	1	0	1	0	1	0
Base Alice	+	x	x	x	+	x	+	+	x	+	+	x	x	+	x	+	+	+	x	x	+	+	+	x	+	x
Base Bob	x	+	x	+	x	x	x	+	x	+	x	+	+	+	+	x	x	+	+	+	x	+	+	x	+	+
Bits Bob	1	0	0	0	0	0	1	0	1	0	1	0	1	0	0	1	1	0	0	1	1	0	1	0	1	0
Décision	×	×	✓	×	×	✓	×	✓	✓	✓	×	×	×	✓	×	×	×	✓	×	×	×	✓	✓	✓	✓	×

Tableau 4 Exemple de distribution de clé quantique via BB84 sans la présence d'Eve.

Le tableau est un exemple d'échange de clé quantique entre Alice et Bob. Nous désignons par ✓ le fait que Bob utilise la bonne base de polarisation pour mesurer le photon envoyé par Alice et donc qu'il réussisse à récupérer la valeur binaire qu'il porte, cette dernière fera partie de la clé finale. Le symbole × désigne le fait que Bob utilise une base différente, et donc que la valeur du photon ainsi mesurer sera écartée, et ne figurera pas dans la clé finale. Si l'on nomme **K**, la chaine obtenue suite à cet échange, alors K=00010000101.

Ce que nous venons d'introduire est le cas idéal où il n'y a pas d'espion qui essaye d'intercepter la communication entre les participants légitimes Alice et Bob. Dans le cas contraire, et si un espion, généralement désigné par Eve (pour eavesdropper) essaye d'intercepter la communication entre Alice et Bob, cela altérera inévitablement l'état quantique et les participants légitimes décèleront son action. Ceci est dû au fait que Eve, ne connaissant pas les bases utilisées par Alice, fera exactement la même chose que Bob en tentant à chaque envoi de choisir la bonne base de polarisation pour mesurer le photon, ce qui arrivera une fois sur deux (1/2). Eve tentera par la suite de retransmettre à Bob les photons polarisés. Eve ayant mesuré un photon elle aura détruit son état quantique, et ne peut le reproduire suivant le principe de non-clonnage. Ainsi en envoyant des photons faussés à Bob, ce dernier tentera de les mesurer croyant qu'ils proviennent d'Alice. Il réussira bien sûr dans la moitié des cas. En présence d'Eve, Bob, donc, réussira à mesurer correctement les photons envoyés par Alice dans ¼ des cas.

La figure suivante illustre le cas d'un échange quantique en présence de l'espion Eve.



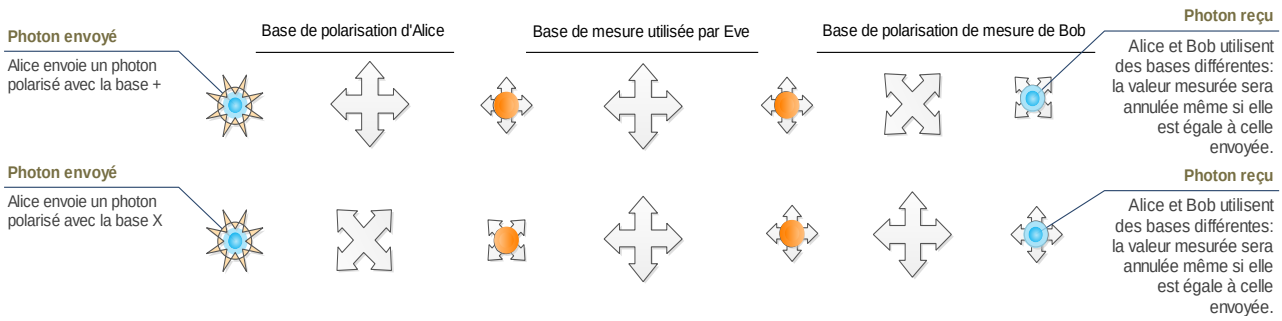


Figure 35 Echange quantique du BB84 en présence d'Eve.

Bits Alice	1	0	0	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	0	1	1	0	1	0	1	0
Base Alice	+	x	x	x	+	x	+	+	x	+	+	x	x	+	x	+	+	+	x	x	+	+	+	x	+	x
Base Eve	+	x	+	+	x	+	x	+	x	x	+	x	+	+	x	x	+	x	x	+	+	+	x	+	x	x
Valeur Eve	1	0	1	1	0	0	1	0	1	0	1	1	0	0	1	1	1	0	0	1	1	0	1	1	0	0
Base Bob	x	+	x	+	x	x	x	+	x	+	x	+	+	+	+	x	x	+	+	+	x	+	+	x	+	+
Valeur Bob	0	1	1	0	1	1	0	0	1	1	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	0
Décision	×	×	×	×	×	×	×	✓	✓	✓	×	×	×	✓	×	×	×	×	×	×	×	✓	×	×	✓	×

Tableau 5 Exemple de distribution de clé quantique via le BB84 en présence d'Eve.

Afin de vérifier s'il y a eu espionnage ou pas, Alice et Bob compareront, publiquement, les valeurs d'un sous ensemble de bits choisi aléatoirement parmi ceux mesurés correctement. Sacrifier ce sous ensemble (d'une taille égale au tiers de la taille de la chaine obtenue [4] [61] fera que la chaine est plus réduite, puisque ces bits seront écartés de la chaine finale.

2.3. Réconciliation

En pratique, un canal quantique ne peut être un canal totalement parfait. En effet certaines erreurs sont dues à son imperfection et peuvent être corrigées. Pour cela Alice et Bob doivent tout d'abord estimer la quantité d'information qu'un espion peut avoir. Cette quantité peut être acceptable, tolérable ou intolérable. Une quantité d'information tolérable veut dire qu'il serait possible via une certaine procédure d'obtenir une chaine secrète même en présence de bruit.

Dans les deux cas de figures, qu'il est ou non espionnage, la chaine résultante de cet échange quantique, devra passer par une série de traitement dans le but de corriger les éventuelles erreurs dues au canal de transmission, et de minimiser encore plus toute information qu'aurait pu avoir un espion ou qui aurait été divulguée suite à l'échange sur le canal classique.

En vue de l'amélioration du protocole de distribution de clé quantique, G. Bassard et L. Salvail avaient justement proposé un processus dit de réconciliation, permettant de rechercher et corriger les éventuelles erreurs apparaissant dans les chaines de bits qu'Alice et Bob auraient obtenu suite à leurs échanges. Ledit algorithme de correction d'erreur qui a montré son efficacité est l'algorithme CASCADE.

L'algorithme CASCADE est un algorithme interactif qui se déroule en plusieurs itérations, et se base sur un algorithme de recherche dichotomique, permettant d'effectuer une recherche dichotomique sur les deux chaines d'Alice et Bob afin d'identifier une position où les valeurs diffèrent et de corriger enfin cette erreur.

CASCADE tel que décrit initialement dans [62] et [63] permute à chaque itération ***i*** les chaines de bits d'Alice et Bob, puis les scinde en blocs de taille k_i. L'exécution de la procédure de recherche dichotomique s'appliquera alors à chaque bloc suite à la détection d'une éventuelle différence entre les blocs d'Alice et Bob, cette détection est possible grâce à une comparaison de parité des deux chaines. Une fois l'erreur détectée et corrigée, la position du bit est identifiée dans tous les blocs auxquels il a appartenu dans les itérations précédentes, puis corriger dans ces mêmes blocs. La correction de la valeur d'un bit erroné dans d'autres blocs auxquels il appartenait provoquera nécessairement l'apparition de nouvelles erreurs qu'il faudra également corriger.

Dans l'algorithme CASCADE seule la taille du bloc de la première itération est déterminée en fonction du taux d'erreur calculé précédemment. La taille des blocs pour les itérations suivantes doublera à chaque fois tel que [63]:

$$k_{i+1} = 2k_i$$ **Équation 13 Taille initiale de bloc pour l'algorithme CASCADE.**

Une analyse détaillée et des résultats empiriques ont montré cependant qu'un choix optimal de k_1 serait de :

$$k_1 = \frac{0{,}73}{\varepsilon}$$ **Équation 14Choix empirique de la taille de k_1**

Où ε est le taux d'erreur calculé.

Le nombre d'itérations quant à lui est déterminé comme étant le plus petit nombre tel que [63]:

$$2^p > n$$ **Équation 15 Nombre d'itération de l'algorithme CASCADE.**

Où n est la taille des chaines d'Alice et Bob.

Des performances de CASCADE sont montrées dans [61] [63] comme suit :

P	K_1	I'(4)	Limite de Shannon	I(4)
0,01	73	6,47	5,89	6,81
0,05	14	4,60	4,01	4,64
0,10	7	3,81	3,28	3,99
0,15	5	3,80	3,05	4,12

Tableau 6 Performance empiriques de CASCADE.

Il s'agit des résultats de l'application de CASCADE sur des échantillons de n=10000 bits, pour quatre taux d'erreurs. Le nombre de bits échangés publiquement lors du protocole afin de réconcilier les chaines d'Alice et Bob. La valeur I(4) est le nombre estimé de bits pour 4 itérations de CASCADE et I'(4) est celui obtenu en pratique sur 10 tests empiriques.

2.4. Amplification

Parce qu'un espion peut acquérir de l'information suite aux échanges publics qu'effectuent Alice et Bob lors de la phase de réconciliation, il est important de procéder à une autre étape permettant d'éliminer cette éventuelle information, il s'agit de l'étape de l'amplification (ou privacy amplification en anglais).

Il s'agit de choisir une fonction de hachage et d'y faire passer la chaine distillée (celle obtenue jusqu'à ce stade) afin d'obtenir une chaine plus réduite certes, mais plus sûre. Un choix adéquat de fonction de hachage est celui des fonctions de hachage de la classe des fonctions universelles (aussi dite universel$_2$ notée H3 de Wagman et Carter) [62] [63].

Claude Crépeau dans [63] introduit la définition de fonction de hachage universelle comme suit :

> Une classe F de fonction $A \to B$ est universel$_2$ (ou simplement universel) si, pour tout paire x_1, x_2 d'éléments distincts de A, la probabilité que $f(x_1) = f(x_2)$ est d'au plus $1/|B|$ quand f est choisit aléatoirement dans F uniformément.

Une fois cette phase réalisée, Alice et Bob peuvent être sûrs d'avoir la même chaine. Une chaine aléatoire et sûre, pouvant être utilisée à des fins cryptographiques.

Le schéma ci-dessous sur la Figure 36 récapitule les différentes étapes du BB84 dans sa version standard.

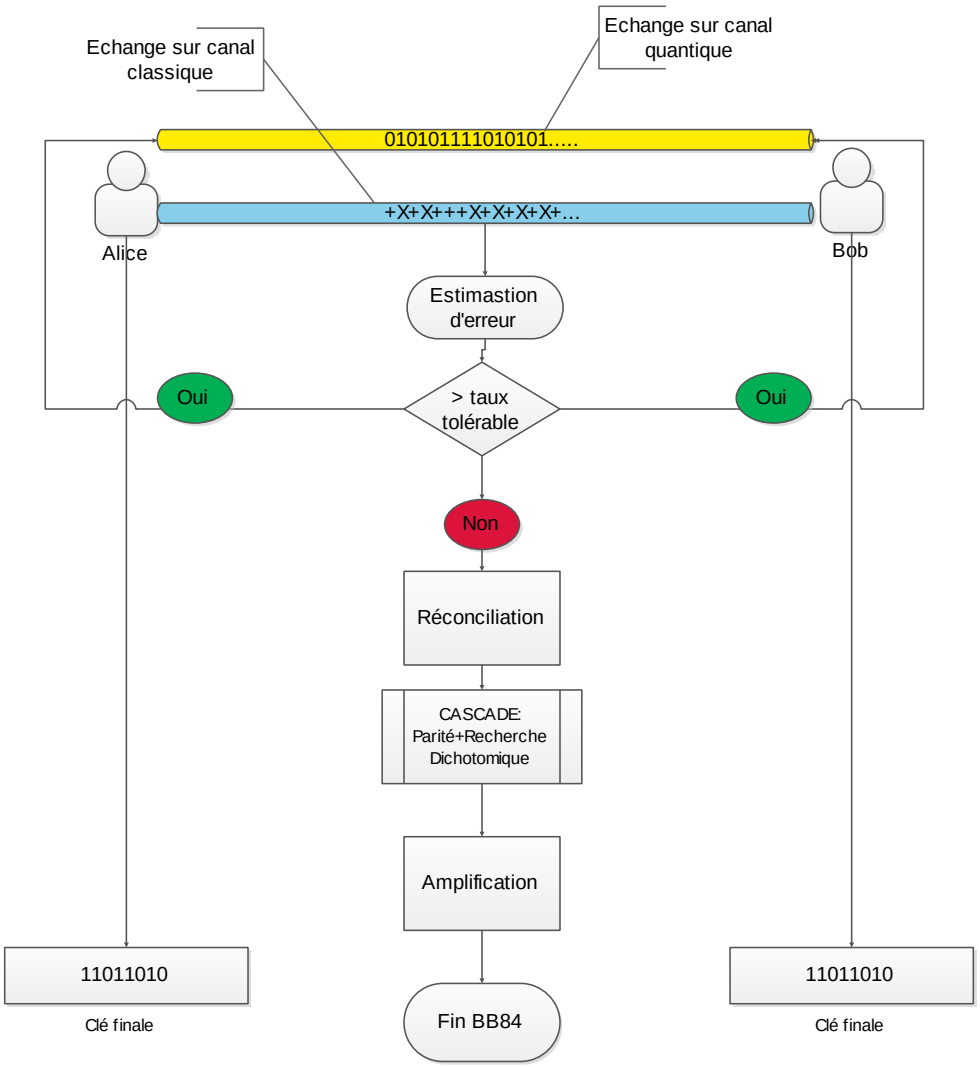


Figure 36 Schéma récapitulatif des étapes du BB84.

2.5. Preuves de sécurité du BB84

La distribution de clé quantique est l'une des applications les plus importantes de la physique quantique. L'exploitation des lois de la physique quantique a permis la mise en place de protocoles permettant d'assurer l'échange d'information sans que les participants à une communication n'aient d'information commune au départ. L'information échangée est alors une information aléatoire et sécurisée, qui peut être utilisée ultérieurement à des fins cryptographiques comme le chiffrement.

Le fait de se baser sur les lois de physique quantique fait que le processus de distribution de clé est vu comme un processus physique lié à l'envoi d'une information d'un point vers un autre. L'espionnage dans ce contexte n'est autre que des mesures effectuées sur le canal de transmission afin d'en tirer les valeurs envoyées sans qu'il y est de perturbations ou de modifications de ces valeurs. Cependant, l'intervention de toute entité sur le canal quantique sera inévitablement détectée comme nous l'avons précisé auparavant.

A part cette démonstration intuitive de la sécurité du protocole, des preuves de sécurité du protocole de distribution de clé quantique ont été avancées. Plusieurs travaux ont été menés dans ce sens, prouvant la sécurité des protocoles de distribution de clé quantique. Dans le présent travail,

nous nous intéressons exclusivement à la preuve de sécurité du protocole BB84 sujet de notre travail.

En 1999, dans [64] H-K. Lo et H.F Chau montrent qu'il est possible, via l'utilisation d'ordinateurs quantiques, de réaliser une distribution de clé quantique inconditionnellement sûre, sur une distance assez longue. Utilisant la preuve avancée par H-K. Lo et H.F Chau, Peter W. Shor et John Preskill dans (Simple Proof of Security of the BB84 Quantum Key Distribution Protocol, 2000) apportent la preuve de sécurité du BB84. Peter W. Shor et John Preskill se basent dans leur preuve sur les protocoles de purification de CSS[24] (***C**alderbank-**S**hor-**S**teane*) et les codes correcteurs quantiques, ce qui leur permet d'éviter l'usage des ordinateurs quantiques comme dans la preuve de H-K. Lo et H.F Chau.

Dans [65] E. Biham *et al* présentent une preuve de la sécurité théorique de la distribution de clé quantique contre un bon nombre d'attaques qui peuvent être effectuées sur le canal quantique par un espion qui a une puissance de calcul illimitée. E. Biham *et al* reviennent sur le fait que la preuve de H-K Lo et H.F Chau nécessite l'usage d'ordinateurs quantiques et de mémoires quantiques par les participants. E. Biham *et al* montrent aussi dans leur preuve à quel point l'aspect aléatoire dans le choix des bases de polarisation mais aussi le choix aléatoire des bits de test sont important pour atteindre une bonne sécurité d'un protocole de distribution de clé quantique. D. Mayers dans [66] apporte également une preuve de sécurité du protocole de distribution de clé quantique tout en prenant en considération l'imperfection des outils de mesure ainsi que celle du canal de transmission, mais en admettant que les pulsations sont à photon unique.

La sécurité inconditionnelle du protocole de distribution de clé quantique BB84 n'est pas à remettre en question, elle a été abordée dans plus d'un travail, nonobstant, la modélisation et l'analyse de ces protocoles a le mérite d'être étudiée [8] d'autant plus que l'avancée théorique n'est pas au même rythme que celle pratique et que souvent le besoin de vérifier l'exactitude des prototypes, surtout dans un domaine comme la sécurité de l'information, est très important avant la mise en place de toute nouvelle procédure. Pour ce faire, toute une discipline est apparue, celle de la vérification formelle des modèles. La première étape de la vérification est celle de la définition du modèle dans une notation mathématique bien déterminée, puis d'utiliser un outil d'analyse automatique pour soit vérifier que le modèle suit le comportement attendu, soit pour vérifier que le système satisfait bien des propriétés qui, elles, sont à exprimer à part dans un langage de spécification propre.

Dans ce contexte, celui de la vérification de modèle, et en 2002, Rajagopal Nagarajan et *al* abordent déjà l'idée de l'analyse des protocoles quantiques en appliquant les techniques de vérification formelles utilisées antérieurement pour l'analyse des systèmes de communication classique [8]. Ils prennent pour exemple le protocole de distribution de clé quantique le plus connu, en l'occurrence le BB84 et proposent un modèle formel et une analyse du protocole via le langage CCS (***C**alculus of **C**ommunication **S**ystem*)[25]. En 2004, Nikolaos K. Papanikolaou, sous la direction de Rajagopal Nagarajan présente des travaux sur l'utilisation des vérificateurs de modèle, entre autres le vérificateur PRISM mais aussi SPIN, pour la vérification du protocole BB84 dans ces travaux de recherche en Master [67]. Plus tard, et en collaboration avec Garry Bowen et Simon J. Gay, le travail

[24] Les codes correcteurs dépassent le cadre du présent travail, cependant des détails concernant les CSS peuvent être trouvés dans [77]

[25] Langage de description/modélisation des processus de communication introduit par Robin Milner vers 1980.

fait l'objet d'un papier intitulé « An automated analysis of the security of quantum key distribution » où les auteurs discutent l'utilisation d'un vérificateur de modèle probabiliste, à savoir PRISM, pour la vérification du protocole de distribution de clé, le BB84 [10]. Les résultats ainsi obtenus montrent que la tergiversation de l'espion par rapport au canal diminue de manière exponentielle avec le nombre de qubits transmis, mais aussi que la probabilité de détection de l'espion augmente exponentiellement avec le nombre de qubits. Ce qui confirme les résultats théoriques obtenus au préalable concernant la sécurité du BB84 [66].

Toujours dans le même sens, Simon J. Gay et *al*, auteurs de [68] semblent être précurseurs dans la mise en place d'outils de vérification dédiés aux protocoles de distribution de clé quantique. Simon J. Gay et *al* introduisent ainsi dans [68] des techniques automatiques pour l'analyse des protocoles quantiques, permettant de modéliser des classes de protocoles quantiques. Leur outil QMC (***Q****uantum* ***M****odel-****C****heking*) est utilisé pour la vérification des protocoles combinant des calculs classiques et quantiques. Le travail a fait l'objet d'une thèse de doctorat où Nikolaos K. Papanikolaou, toujours sous la direction de Rajagopal Nagarajan, revient en détail sur le vérificateur QMC ainsi que le langage de spécification des modèles quantiques [69].

Plus tard en 2010, Elboukhari et *al* reprennent les travaux de Nikolaos K. Papanikolaou, en utilisant le vérificateur de modèle PRISM pour modéliser et vérifier le protocole de distribution de clé quantique, le BB84, tout en accentuant leur travail sur la propriété de détection de l'espion [70] puis dans un autre travail intitulé « *Verification of quantum cryptography protocols by model checking »* [71] en introduisant le paramètre d'efficacité du canal quantique et la force de l'espion

3. D'autres protocoles

Il est certain que le protocole de distribution de clé quantique le plus connu est le BB84, ceci dit, d'autres protocoles ont vu le jour. Sans être exhaustifs, nous revenons ici, sommairement, sur les principes de deux d'entre eux, le B92 et le SARG04.

3.1. B92

Le B92 fut proposé par Bennett en 1992 [72]. L'idée principale du B92 est d'utiliser deux bases non-orthogonales au lieu de quatre comme c'était le cas du BB84. Alice prépare une séquence binaire de systèmes quantiques représentant 0 et 1. Comme Bob ne peut pas deviner à chaque fois la base utilisée, il effectuera par des mesures des tests lui permettant d'obtenir la bonne réponse, puis Bob déclare publiquement à Alice quelle mesure lui a fourni un résultat positif [72] [73].

Plus formellement, soit / (45^o) et | (90^o) les deux polarisations utilisées par Alice représentant respectivement les valeurs binaire 1 et 0. Celles de Bob sont — (0^o) et \ (135^o) traduisant respectivement les valeurs 1 et 0.

Alice prépare les photons dans l'une des bases de polarisation qu'elle a, et les envoie à Bob, qui comme dans le BB84, tente de les mesurer correctement. Il enregistre à chaque fois si oui ou non il a détecté le photon polarisé envoyé par Alice (ce qui arrivera une fois sur deux quand les bases sont non-orthogonales).

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	×	×	×	✓	×	✓	×	×

Tableau 7 Exemple explicatif du protocole B92.

Bien entendu, une partie du résultat sera sacrifiée durant la phase de réconciliation afin de vérifier l'absence de tout espion.

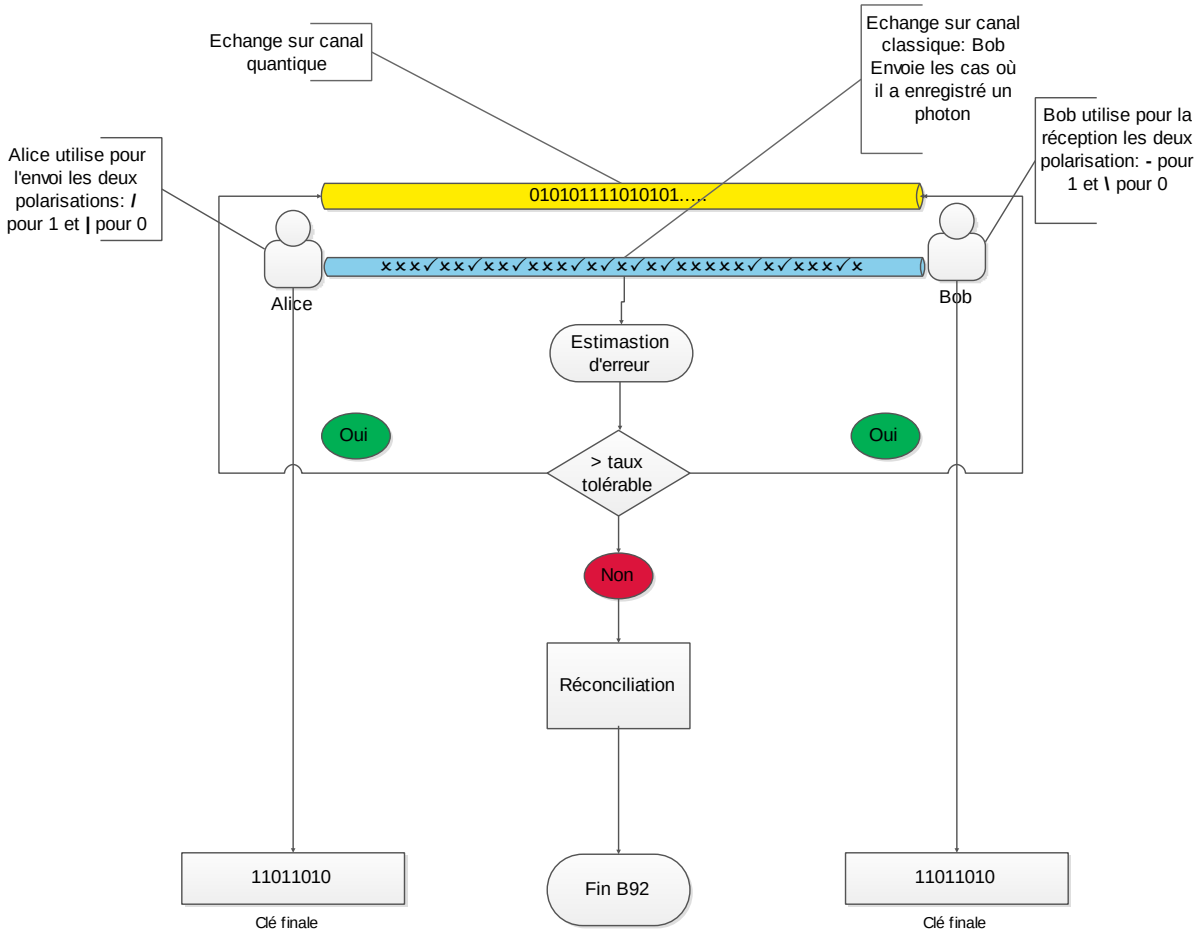


Figure 37 Schéma récapitulatif des étapes du B92.

3.2. SARG04

Un autre protocole de distribution de clé quantique que nous introduisant brièvement, il s'agit du protocole SARG04. Le protocole a été proposé par Scarani Valerio et *al* en 2004 [74]. Le protocole SARG, du nom de ses concepteurs **S**carani, Valerio; **A**cin, Antonio; **R**ibordy, Grégoire; et **G**isin, Nicolas, est une variante du BB84 et diffère du BB84 dans la phase d'annonce des bases.

Dans le SARG04, le codage de l'information classique (0 et 1) n'est pas le même que dans le BB84 comme nous l'avons vu précédemment, plus encore dans SARG04 ce ne sont plus les états, mais les pairs d'états qui représentent l'information. $|0>$ et $|1>$ de la base '+' par exemple représentent le bit 0, alors que $|0>$ et $|1>$ de la base 'X' représentent le bit 1. Alice révèle alors des paires d'états non-orthogonaux au lieu d'annoncer les bases mêmes utilisées. Lors de la réception, Bob de son côté

mesure la déviation de chaque polarisation, et à chaque résultat négatif l on déduit que Alice a certainement envoyé l'état opposé.

Nous reprenons l'exemple donné dans [75]pour plus d'explication.

Bit Alice	1	0	1	1	1	0	1	1	1	0	1	0	1	1	0	0	0	1	1	1
Base d'Alice	↓	↑	↑	↓	↓	↓	↑	↓	↑	↓	↑	↑	↑	↓	↑	↓	↓	↓	↑	↓
Polarisation Photon Alice	↖	↙	↘	↖	↖	↗	↘	↖	↘	↖	↘	↙	↘	↖	↙	↗	↗	↖	↘	↖
Base Bob	↓	↓	↑	↓	↑	↑	↓	↓	↑	↑	↑	↓	↑	↓	↓	↑	↓	↓	↑	↑
Mesure Bob	↖		↘	↖				↖	↘		↘		↘	↖			↗	↖	↘	
Annonce Alice	↓		←	→				↑	←		↑		→	←			↓	↑	→	
Décision	×	×	×	×	×	×	×	✓	×	×	×	×	✓	✓	×	×	×	✓	✓	×

Tableau 8 Exemple explicatif du SARG04.

Le SARG04 a été conçu contre l'attaque PNS (***P****hoton* ***N****umber* ***S****plitting*) [74] [75]. Une PNS est une attaque menée contre un dispositif où les pulsations contiennent plus d'un photon. L'espion Eve pourrait garder un photon et laisser les autres passer, puis, suite à une écoute passive, récupérer la base dans laquelle il a été envoyé (phase d'annonce de bases) et mesurer le photon ainsi correctement. Encore faut-il signaler que Eve doit disposer également de mémoire quantique lui permettant de *garder* le photon sans mesure jusqu'à récupération de la base.

Conclusion

La distribution de clé quantique est une application phare de la mécanique et la physique quantique où leurs principes sont mis à la disposition de la sécurité de l'information, précisément au service d'une confidentialité de haut niveau par la proposition d'une solution, indéniablement, sûre au problème de distribution de clé, qui est un problème classique bien connu de la communauté du domaine et pour lequel plusieurs solutions ont été proposées sans que celles-ci n'apportent de solutions définitivement sûres.

L'idée conductrice des protocoles de distribution de clé quantique et le codage de l'information sur un support totalement différent de celui classique déjà utilisé en informatique classique. L'usage du qubit, et donc des états quantique permet ainsi, grâce aux principes de la mécanique quantique, de détecter toute intrusion lors des échanges, car tout tentative entraine un changement d'état qui sera inévitablement décelé lors des échanges entre les participants légitimes.

Les premiers protocoles de distribution de clé quantique se faisaient au laboratoire permettant un échange quantique de quelques centimètres seulement. Actuellement, l'application de ces protocoles dépasse le cadre théorique et celui des laboratoires de recherche dans lequel elle évoluait durant des années, pour s'imposer comme une solution adoptée par plusieurs entreprises tel que IdQuantique, MagiQ…

Nonobstant, les protocoles de distribution de clé sont souvent conçus pour un échange entre deux parties, et ceux proposés pour un échange de groupe sont souvent basés sur l'enchevêtrement quantique (à l'image du E91 basé sur les paires EPR). L'une des raisons pour lesquelles nous nous sommes penchés sur ce volet et pour lequel nous proposons une solution de distribution de clé quantique dans un groupe. Une solution qui fera l'objet du prochain chapitre où elle sera détaillée.

Partie II : Réalisation

Chapitre 5 : Contributions

Introduction

La distribution de clé quantique est l'une des applications les plus connues de la physique et la mécanique quantique. Elle permet de résoudre l'un des problèmes majeurs de la sécurité et plus précisément de la confidentialité, à savoir la création et la distribution d'une clé de chiffrement sûre et sécurisée. Les principes de la mécanique quantique (incertitude, superposition,…) sont mis en œuvre afin de générer une clé sûre mais aussi aléatoire, pouvant être utilisée à des fins de chiffrement, et offrant ainsi une sécurité inconditionnelle.

Le protocole phare de distribution de clé quantique et qui a mis en avant ces principes est le fameux BB84 comme nous l'avons présenté dans le chapitre précédent. Le BB84, ainsi que d'autres protocoles qui ont suivi, et qui se basent sur le même principe, ont été conçu pour un échange se limitant seulement à deux participants. Rares sont alors les protocoles conçus pour un échange et une distribution de clé quantique dans le contexte de groupe de communication. Ce qui fut alors la motivation majeure de notre travail.

A cet effet, nous proposons alors une solution de distribution de clé quantique dans un groupe. La solution s'inscrit dans la catégorie des schémas par accord où tous les membres du groupe participent à l'élaboration de la clé par l'échange de grains de clé sécurisés via des clés intermédiaires quantiques. La clé finale est obtenue suite à des opérations de XOr des différents grains. Une sorte de OTP appliqué à plusieurs reprises et renforçant ainsi la sécurité apportée par l'aspect quantique de la solution.

Afin de valider la solution proposée, la sécurité du protocole a été considérée par le vérificateur de modèle PRISM. Il s'agit d'un vérificateur de modèle formel, permettant d'explorer exhaustivement toutes les possibilités d'états du modèle proposé tout en évitant les aléas de la simulation.

Les résultats de la vérification du modèle seront discutés à la fin du chapitre.

1. Description du protocole proposé

Il est important de signaler, comme cela a été précisé dans de nombreux travaux antérieurs au notre comme dans [66] et [76], que pour assurer une distribution de clé quantique entre Alice et Bob, il est évident que ces derniers doivent au préalable s'authentifier, sinon s'assurer que le canal public est sûr [66] autrement il serait non raisonnable de prétende à un échange sûr. Nonobstant l'importance de l'aspect authentification dans toute communication, ce volet ne sera pas couvert par le présent travail, mais nous nous intéressons exclusivement à la distribution de clé de chiffrement. Nous supposons donc dans la suite de notre travail que tous Alice et Bob sont authentifiés.

L'idée générale du protocole est de permettre à un groupe de participants de générer une clé de groupe secrète et sûre, et ce en impliquant chaque membre du groupe d'un côté et de l'autre en se basant sur le protocole de distribution de clé BB84 qui, initialement, est proposé pour assurer la distribution de clé entre seulement deux participants, alors que nous exploitons son efficacité déjà prouvée (Voir section Preuves de sécurité du BB84) dans un contexte de groupe, chose qui au moment de la rédaction de ce travail et suivant la recherche faite, n'existe pas. Les modèles de distribution de clé quantique multi-parties existants sont abordés dans la section (Distribution de clé, Partage de secret et réseau quantiques)

On rappellera que seulement dans les travaux de Metwaly et *al* [6] et [7] les auteurs proposent, respectivement, une architecture centralisée, et décentralisée pour la distribution de clé quantique dans un groupe alors que notre solution est plutôt une architecture distribuée.

Nous nommerons la solution proposée QDGKM pour (***Q**uantum **D**istributed **G**roup **K**ey **M**anagement*). Elle est composée de deux phases. La première phase du protocole consiste en la création de la clé de groupe que l'on note K. C'est la combinaison de tous les grains et apparaitra comme telle au niveau du dernier nœud de groupe. Pour cela chaque participant apporte « son grain de sel » en générant une partie de la clé. Une partie aléatoire qui sera combinée avec les parties précédentes via une opération de XOr, puis transmise au membre suivant en la chiffrant avec une clé intermédiaire.

Une clé intermédiaire est une clé quantique que l'on note $K_{i,i+1}$ générée entre le participant i et $i+1$ du groupe via le protocole quantique BB84.

La seconde phase consiste en la distribution de la clé K pour tous les autres membres via une série de chiffrement-déchiffrement successifs.

Pour assurer la confidentialité passée et future, une opération de rekeying est prévue après chaque adhésion ou départ (exclusion) de membre. Nous reviendrons sur ces cas dans les sections suivantes.

1.1. Notations

Nous utiliserons les notations suivantes dans la suite de la section afin de décrire les différentes étapes du protocole proposé.

Notation	Description de la notation
K	Clé du groupe
$K_{i;i+1}$	*Ki*,*i*+1 clé quantique intermédiaire partagée entre le participant *i* et son voisin *i* + 1
S_i	Grain du participant *i*
pred(new)	Predecesseur du nouveau participant
succ (new)	Successeur du nouveau participant
Id	Identifiant du participant

Tableau 9 Desciption des notations utilisées.

1.2. Hypothèses

Avant d'aller plus en détails dans la présentation de la solution proposée pour la résolution de la problématique de la distribution de clé dans un groupe, il est nécessaire de clarifier et mettre au point certaines hypothèses de départ.

Notons en premier lieu que dans le cadre de ce travail nous nous intéressons exclusivement à l'aspect confidentialité de la sécurité. Plus précisément à la problématique de la distribution de clé dans un contexte de groupe. Ainsi, le volet authentification ne sera pas abordé dans ce travail et nous supposons que deux voisins successifs sont authentifiés l'un par rapport l'autre, ce qui leur permet de dérouler le protocole BB84 en toute confiance.

Initialement, les membres sont organisés sous forme d'anneau, et ayant chacun un identifiant noté 'Id'. Les identifiants augmentent dans le sens des aiguilles d'une montre.

Chaque deux participants, voisins successifs, $\mathrm{i}, \mathrm{i}+1$, exécutent le protocole BB84 de sorte à ce qu'ils obtiennent une clé secrète. Il s'agit d'une clé intermédiaire que l'on note $\mathrm{K}_{\mathrm{i},\mathrm{i}+1}$

Chaque participant i génère un grain aléatoire, S_i, qu'il gardera secret. La combinaison de tous les S_i, par une opération XOr, formera la clé finale K.

Enfin, on supposera que tous les grains ainsi que toutes les clés quantiques intermédiaires ont la même taille.

1.3. Cas général

L'idée du protocole est simple, et le cas général se résume, comme nous l'avons précisé avant, en deux phases, la première phase qui consiste en la création de la clé, puis la seconde phase qui consiste en la distribution de cette même clé.

Au départ de la première phase, chaque deux voisins exécutent le BB84 afin d'obtenir une clé quantique secrète intermédiaire. Puis chaque nœud participant génère un grain, aléatoire, qu'il fera passer à son voisin via la clé secrète intermédiaire. Une fois le grain reçu par le voisin, ce dernier pourra le récupérer puis y combiner son propre grain par une opération de XOr.

Chaque deux voisins exécutent le même processus, et tous les grains seront à la fin combinés par un XOr. Le résultat obtenu au niveau du dernier participant constitue la clé de groupe qui devra être distribuée au restant des membres. C'est là que commence la deuxième phase.

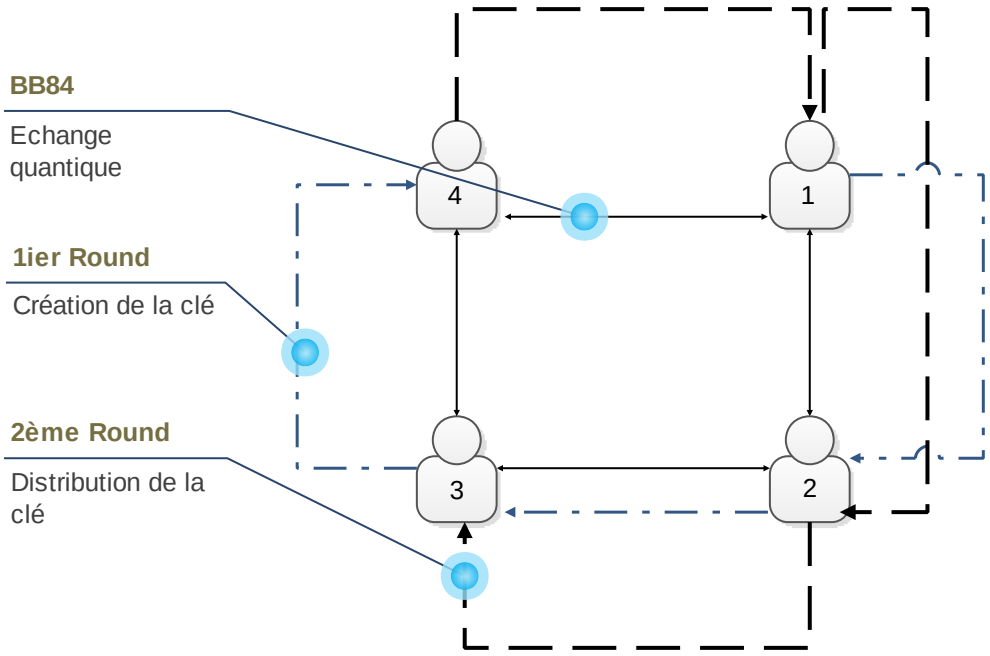


Figure 38 Cas général du protocole: exemple de 4 participants.

La figure ci-dessus résume le cas général du protocole proposé avec un groupe de quatre (04) participants.

Nous reprenons les étapes du protocole dans l'algorithme suivant :

Algorithme 1

i=1; *j*=1; $K=S_1 \oplus K_{1,2}$;
//1er Round
While (*j*<N) Do
i→succ(i): K; //Le participant '*i*' envoie K à son successeur
$K=K \oplus K_{(i,succ(i))}$;//Le participant '*i*' récupère la clé partielle K
$K=K \oplus Succ(i) \oplus K_{(succ(i),succ(succ(i)))}$; //Le participant '*i*' ajoute sont grain, et chiffre le résultat avec la clé initermédiaire qu'il partage avec son propre voisin.
i=succ(i);
j=*j*+1;
End;
//2ème Round
While (j<=2(N-1)) Do
i→succ(i): K; //Le participant '*i*' envoie K à son successeur
$K=K \oplus K_{(i,succ(i))}$;//Le participant '*i*' récupère la clé K
i= succ(i);
j=j+1;
End;
Stop.

1.4. Cas de l'ajout/adhésion d'un participant

Afin d'assurer la confidentialité passée, et si un nouveau participant rejoint le groupe, une nouvelle clé doit être générée.

Le nouveau participant exécute à cet effet deux BB84, d'un côté et de l'autre, générant ainsi des clés quantiques intermédiaires avec son prédécesseur et son successeur. C'est là que commence la première phase, où le prédécesseur génère un nouveau grain, le combine avec l'ancienne clé de groupe via une opération de XOr et envoie le tout chiffré via la clé quantique au nouveau participant. A son tour, ce dernier récupère le résultat du XOr et y rajoute, par la même opération, son propre grain puis renvoie le tout à son successeur qui exécutera le même procédé.

Le nouveau participant ne fera que récupérer le résultat du XOr de l'ancienne clé de groupe et le grain de son prédécesseur, et non pas l'ancienne clé de groupe en soi.

Une fois arrivée au niveau du successeur du nouveau membre, la nouvelle clé de groupe sera créée, et il ne restera qu'à la distribuer au reste du groupe dans une succession de chiffrement/déchiffrement via les clés quantiques intermédiaires déjà partagées entre les différents participants.

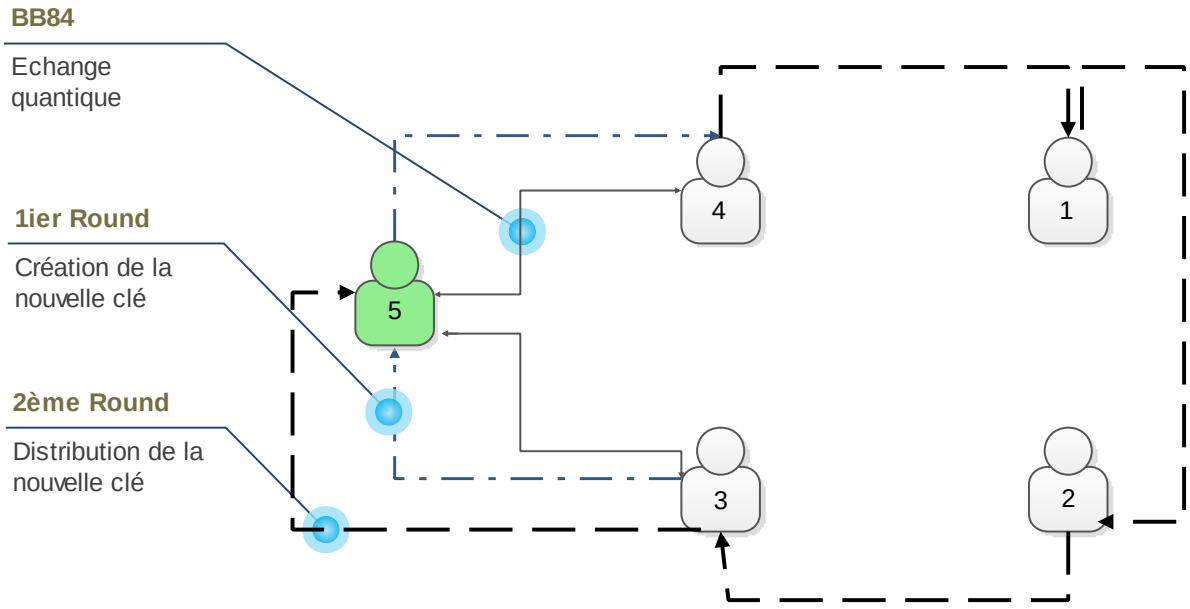


Figure 39 Cas de l'ajout d'un participant.

Algorithme 2

BB84(new, pred(new));
BB84(new, succ(new));
K=K⊕S(id(pred(new))⊕K(pred(new),new) ;//Predecesseur ajoute son grain
Pred(new) →new: K;//Predecesseur envoie K au nouveau participant
K=K⊕K(pred(new),new)//Le nouveau participant récupère la clé K
K=K⊕S(new)⊕K(new, succ(new)); // Le nouveau participant rajoute sont grain
New→Succ(new): K// Le nouveau participant envoie K à son successeur
K=K⊕K(new, succ(new))//Successeur récupère K
K=K⊕S(id(succ(new));// Successeur rajoute son grain
i=id (succ (new));
j=1;
While (j<= N) Do
$K=K\oplus K_{(i,\ succ(i))}$;
i→succ(i): K;//i envoie la clé K à son successeur

i= succ(i);
j=j+1;
End;
refresh (id);
Stop.

1.5. Cas de la suppression d'un participant

Afin d'assurer la confidentialité future, et dans le cas où un participant quitte, volontairement ou non, le groupe, une nouvelle clé doit être générée. Pour ce faire, et une fois le participant en question hors du groupe, son prédécesseur et son successeur (voisins suite au changement qui a touché le groupe) exécutent un BB84 leur permettant d'avoir une nouvelle clé quantique intermédiaire.

Le prédécesseur génère un nouveau grain, le combine avec l'ancienne clé de groupe, puis l'envoie, chiffré, à son nouveau successeur. Ce dernier récupère le résultat et lui rajoute son nouveau grain, créant ainsi la nouvelle clé de groupe qui sera distribuée au reste du groupe en la faisant passer d'un nœud à un autre tout en la chiffrant/déchiffrant via les clés quantiques intermédiaires.

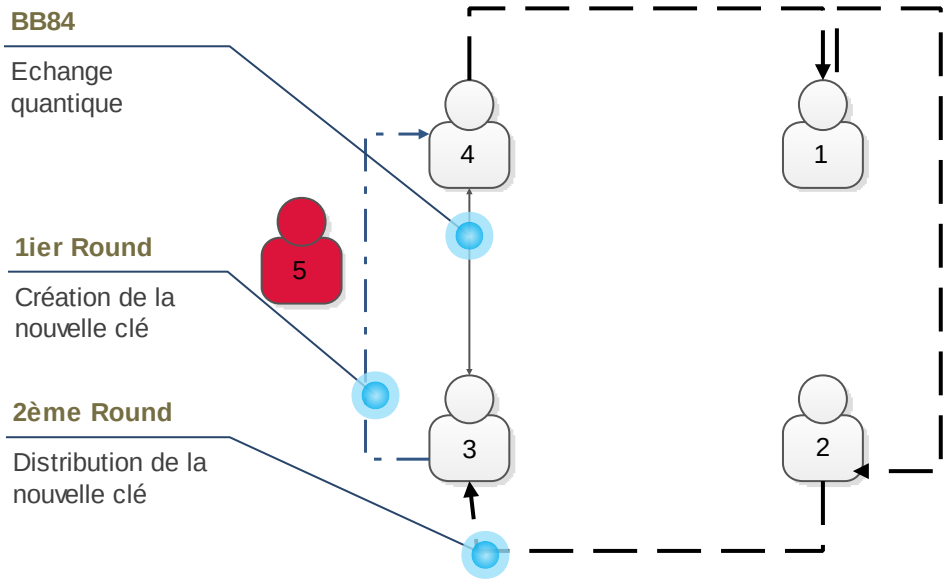


Figure 40 Cas de la suppression d'un participant.

Algorithm 03

i=id(left_node); K=0;
BB84(pred(i)), (succ(i));//BB84 est exécute entre le prédécesseur et le successeur du participant partant.
$K=S_{(pred(i))} \oplus K_{(pred(i),succ(i))}$;//Le prédécesseur du partant génère un nouveau grain et le combine avec l'ancienne clé de groupe
Pred(i)→succ(i): K;// puis l'envoie à son successeur
$K=K \oplus K_{(pred(i),succ(i))}$ // Le successeur récupère la clé K

$K=K \oplus S_{(succ(i))}$;// Le successeur rajoute son grain
i= succ(i);j=1;
While (*j*< N) Do
$K=K \oplus K_{(i, succ(i))}$;
i→succ(i): K;//i envoie la clé K à son successeur
i= succ(i);
j=j+1;
End,
refresh (id);
Stop.

2. Vérification du protocole

Il est certain que la preuve de sécurité du protocole de distribution de clé quantique a longtemps été avancée par les chercheurs, Shor et Preskil [77] ainsi que Dominics Mayers [66] et bien d'autres ont démontré dans leurs travaux la sécurité du fameux protocole BB84, ceci dit, la preuve mathématique d'un protocole n'est suffisante pour en assurer la sécurité lorsqu'il s'agit d'implémentation réelle d'un système basé sur un tel protocole, d'autant plus que des aléas, souvent matériels, risquent de survenir. Pour cela, l'analyse de tout système ainsi que sa vérification au préalable est une tâche bien importante, avant de passer à toute éventuelle implémentation.

A l'heure de la réalisation de ce travail, des simulateurs dédiés pour la vérification de système de distribution de clé quantique n'existent pas, et les travaux sont orientés vers une solution alternative, à savoir la vérification formelle de tels protocoles, ce qui semblent d'ailleurs être un exemple très intéressant pour l'application des méthodes de vérification formelles comme le souligne Nagarajan dans [8].

2.1. Qu'est ce que la vérification formelle

La vérification formelle s'est développée durant la dernière décennie pour passer du simple cadre conceptuel pour l'analyse de logicielle et matérielle des systèmes, à une pratique industrielle [78] elle est également perçue comme une alternative à la simulation permettant de surpasser les limites de cette dernière. La vérification formelle un processus systématique utilisant une des techniques mathématiques afin de vérifier que la conception (et de là même les spécifications) est préservée durant l'implémentation, assurant ainsi une justesse de fonctionnement. Concrètement, si l'on suppose un modèle de conception, une description de l'environnement dans lequel la conception est supposée opérer et quelques propriétés que la conception est censée assurer, l'idée de la vérification formelle et de déclencher une recherche exhaustive afin de trouver soit un cas où les propriétés ne sont pas assurées ou que ces dernières le sont toujours [79].

La première étape d'une vérification formelle est de définir un modèle pour le système à analyser, puis de choisir un outil de vérification permettant soit de vérifier que le comportement du système correspond réellement à l'attente, soit que des propriétés bien spécifiques sont satisfaites [8].

2.2. Vérificateur de modèle au service du BB84

Le but d'un vérificateur de modèle est d'explorer le comportement d'un système de manière exhaustive dans le but de trouver des erreurs. Le résultat d'une telle recherche est soit un modèle vérifié, soit un contre exemple démontrant que le système fait défaut [78]

Dans notre cas nous avons opté pour le vérificateur de modèle PRISM. La justification est fondée sur le fait que plus d'un travail a été mené dans ce sens. Plusieurs chercheurs ont utilisé le vérificateur de modèle PRISM en vue de la vérification de modèle de distribution de clé quantique. En générale, les travaux tournent autours de la vérification du BB84 et de certaines variantes du modèle de ce protocole, ou de propriétés différentes variant d'un travail à l'autre.

Dans [8] les auteurs avancent déjà l'usage du vérificateur PRISM, puis en 2004, Nikolaos K. Papanikolaou, sous la direction de Rajagopal Nagarajan, il présente des travaux sur l'utilisation des vérificateurs de modèle, entre autres le vérificateur PRISM mais aussi SPIN, pour la vérification du protocole BB84 dans ces travaux de recherche en Master [67]. Ce qui a fait l'objet d'un papier intitulé « An automated analysis of the security of quantum key distribution » en collaboration avec Garry Bowen et Simon J. Gay, où les auteurs discutent l'utilisation de PRISM, pour la vérification du protocole de distribution de clé, en l'occurrence le BB84 [10]. Les résultats obtenus autours d'une certaine propriété, exprimant le rapport entre la tergiversation de l'espion et le canal, montrent que celle-ci diminue de manière exponentielle avec le nombre de qubits transmis, mais aussi que la probabilité de détection de l'espion augmente exponentiellement avec le nombre de qubits. Ce qui confirme les résultats théoriques obtenus, au préalable, en faveur de la sécurité du BB84 [66].

Plus tard en 2010, Elboukhari *et al* reprennent les travaux de Nikolaos K. Papanikolaou, et utilisent le vérificateur de modèle PRISM ; toujours pour modéliser et vérifier le protocole de distribution de clé quantique, le BB84, mais en étudiant la propriété de détection de l'espion [70] . Toujours dans le contexte de la vérification du BB84 via PRISM, et dans un autre travail, les mêmes auteurs, Elboukhari *et al*, introduisent le paramètre d'efficacité du canal quantique et la force de l'espion [71].

Il s'agit donc d'un vérificateur de modèle qui a déjà donné ses preuves dans le contexte de la vérification de modèle de protocole de distribution de clé quantique, ce qui est exactement le contexte dans lequel déverse notre sujet de recherche, ce qui justifie le choix de cet outil en vue de la vérification du modèle de distribution de clé quantique que nous proposons.

3. PRISM : présentation de l'outil

PRISM est un vérificateur de modèle probabilistique, il s'agit d'un outil de vérification mais aussi de modélisation et d'analyse des systèmes ayant un comportement aléatoire. De plus, PRISM est un outil Open Source et gratuit, téléchargeable sur le site www.prismmodelchecker.org. L'outil en est à sa version 4.3.1 lancée le 27/05/2016 et ses développeurs Dave Parker (Université of Birmingham), Gethin Norman (Université of Glasgow) et Marta Kwiatkowska (Université of Oxford) ont toujours la charge de son développement et amélioration [80].

Afin de construire et d'analyser un modèle sous PRISM il convient de le décrire dans un premier temps dans un langage d'états basé sur le formalisme Alur and Henzinger [81].

En fait, sous PRISM, un modèle est décrit par un ensemble de modules qui interagissent entre eux et des variables qui peuvent être locales à un module précis ou globales, partagées entre les différents modules. Les valeurs de l'ensemble des variables à un instant t constitue l'état du module auquel elles appartiennent.

L'état global du modèle est déterminé quant à lui par les états locaux de tous les modules qui le constituent.

Le comportement du module se définit par un ensemble de transitions que le module effectue à chaque fois que le prédicat renfermant des variables (du module lui-même ou celles d'autres modules) est vrai.

Ces transitions sont en fait un ensemble de commandes ; où chaque commande prend la forme suivante :

$$[] \text{ guard} \rightarrow \text{prob\_1} : \text{update\_1} + \ldots + \text{prob\_n} : \text{update\_n}$$

Équation 16 Forme générale d'une commande PRISM [81].

Où:

[]: est une étiquette permettant une synchronisation de plusieurs commandes appartenant à des modules différents ;

guard : est le prédicat;

prob_1: est la probabilité que la transition update_i ait lieu ;

update_i : est une transition possible.

Exemple :

Pour le cas du protocole QDGKM par exemple on peut avoir la ligne suivante :

[] (al_state=0) & (eve_state>0)& (round2=false) -> 0.5 : (al_state'=1) & (al_bas'=0) +0.5 : (al_state'=1) & (al_bas'=1);

Cette ligne s'interprète comme suit :

Si Alice est dans l'état 0, Eve dans l'état 0 et que le round 2 n'est pas déclenché, alors Alice peut passer à l'état 1 en choisissant la base 0, avec la probabilité 0,5, ou bien passer à l'état 1 en choisissant la base 1 avec la probabilité 0,5.

Avec l'outil PRISM le modèle est saisi dans un éditeur spécial, permettant d'avoir tous les modules dans une seule section comme le montre la figure ci-dessous.

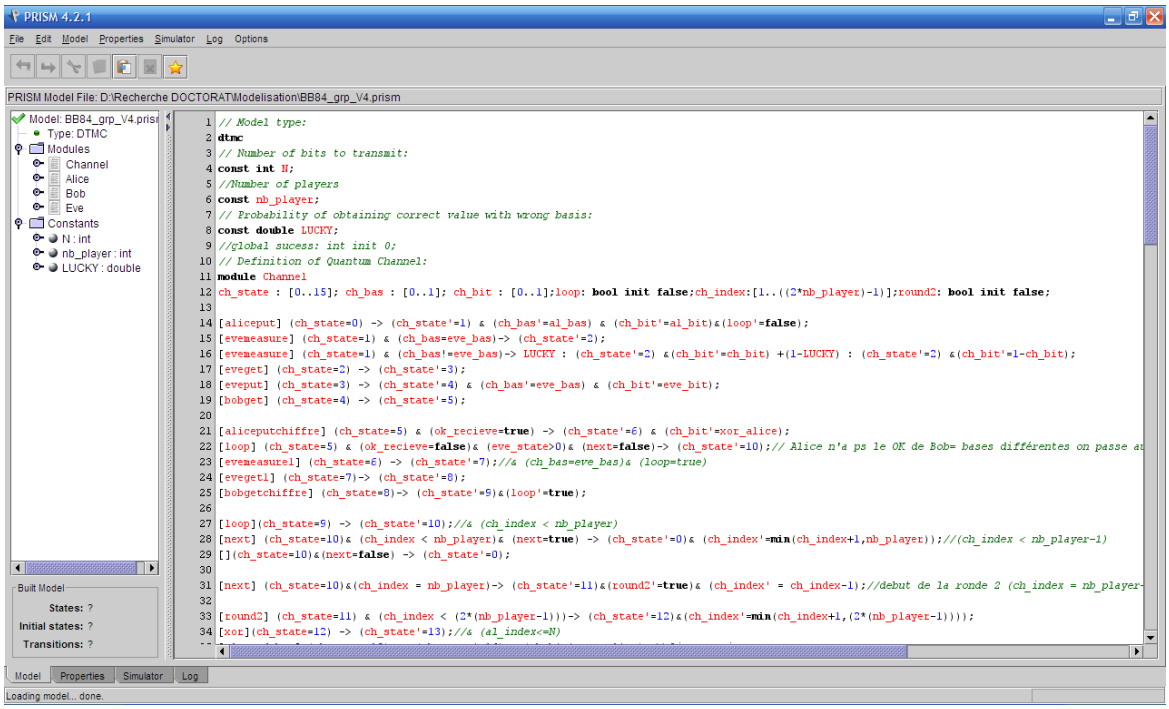


Figure 41 Prise de vue d'un modèle PRISM.

Afin d'évaluer un modèle PRISM, il faut définir une, ou plusieurs propriétés. Cela se fait via le langage de spécification des propriétés qui renferme plusieurs logique temporelles de spécification comme PCTL (***P**robabilistic **C**omputation **T**ree **L**ogic*), PCTL*, CSL (***C**omputation **T**ree **L**ogic*)[26]

Les propriétés sont à leur tour saisies dans un éditeur à part comme le montre la figure ci-dessous.

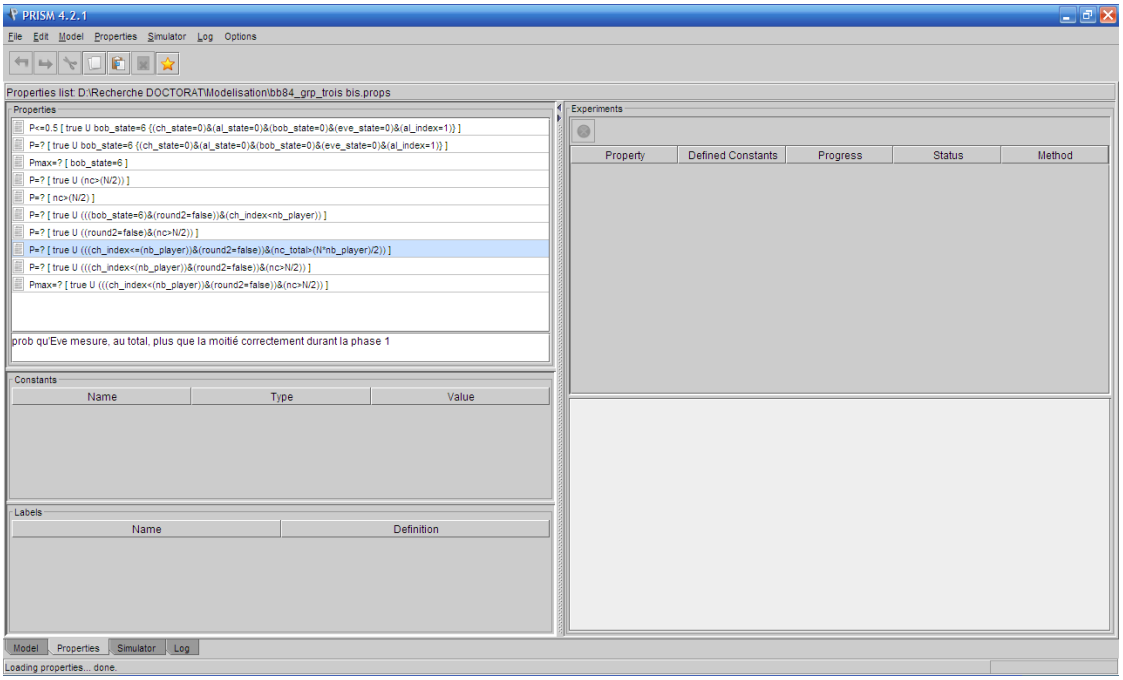


Figure 42 Espace spécification de propriétés PRISM.

[26] Nous notons ici que le présent travail ne couvre pas plus les langages de propriétés.

4. Vérification du protocole

Comme nous venons de l'introduire donc, nous avons opté pour l'outil PRISM pour la vérification du protocole proposé. Pour cela, nous avons mis en place un modèle du protocole, puis spécifié deux des propriétés de ce dernier. Les propriétés concernent la possibilité pour l'espion Eve de retrouver durant la création et la distribution de la clé une quelconque information signifiante la concernant.

4.1. Modèle du protocole QDGKM

Le modèle du protocole QDGKM que nous proposons est constitué de quatre (04) modules : Alice (11 états), Bob (09 états), Eve (10 états) et le canal de communication (15 états). Les modules peuvent communiquer entre eux via les étiquettes (notées []) qui permettent des exécutions synchronisées (par exécution nous insinuons ici le passage d'un état à un autre).

Le module du canal de communication est celui qui se charge de la synchronisation des différents autres modules. Il permet à Alice d'y *déposer* un qubit et de l'envoyer à Bob, et à ce dernier de le *récupérer*. Il permet, éventuellement, à Eve d'exécuter une attaque Intercept and Resend.

Les modules Alice et Bob exécutent le protocole BB84 afin de générer la clé quantique intermédiaire et de l'utiliser pour le chiffrement du grain. Sauf que, il faut le signaler, pour des raisons d'implémentation du protocole dans PRISM, mais aussi pour des raisons d'authenticité par rapport à la technologie disponible actuellement, l'exécution du BB84 se fera qubit par qubit. En effet, du côté implémentation, il n'est envisageable d'avoir une variable de type chaine, permettant de reproduire la chaine de qubits générée. D'autre part, si l'on optait pour la génération de la chaine de qubits de la clé intermédiaire puis son utilisation, cela impliquerait l'implémentation d'une mémoire quantique, ce qui est actuellement non faisable ou du moins pas évident vue la difficulté de la mise en place d'une telle technologie.

En conséquence, dans le modèle du QDGKM que nous avons implémenté sous PRISM, Alice choisit aléatoirement pour chaque qubit de la clé intermédiaire la valeur a envoyé et la base utilisée. Elle envoie la valeur à Bob qui tente, aléatoirement, de choisir la bonne base, il réussira une fois sur deux.

Alice et Bob vérifient qu'ils ont bien utilisé les mêmes bases en révélant au fur et à mesure la base utilisée. Si celle-ci est correcte, ils continuent le processus comme le prévoit le scénario standard du protocole BB84.

Si le qubit d'Alice et Bob est correct et qu'Eve ne l'a pas détecté, il servira directement au chiffrement du bit du grain d'Alice. Le chiffrement n'est autre qu'une opération de XOr comme nous l'avons déjà précisé dans la présentation du QDGKM. Bob recevant le bit ainsi chiffré, le déchiffre et y combine le bit de son grain, également, par une opération de XOr. Si par contre Eve est détectée par Alice et Bob, l'envoi sera annulé et un autre sera lancé.

L'opération est répétée autant de fois que le nécessite le nombre de bits du grain.

On concède à Eve le pouvoir d'intervenir et d'agir sur tous les qubits envoyés entre Alice et Bob sur le canal de communication. A chaque envoi, elle peut exécuter ou non, de manière aléatoire, une attaque de type Intercept and Resend sur le qubit envoyé. Si son attaque est menée à bien, et qu'elle n'a pas été détectée par Alice et Bob, la valeur mesurée sera comptabilisée à son compte. Si par contre elle est détectée, la valeur du bit sera annulée et Alice et Bob reprendront une autre tentative d'envoi.

Pour reprendre la structure en anneau du groupe, les modules sont lancés autant de fois que le nécessite le nombre de participants au groupe.

Une fois le protocole terminé, tous les modules sont arrêtés de manière synchrone à l'étiquette [stop]

Les différents modules et le fonctionnement général du modèle sont repris dans la figure ci-dessous.

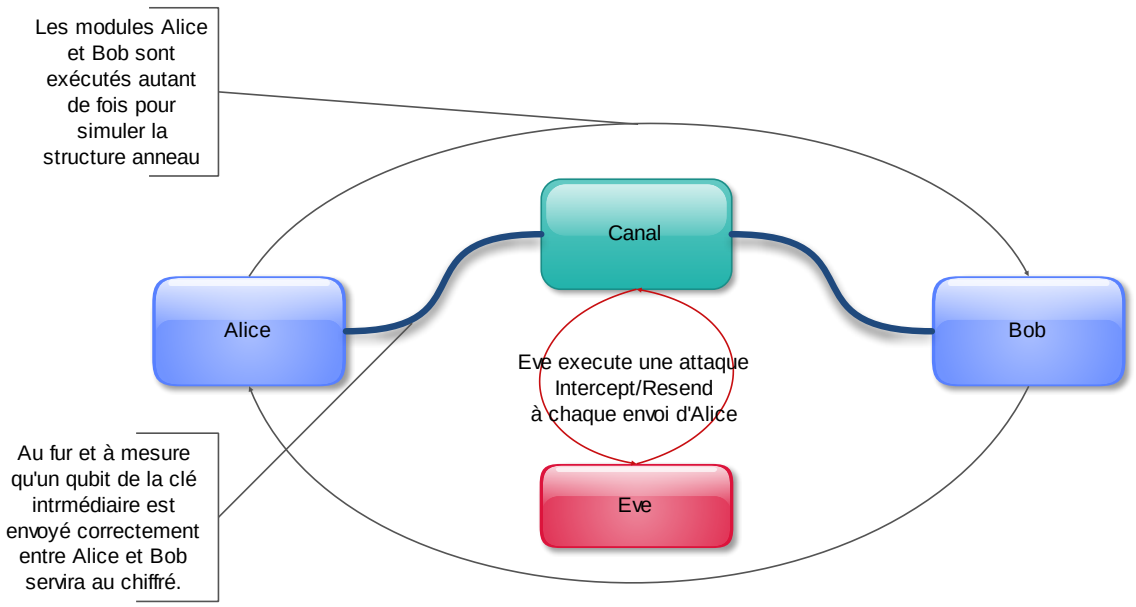


Figure 43 Schéma récapitulatif des modules du QDGKM.

4.2. Propriétés du protocole

Lors de l'étude du protocole QDGKM, et après sa modélisation, nous nous sommes focalisés sur l'étude de la capacité d'Eve d'intercepter une quelconque information utile lors des échanges entre les membres du groupe. Ainsi nous avons dégagé deux propriétés.

4.2.1.Première propriété

La première propriété est liée au premier round du protocole, à savoir lors de l'exécution du BB84 entre chaque deux participants, et donc lors de la création des clés quantiques intermédiaires.

Cette propriété concerne de ce fait la probabilité qu'Eve puisse mesurer correctement plus que la moitié des qubits échangés entre chaque deux participants du groupe.

Soit P1 la première propriété. P1 s'exprime dans le langage de spécification PRISM comme suit :

P1=? [true U (((ch_index<(nb_player)&(round2=false))& (nc>N/2))]

Où:

ch_index : est l'indice du canal de communication comptabilisant le nombre de participants ;

nb_player : est le nombre de participants. Il exprime la taille du groupe ;

round2 : désigne la deuxième étape du protocole. C'est une variable mise a « false » tant que l'on en est à l'exécution du BB84 entre deux voisins ;

nc: est le nombre de qubits mesurés correctement par Eve;

N : est le nombre de qubits, exprimant ainsi la longueur de la chaine échangée entre deux participants.

La figure suivante représente les résultats obtenus lors de la vérification de la propriété P1.

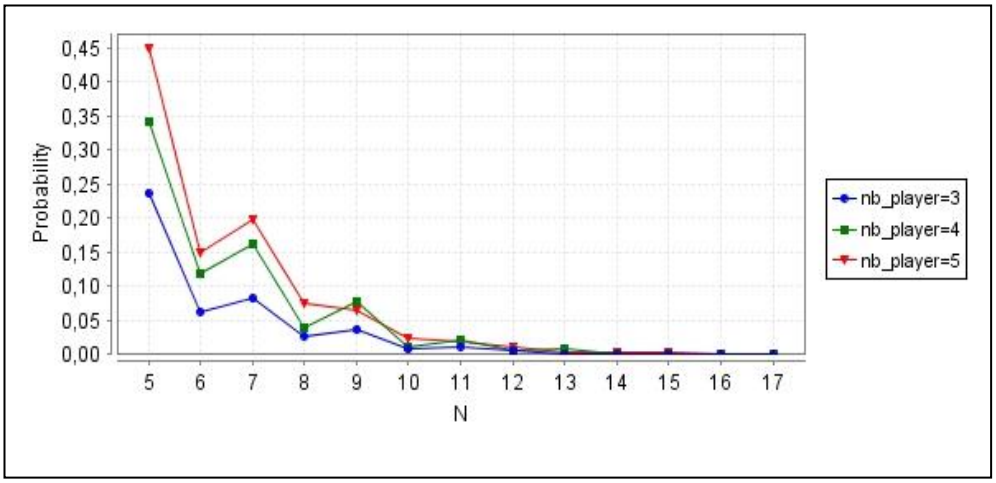


Figure 44 Vérification de la propriété P1 : Eve ne peut pas mesurer plus que la moitié des qubits transmis entre deux voisins.

Pour des raisons de limite matérielle, la vérification s'est faite pour des groupes de 3, 4, et 5 participants, pour des chaines de qubits allant de 5 à 17 qubits.

Suivant les résultats obtenus lors du déroulement de la vérification de la propriété P1 qui sont représentés dans la Figure 44 ci-dessus. Il est clair que la probabilité qu'Eve puisse mesurer correctement plus que la moitié des qubits échangés entre deux voisins ne dépasse pas 0,5. Ce qui conforte les thèses avancées au préalable de notre travail à propos de la sécurité du BB84, et rejoint les différents résultats obtenus jusqu'à maintenant concernant la sécurité du BB84 et que nous avons discuté dans la section Preuves de sécurité du BB84, ce qui constitue une preuve de plus.

Mieux encore, on remarque que la probabilité qu'Eve puisse mesurer correctement plus que la moitié des qubits échangés entre chaque deux voisins diminue de manière proportionnelle au nombre de qubits échangés jusqu'à pratiquement s'annuler.

On en déduit que plus le nombre de qubits constituant la chaine échangée entre deux voisins (ce qui représente en fait la clé intermédiaire servant au chiffrement lors de la phase de distribution) augmente, moins Eve aura de chance d'en mesurer correctement des qubits, diminuant ainsi l'information utile qu'un espion aurait pu obtenir.

4.2.2.Seconde propriété

Toujours en vue de la validation du QDGKM, et l'étude de sa sécurité, nous avons défini une seconde propriété. Cette propriété concerne un champ plus large que celui des deux voisins. Elle est liée à toute la première phase du protocole.

Cette propriété exprime donc la probabilité qu'Eve puisse mesurer correctement plus que la moitié de tous les qubits échangés.

On notera P2 la seconde propriété. P2 est exprimée comme suit :

P2=? [true U (((ch_index<(nb_player)&(round2=false))& (nc_total>(N*nb_player)/2))]

Où :

nc_total : est le nombre total de qubits mesurés correctement par Eve durant la première phase du processus.

Pour les mêmes raisons de limitation exprimées précédemment, nous avons maintenu les mêmes paramètres de vérification (des groupes de 3, 4, et 5 participants, pour des chaines de qubits allant de 5 à 17 qubits). Les résultats ainsi obtenus sont représentés dans Figure 45 ci-dessous.

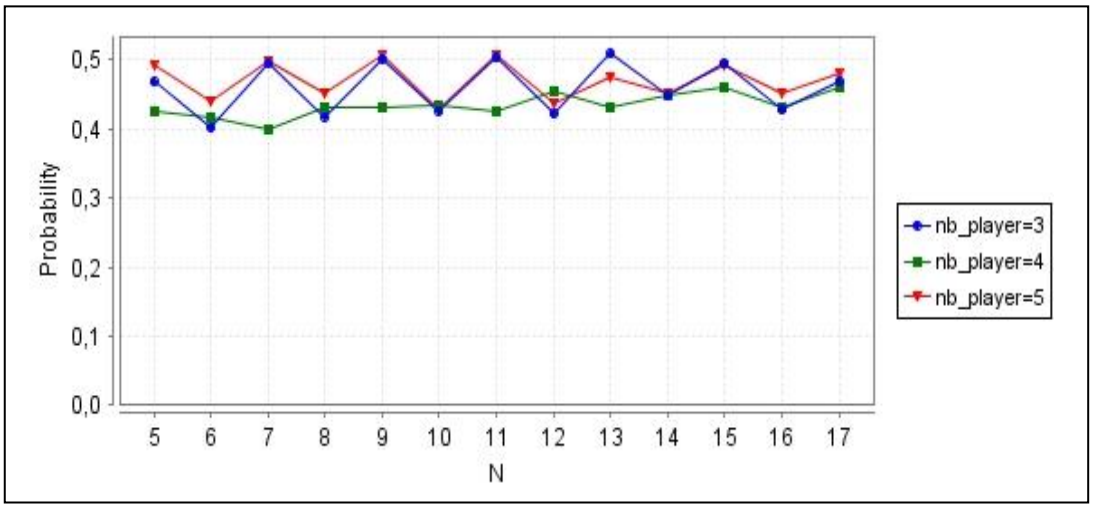


Figure 45 Vérification de la propriété P2: Eve ne peut pas mesurer correctement plus que la moitié des qubits de la clé.

Les résultats montrent également que la probabilité qu'Eve mesure correctement plus que la moitié de l'ensemble des qubits échangés peut avoisiner 0,5 mais ne dépasse pas ce seuil. Ce qui rassure encore plus sur la qualité des clés intermédiaires qui seront utilisées pour le chiffrement des grains. D'autant plus que le chiffrement se fait via XOr, ce qui constitue un bon OTP (***O**ne **T**ime **P**ad*) non seulement parce que les clés qui vont être utilisées sont totalement aléatoires mais aussi parce qu'elles sont sûres tout au long de leur génération durant cette première phase du QDGKM, ce qui permet d'avoir une clé finale sûre.

Suite aux résultats de vérification des deux propriétés P1 et P2, il est possible d'avancer que la solution proposée est une solution sécurisée de distribution de clé quantique dans un groupe. Ceci est dû au fait qu'il est impossible pour un espion d'obtenir la clé finale du groupe. Cela est justifié par le fait que la clé finale est une combinaison de grains générés chacun par un participant. Ces grains sont non seulement secrets mais aussi combinés à chaque fois via l'opération XOr avant d'être transmis d'un participant à un autre après avoir été chiffrés avec les clés quantiques intermédiaires. Les clés intermédiaires à leur tour sont sécurisées de par leur nature quantique, et dont la sécurité a également été vérifiée par la propriété P1.

Le choix de l'opération XOr lors des échanges chiffrés est justifié par la simplicité de l'opération mais aussi par le fait qu'il s'agit de l'application d'un chiffrement identique au OTP dont la sécurité n'est plus à prouver.

Ainsi, et suivant les résultats obtenus par la vérification de la propriété P1 et P2, l'espion Eve ne peut obtenir d'information significative à propos de la clé de groupe générée suite à l'exécution du protocole QDGKM.

Pour ces raisons, et d'autres que nous discutons en détails dans la section suivante, il est possible d'avancer que le QDGKM est une bonne solution de distribution de clé dans un groupe.

5. Discussion du protocole

Après la vérification des propriétés P1 et P2 que nous avons dégagé, il semble intéressant de discuter d'autres critères et caractéristiques de la solution proposée.

5.1. Clé par accord

La solution proposée n'est pas une solution centralisée où il est indispensable d'avoir un leader de groupe qui gère toutes les opérations. Tout au contraire, il s'agit d'une solution où la clé de groupe est obtenue par accord et suite à la contribution de tous les participants du groupe. Chaque membre apporte ainsi son grain à la clé finale. Cette dernière étant la combinaison de tous les grains transmis de manière sécurisée d'un participant à un autre via les clés intermédiaires sûres.

Le fait que la solution est une solution par accord permet d'éviter les contrariétés imposées par les solutions centralisées de distribution de clé tel que les goulots d'étranglement qui en constitue le premier inconvénient.

5.2. Contribution des membres

La solution proposée ne nécessite pas de Leader pour orchestrer les opérations. Tous les membres du groupe participent de manière égale à l'élaboration de la clé. Chaque participant apporte un grain de la clé, qui est une part secrète, mais aussi aléatoire, combinée aux autres afin d'obtenir la clé finale.

5.3. Dynamique et confidentialité

La solution proposée est sensible à la dynamique du groupe. Ainsi à chaque mouvement de départ ou d'adhésion d'un membre, un processus est déclenché afin qu'une nouvelle clé soit générée, ce qui est exprimé dans les algorithmes 2 et 3. Ceci permet d'assurer, respectivement, la confidentialité passée et future.

5.4. Sécurité

La sécurité du protocole est assurée par l'usage du XOr comme opération de chiffrement mappant le OTP. Un chiffre prouvé sécurisé où les clés doivent être aussi longues que le message à chiffrer mais aussi *aléatoires.* Une propriété assurée par l'utilisation des clés quantiques dont la sécurité a été déjà discutée. Ces clés représentent dans la solution proposée les clés intermédiaires, notées $K_{i,(i+1)}$ et sont utilisées pour chiffrer les grains de clé secrets générés par chaque participant.

D'autre part, La sécurité de la solution ainsi proposée a été vérifiée en explorant, via le vérificateur PRISM, deux propriétés très importantes que nous avons dégagées. Les propriétés concernant la possibilité que l'espion Eve puisse récupérer une information utile concernant la clé finale du groupe lors des échanges quantiques, ce qui s'avère une tâche laborieuse, sinon impossible tel que le montrent les résultats discutés dans la section précédente.

En effet, nous avons montré d'un côté que la probabilité que l'espion obtienne plus la moitié des qubits échangés entre deux voisins successifs ne dépasse pas 0,5 ; mieux encore, que cette probabilité décroit avec l'augmentation de la taille de la clé intermédiaire.

D'un autre côté nous avons également montré que l'espion ne peut obtenir de manière correcte plus que la moitié des qubits échangés lors de la première phase, ce qui constitue la totalité des échanges quantiques du QDGKM, et que cela ne dépasse pas non plus la probabilité de 0,5.

Ceci nous permet de dire que l'espion ne peut obtenir une quelconque information utile concernant la clé finale.

6. Comparaison

La solution que nous proposons, et que nous avons baptisé QDGKM, est un protocole de distribution de clé dans un groupe organisé en anneau, où tous les membres du groupe participent à l'élaboration de la clé finale. D'après la recherche menée, il n'y a pas de solutions semblables à celle que nous proposons, à savoir une solution quantique et par accord à la problématique de distribution de clé dans un groupe. Par contre, dans le domaine classique, des solutions de distribution de clé dans un groupe, et plus particulièrement, des solutions par accord, ont été proposées.

Par conséquent, les performances du QDGKM sont comparées avec celles d'autres protocoles de distribution de clé dans un groupe de la même famille, c'est-à-dire des protocoles par accord, mais non quantiques.

Nous nous basons sur les comparaisons réalisées par Yacine Chellal dans [36]

Le Tableau 10 ci-dessous résume cette comparaison.

Protocole	Nb. Round	Nb. messages		Echange Diffi-Hellman	Besoin de Leader
		multicast	**unicast**		
Ingemarson *et al*	n-1	0	n(n-1)	Oui	Non
GDH	N	n	n-1	Oui	Non
Octopus	2(n-1)/4+2	0	3n-4	Oui	Oui
STR	N	n	0	Oui	Non
DH-LKH	Log n	Log n	0	Oui	Non
D-LKH	3	1	n	Non	Oui
D-OFT	Log n	0	2 log n	Non	Non
D-CFKM	N	0	2n-1	Non	Non
Fiat *et al*	2	n	n	Oui	Oui
Burmester *et al*	3	2n	0	Non	Non
CKA	3	n	n-1	Non	Oui
QDGKM (Quantum Distributed Group Key Management)	2	0	2(n-1)	Non	Non

Tableau 10 Comparaison des performances du QDGKM avec d'autres protocoles de distribution de clé dans un groupe.

Comme la solution proposée est sensible à la dynamique du groupe, et que chaque opération d'adhésion ou de départ (sinon expulsion) d'un membre déclenche un nouveau processus de création de clé afin de préserver la confidentialité passée et future, nous avons jugé important, pour

des comparaisons ultérieurs, de calculer les couts de chaque opération en termes de nombre de message nécessaires.

Nous supposons que N est le nombre des participants du groupe et m le nombre des participants du nouveau groupe en cas de fusion ou de partitionnement.

Dans ce cas les couts sont comme suit :

Propriétés	Cout = nombre de messages échangés
Création de clé	2(N-1)
Adhésion	(N+1)
Séparation/Expulsion	N
Fusion	N+(m*2)
Partitionnement	N-m

Tableau 11 Couts du protocole QDGKM en termes de messages.

Ces résultats pourraient éventuellement servir ultérieurement pour des comparaisons du protocole avec d'autres de la même famille, autrement dit par accord, et de même type, c'est-à-dire quantique, en vue d'en mesurer les performances et d'en déduire la/les meilleures solutions.

Conclusion

Nous avons présenté dans le dernier chapitre de cette thèse l'essentiel de ce travail consistant en la proposition d'une solution de distribution de clé quantique dans un groupe organisé en anneau mais également la validation de ladite solution proposée. Cette dernière est une solution par accord, cela revient à dire que tous les membres du groupe participent à l'élaboration de clé finale par combinaison de plusieurs parties aléatoires secrètement échangées entre voisins. La sécurité des parties est assurée par des clés intermédiaires quantiques dont la sécurité n'est plus à prouver car elle se base sur le protocole de distribution de clé quantique BB84 dont la sécurité fut largement étudiée dans la littérature du domaine, mais aussi affirmée par l'étude menée.

Une fois la présentation du protocole faite, sa sécurité devait être étudiée. Une simulation n'aurait pu être réalisée pour plus d'une raison, la première étant la non-disponibilité d'outils dédiés à cet effet. Nous avons alors optés, comme beaucoup d'autres travaux dans le même contexte, pour une vérification formelle, d'autant plus que le domaine de la vérification formelle a bien donné ses preuves. Une vérification formelle permet de vérifier le bon fonctionnement d'un modèle, et de retrouver des contre exemples le cas contraire. Nous avons utilisé pour cela un vérificateur de modèle ayant déjà servi à la vérification de protocoles quantiques, entre autre le BB84. Il s'agit du vérificateur de modèle PRISM.

Afin d'utiliser PRISM, il fallait en premier lieu modéliser le protocole. C'est ce que nous avons fait, et le modèle ainsi obtenu compte quatre (04) modules désignant les acteurs principaux du protocole, en l'occurrence : Alice, Bob, Eve et le canal. La communication entre Alice et Bob est répétée autant de fois pour reprendre la structure en anneau du groupe. A Eve nous avons légué l'avantage de pouvoir intervenir sur tous les échanges qui se font sur le canal.

Une fois le protocole modélisé, nous nous sommes intéressés à sa validation via PRISM. Ce qui nous intéresse dans le cadre de ce travail est l'aspect sécurité du protocole. Pour cela, la sécurité s'est portée sur deux propriétés concernant la possibilité que l'espion Eve, puisse retrouver une quelconque information signifiante lors des échanges quantiques. Il s'avère après vérification que la probabilité qu'Eve puisse mesurer correctement plus que la moitié des qubits de la chaine finale, et donc de la clé finale, ne dépasse pas les 0,5 et que la probabilité qu'elle puisse mesurer correctement plus que la moitié des qubits des clés intermédiaires est inversement proportionnelle au nombre de qubits constituants la clé intermédiaire. Les résultats ainsi obtenus rassurent sur la qualité de la clé ainsi générée.

D'autres caractéristiques du protocole ont également été discutées à la fin du chapitre que nous avons parachevé par une comparaison dudit protocole avec d'autres protocoles de même famille.

Conclusion générale et perspectives

La confidentialité de l'information peut se définir aisément comme étant le fait que seule une entité ou un groupe d'entités autorisées, puissent accéder à l'information. Pour atteindre ce but, l'information en question doit être chiffrée. C'est là qu'intervient la cryptographie. Qu'elle soit à clé secrète, dite symétrique, où une seule clé est utilisée pour chiffrer et déchiffrer, ou à clé publique, appelée également asymétrique, où deux clés sont utilisées distinctement, l'une pour chiffrer l'autre pour déchiffrer, la sécurité des messages repose essentiellement sur la difficulté pour un espion de trouver la/les clés utilisées.

Kryptos une étrange sculpture se dressant dans l'enceinte du quartier général de la CIA, à Langley, Virginie. Une œuvre de l'artiste américain Jim Sanborn composée d'un petit bassin à bulles entouré de bois pétrifié, de blocs de granite, et d'une large plaque de cuivre en forme de S dans laquelle ont été découpées environ 1700 lettres de l'alphabet reste encore non déchiffrée. Le chiffre d'Alexander d'Agapeyeff inscrit à la fin de son ouvrage qu'il publia en 1939 intitulé 'Codes and ciphers' est un autre chiffre mystère. Plus loin dans l'histoire, 1897, Edward Elgar adressa une lettre à Dora Penny, qu'il surnommait Dorabella. Le message qui y figurait était composé d'étranges caractères en demi-cercles répartis sur 3 lignes [82] et bien d'autres exemples de chiffres dont le but était certes la transmission d'une information de manière confidentielle à un destinataire bien précis sont restés indéchiffrables jusqu'à aujourd'hui, et pour cause, la non disponibilité de la clé de déchiffrement.

L'importance de la clé de chiffrement à fait que plusieurs protocoles ont été proposés, afin de permettre aux protagonistes de s'échanger et d'élaborer des clés plus ou moins sûres. Mais avec l'avancée technologique et la capacité de calcul des ordinateurs, les algorithmes de chiffrement les plus sûrs aujourd'hui ne sont pas à l'abri d'une éventuelle attaque, permettant de retrouver aussi rapidement les clés de chiffrement/déchiffrement.

Pour cela, une nouvelle technologie a vu le jour, se basant non plus sur la complexité des algorithmes de chiffrement et encore moins sur les problèmes mathématiques sur lesquels se basent les protocoles de gestion de clé classiques. Il s'agit de la distribution de clé quantique.

En effet, la distribution de clé quantique exploite les principes de la mécanique et de la physique quantique, offrant ainsi une solution physique à la problématique de distribution de clé. La solution propose la possibilité de générer et de distribuer une clé sûre et sécurisée entre participants. Grace aux principes de a mécanique quantique, toute intervention de l'espion sera inéluctablement détectée avant même la fin de l'élaboration de la clé.

Pour longtemps, la distribution de clé quantique a été considérée comme de simples expériences permettant l'échange de clé de petite taille sur une distance de quelques centimètres. Actuellement, il s'agit plus d'une solution à prévoir si l'on veut assurer une sécurité inconditionnelle de l'information, d'autant plus que des industriels importants comme Idquantique et MagiQ, offrent des

solutions sécuritaires basées sur la distribution de clé quantique et mettent sur le marché des dispositifs matériels dédiés à cet effet.

Dans le care de ce travail, nous nous sommes penchés sur l'étude de cette solution sécuritaire, plus particulièrement sur la famille P&M de protocole de distribution de clé quantique. Le plus connu des protocoles de cette famille étant le BB84 reconnu pour être le premier protocole de distribution de clé quantique entre deux participants, les traditionnels Alice et Bob. Essentiellement, notre premier objectif d'étude consistait à explorer la possibilité d'appliquer un protocole de distribution de clé quantique dans un contexte de groupe.

Au préalable deux propositions ont été avancée [6] et [7] mais la solution que nous proposons, elle, se base sur le protocole BB84, et s'inscrit, à la différence des deux autres, dans la famille des protocoles par accord. Ainsi tous les membres d'un groupe de communication participent à l'élaboration de la clé en combinant des grains de clé aléatoires qu'ils échangent après chiffrement via des clés quantiques. Apres deux rounds, l'ensemble du groupe dispose de la même et unique clé.

Le deuxième objectif de notre étude était de s'assurer de la sécurité de la solution proposée. Pour ce faire, nous avons opté pour la vérification formelle au lieu de la simulation et ce pour deux raisons essentielles, la première étant qu'il n'existe pas de simulateurs dédiés ; la seconde est que dans ce domaine, la vérification formelle s'avère un meilleur choix par rapport à la simulation, entre autres parce qu'elle permet une exploration exhaustive des états du protocole. Ainsi le choix s'est porté sur un vérificateur formel probabilistique, en l'occurrence PRISM. PRISM a déjà été utilisé dans le même contexte de vérification de protocoles quantiques dans des travaux antérieurs au notre surtout dans ceux de Nagarajan Rajagopal qui est un précurseur en la matière. Le vérificateur PRISM permet de décrire dans un langage le protocole sous forme de modèle décrivant les états de ce dernier, puis de spécifier les propriétés du modèle dans un autre langage. Le lancement de la vérification permet d'explorer exhaustivement tous les états du modèle et de trouver les cas où il ne répond pas aux spécificités attendues.

Donc, pour vérifier la solution proposée, et que nous avons baptisé QDGKM, nous avons, dans un premier temps, décrit la solution sous forme de modèle en utilisant le langage de description de PRISM. Ce qui a donné lieu à un modèle comptant quatre (04) modules. Par la suite nous nous sommes intéressés particulièrement à deux propriétés : la possibilité que l'espion Eve puisse mesurer correctement plus que la moitié des qubits constituants la clé intermédiaires de deux membres successifs et la possibilité qu'Eve puisse mesurer correctement plus que la moitié des qubits échangés durant la première phase du protocole. La vérification de la première propriété a montré que plus la longueur de la clé est importante moins Eve a de la chance d'avoir des mesures correctes, alors qu'il paraît clair selon les résultats obtenus pour la vérification de la seconde propriété que la possibilité que Eve mesure correctement plus que la moitié des valeurs constituant la clé ne dépasse pas 0,5, ce qui confirme la sécurité de la solution proposée. D'autre part, la solution répond à plusieurs autres caractéristiques comme le fait d'être sensible à la dynamique du groupe afin d'assurer la confidentialité passée et future, ce qui n'est pas négligeable même si le cout en termes de temps de calcul serait probablement considérable. Ce qui constitue en soit une perspective de travail futur.

Pour être plus complète encore, la présente recherche pourrait être renforcée par des calculs d'entropie, permettant ainsi de pouvoir connaitre précisément la quantité d'information qu'Eve

aurait pu avoir lors d'une écoute, et les conséquences d'une telle écoute. Il serait alors possible de déterminer un taux d'écoute tolérable.

Dans le cadre de notre recherche nous nous sommes intéressés à la seule problématique de distribution de clé afin d'assurer la confidentialité via une clé sûre en faisant abstraction de l'aspect authentification en supposant, comme dans d'autres travaux antérieurs au notre, que les participants à une communication sont tous authentifiés. Ce qui serait intéressant d'avantage est d'inclure dans le protocole, pour une complétude, une primitive cryptographique assurant l'authentification.

La modélisation des protocoles quantiques n'est pas une tâche facile en l'absence d'outils dédiés. Ce qui se fait actuellement est d'adopter des outils comme PRISM afin de modéliser au mieux les protocoles quantiques, même si certains efforts se font comme le QMC [68] mais l'utilisation d'autres vérificateurs et étudier la faisabilité de la modélisation du protocole reste toujours une alternative de choix et une perspective assez intéressante que nous avons discuté avec Dr Florian Kammueller, qui travaille également en étroite collaboration avec Nagarajan Rajagopal, initiateur des travaux de modélisation des protocoles quantiques sur PRISM, à l'université Middelsex, en Grande Bretagne. Des outils comme Isabelle ou Coq semblent être une piste à explorer.

Bibliographie

1. **Hermant, Laurence et Lardillon, Cécile.** Télécoms, réseaux, multimédia. *https://www.inria.fr.* [En ligne] 08 Janvier 2010. [Citation : 25 Septembre 2016.] https://www.inria.fr/actualite/mediacenter/securite-des-systemes-cryptographiques.

2. **Zyga, Lisa.** New largest number factored on a quantum device is 56,153. *http://phys.org.* [En ligne] 28 Novembre 2014. [Citation : 24 Septembre 2016.] http://phys.org/news/2014-11-largest-factored-quantum-device.html.

3. **Cartlidge, Edwin.** Un saut quantique pour l'industrie. *http://www.snf.ch.* [En ligne] 12 Septembre 2016. [Citation : 25 Septembre 2016.] Site du Fond National Suisse de la recherche scientifique. http://www.snf.ch/fr/pointrecherche/newsroom/Pages/news-160912-horizons-un-saut-quantique-pour-la-industrie.aspx.

4. *Quantum cryptography: Public-key distribution and coin tossing.* **Bennett, Charles et Brassard, Gilles.** Bangalore : s.n., 1984. IEEE International Conference on Computers, Systems and Signal Processing. pp. 175-179. In proceedings of IEEE International Conference on Computers, Systems and Signal Processing.

5. **Leverrier, Anthony.** *Etude théorique de la distribution quantique de clés à variables continues.* Ecole Nationale Supérieure des Télécommunication. Paris : s.n., 2009. Thèse de doctorat.

6. *Architecture of multicast centralized key management scheme using quantum key distribution and classical symmetric encryption.* **Metwaly, Ahmed Farouk, et al.** 8, s.l. : Springer-Verlag, 19 Mars 2014, European Physical Jornal Special Topics, Vol. 223, pp. 1711-1728. DOI: 10.1140/epjst/e2014-02118-x. ISSN 1951-6355.

7. *Architecture of Decentralized Multicast Network Using Quantum Key Distribution and Hybrid WDMTDM.* **Metwaly, Ahmed Farouk et Mastorakis, Nikos E.** [éd.] Mastorakis Nikos E et J.Rudas Imre. Dubai : WSEAS Press, 2015. Advances In Information Science And Computer Engineering. pp. 504-518. Proceedings of the 9th International Conference on Computer Engineering and. ISSN 1790-5109.

8. **Nagarajan, Rajagopal et Gay, Simon.** Formal verification of quantum protocols. 2002. arXiv preprint quant-ph/0203086.

9. **J.Gay, Simon, Nagarajan, Rajagopal et Nikolaos, Papanikolaou.** Probabilistic Model–Checking of quantum protocol. 05 October 2005. http://arxiv.org/abs/quant-ph/0504007v2. quant-ph/0504007v2.

10. **Nagarajan, Rajagopal, et al.** An Automated Analysis of the Security of Quantum Key Distribution. 2005. arXiv preprint cs/0502048.

11. **Laurent, Bloche et Christophe, Wolfhugel.** *Sécurité informatique: principes et méthodes.* Paris : Edition Eyrolles, 2007. 2-212-12021-4.

12. **Willam, Stallings.** *Cryptography and Network security Principles and practice.* New York : Pearson, 2011. 978-0-13-705632-3.

13. **NIST.** An Introduction to Computer Security:The NIST Handbook. Special Publication 800-12.

14. **Alfred, J. Menezes, Paul C, van Oorschot et Scott A, Vanstone.** *Handbook of Applied cryptography.* s.l. : CRC Press, Août 1996. 0849385237.

15. **Henk C.A., van Tilborg.** *Encyclopedia of cryptography and security.* s.l. : Springer, 2005. 978-0387-23483-0.

16. **Hassler, Vesna.** Introduction to communication security. [auteur du livre] Sklavos Nicolas et Zhang Xinmiao. *Wireless security and cryptography: Specification and implementation.* s.l. : CRC Press, 2007.

17. **Martin, Bruno.** *Codage, cryptographie et applications.* s.l. : Presses polythechniques et universitaires romandes, 29 avril 2004. 2880745691.

18. **Nigel, Smart.** *Cryptography:An introduction.* s.l. : McGraw Hill, 2009. 0077099877.

19. **Frédéric, Grosshans.** *Communication et cryptographie quantique avec variables continues.* Paris : Institut d'optique Laboratoire Charles Fabry, Université Paris XI UFR Scientifique d'Orsay, 2002. 7080.

20. *La cryptographie militaire.* **Kerckoffs, Auguste.** s.l. : Journal des Sciences Militaires, Janvier-Fevrier 1883, Vol. 9, pp. 5-38 161-191.

21. **Stefan, Katzenbeisser et Fabien A.P., Petitcolas.** *Information Hiding Techniques for Steganography and Digital Watermarking.* Boston-Londre : Artech House, 2000. 1-58053-035-4.

22. **Mohamed Abdulla, Suhail.** Digital Watermarking for Protection of Intellectual Property. [auteur du livre] Chun-Shien Lu. *Multimedia security: Steganography and digital watermarking techniques for protection of intellectual property.* London : Idea Group Publishing, 2004.

23. **Scotte, Crover.** On Public-Key Steganography in the Presence. s.l. : Technical Report RC 20931,IBM, 1997.

24. **Remacle, Philippe, et al.** ENÉE LE TACTICIEN: La défense des places. *http://remacle.org/.* [En ligne] 2003. [Citation : 13 Octobre 2016.] Oeuvre numérisée par Marc Szwajcer. http://remacle.org/bloodwolf/erudits/enee/defensedesplaces.htm.

25. **La Coupole et enseignants-chercheurs en Informatique de l'Université des Sciences et Techniques de Lille.** Codes secrets et cryptologie: La stéganographie à travers les âges. *http://www.lifl.fr.* [En ligne] [Citation : 13 Octobre 2016.] http://www.lifl.fr/~wegrzyno/enigma/enigma/stegano.html.

26. **Davide, Maltoni, et al.** *Handbook of fingerprint recongnition.* London : Springer-Verlag, 2009. 978-1-84882-253-5.

27. **John R., Vacca.** *Public Key Infrastructure.* s.l. : Auerbach Publications, 2004. 0-8493-0822-4.

28. **Ralph C, Merkle.** Protocols for public key cryptosystems. s.l. : IEEE Symp. on Security and Privacy, 1980.

29. *Multiuser cryptographic techniques.* **Whitfield, Diffie et Martine E, Hellman.** s.l. : AFIPS '76: Proceedings of the June 7-10, national computer conference and exposition, 1976.

30. **Bruce, Schneier.** *Cryptographie Appliquée.* Paris : International Thomson Publishing, 1995. 2-84180-000-8.

31. *How to share a secret.* **Adleman, Shamir.** 11, s.l. : Communications of the ACM, 1979, Vol. 22.

32. *Safeguarding Cryptographic Keys.* **Blakley, George R.** Arlignton. VA : National Computer Conference, 1979. America Federation of Information Processing Societies Proceedings National Computer Conference. Vol. 48, pp. 313-317.

33. **Lakshminath R, Dondeti, Sarit, Mukherjee et Ashok, Samal.** *Survey and comparision of secure group communication.* 1999.

34. *A Survey of Key Management for Secure Group Communication.* **Sandro, Rafaeli et Davis, Hutchison.** 3, September 2003, ACM Computing Surveys, Vol. 35, pp. 309–329. 0360-0300.

35. *A survey of group key management.* **Bio, Jiang et Hu, Xiulin.** s.l. : IEEE Computer Society, 2008. International Conference on Computer and Software Engineering. 978-0-7695-3336-0.

36. *Group Key Management Protocols: A Novel Taxonomy.* **Yacine, Challal et Hamida, Seba.** 1, 2005, INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY, Vol. 2, pp. 105-118. 1305-2403.

37. **Gharout, Said.** Sécurité des communications dans les groupes dynamiques. Compiègne, France : UTC Université des Technologie Compiègne, 2009. Thèse de Doctorat.

38. *Cramming more components onto integrated circuits.* **Gordon E, Moor.** 8, s.l. : Electronics, 1965, Vol. 38.

39. **Le Bellac, Michel.** *Physique quantique.* 2e édition. s.l. : CNRS Edition, 2007. ISBN 978-2-271-06584-1.

40. **Guilleux, Joël.** Quelques Divinités du panthéon : Aton. *http://antikforever.com.* [En ligne] [Citation : 13 Octobre 2016.] http://antikforever.com/Egypte/Dieux/aton.htm.

41. **Cohen-Tannoudji, Claude, Diu, Bernard et Laloe, Franck.** *Mécanique quantique.* Paris : Hermann, 1977. 2-7056-5733-9.

42. **Louis-Gavet, Guy.** *La physique quantique.* Paris : Eyrolles, 2012. 978-2-212-55276-8.

43. **Michael A, Nielsen et Isaac L, Chuang.** *Quantum Computation and Quantum Information.* New York : Cambridge University Press, 2010. 978-1-107-00217-3.

44. *A single quantum cannot be cloned.* **Wootters, William K. et Zurek, Wojciech H.** 5886, 28 October 1982, Nature, Vol. 299, pp. 802-803. doi:10.1038/299802a0.

45. **Alexandre, Blais.** http://www.physique.usherbrooke.ca/~ablais/. *http://www.physique.usherbrooke.ca.* [En ligne] Fevrier 2014. [Citation : 23 Janvier 2016.] https://dl.dropboxusercontent.com/u/2644574/Web/Enseignement/MQII/MQII.pdf.

46. **Loepp, Susan et Wotters, William K.** *Protectiing Information from classical error correction to quantum cryptography.* s.l. : Cambridge University Press, 2006. ISBN 978-0-511-22485-0.

47. **Cobourne, Sheila.** *Quantum Key Distribution Protocols and Applications.* Department des Mathematiques, Royal Holloway, Université de London. Londres : s.n., 2011. Rapport technique. http://www.rhul.ac.uk/mathematics/techreports. RHUL-MA-2011-05.

48. **Bouwmeester, Dirk, Ekert, Artur et Zeilinger, Anton.** *The Physics of Quantum Information: quantum cryptography, quantum teleportation, quantum computing.* Springer Verlag. 2000. ISBN 3-540-66778-4.

49. *Experimental demonstration of long-distance continuous-variable quantum key distribution.* **Jouguet, Paul, et al.** 5, 2013, Nature Photonics, Vol. 7, pp. 378-381. 10.1038/NPHOTON.2013.63.

50. **Cartlidge, Edwin.** Secure quantum communications go the distance. *Physicsword.com.* [En ligne] 13 Novembre 2014. [Citation : 07 Aout 2016.] http://physicsworld.com/cws/article/news/2014/nov/13/secure-quantum-communications-go-the-distance.

51. *Provably Secure and Practical Quantum Key Distribution over 307 km of Optical Fibre.* **Korzh, Boris, et al.** 3, 2015, Nature Photonics, Vol. 9, pp. 163–168.

52. *Multiplexing, Multi-User Quantum Key Distribution Using Wavelength Division.* **Brassard, Gilles, et al.** [éd.] SPIE. 6, s.l. : SPIE Proceedings, 15 Decembre 2003, SPIE Applications of Photonic Technology, Vol. 5260, pp. 149-153. SPIE Proceedings. doi: 10.1117/12.543338.

53. *Outline of the SECOQC quantum-key-distribution Network in Vienna.* **Andreas, Poppe, Momtchil, Peev et Oliver, Maurhart.** 2, 01 Avril 2008, International Journal of Quantum Information, Vol. 6, pp. 209-218. Disponible sur: http://arxiv.org/abs/0804.0122v1. DOI: 10.1142/S0219749908003529.

54. *How to share a quantum secret.* **Cleve, Richard, Gottesman, Daniel et Lo, Hoi-Kwong.** 3, 19 Juillet 1999, PHYSICAL REVIEW LETTERS, Vol. 83, pp. 648-651. Disponible sur arxiv:quant-ph/9901025v1.pdf. ISSN 0031-9007.

55. *On the Theory of Quantum Secret Sharing.* **Daniel, Gottesman.** 4, 2000, Physical Review A, Vol. 61, p. 42311. Disponible sur arxiv:quant-ph/9910067v1.pdf.

56. *Multiparty quantum secret sharing of secure direct communication using single photons.* **Lian-Fang, Han, et al.** 9, s.l. : Elsevier, 1 May 2008, Optics Communications, Vol. 281, pp. 2690-2694. doi:10.1016/j.optcom.2007.12.045 . 0030-4018.

57. *Quantum secret sharing between multiparty and multiparty with entanglement swapping.* **Song, LIN, et al.** 04, s.l. : ELSEVIER, December 2008, The Journal of China Universities of Posts and Teelecommunications, Vol. 15, pp. 63-68. ISSN 1005-8885.

58. *An efficient and secure multiparty quantum secret sharing scheme based on single photon.* **Wang, Tian-yin, et al.** 24, s.l. : Elsevier, 9 Septembre 2008, Optics Communications, Vol. 281, pp. 6130–6134. doi:10.1016/j.optcom.2008.09.026. ISSN 0030-4018.

59. *Quantum key agreement protocol based on BB84.* **Chong, Song-Kong et Hwang, Tzonelih.** 6, s.l. : Elsevier, 15 March 2010, Optics Communications, Vol. 283, pp. 1192-1195 . doi:10.1016/j.optcom.2009.11.007. 0030-4018.

60. **Ekert, Arthur.** Quantum Cryptography. [éd.] Alexander V. Sergienko. *Quantum Communications and Cryptography.* s.l. : CRC Taylor & Francis, 2006, 1, pp. 1-13.

61. *Secret-key reconciliation by public discussion.* **Brassard, Gilles et Salvail, Lois.** [éd.] Springer. Berlin Heidelberg : s.n., 1993. pp. 410-423. In Workshop on the Theory and Application of of Cryptographic Techniques.

62. *Experimental Quantum Cryptography.* **Bennett, Charles, et al.** [éd.] Springer-Verlag. 1, 1992, Journal of cryptology, Vol. 5, pp. 3-28. ISSN 0933-2790.

63. **Claude, Crépeau.** Réconciliation et distillation publiques de secret. 1995. Disponible sur :www.cs.mcgill.ca/~crepeau/GZIP/Cre95.ps.gz.

64. *Unconditional security of quantum key distribution over arbitrarily long distances.* **Hoi-Kwong, Lo et Hoi Fung, Chau.** 5410, 1999, Science, Vol. 283, pp. 2050-2056.

65. *A Proof of the Security of Quantum Key Distribution.* **Biham, Eli, et al.** [éd.] ACM. 2000. Proceedings of the thirty-second annual ACM symposium on Theory of computing. pp. 715-724. ISBN/ISSN: 1581131844.

66. *Unconditional security in quantum cryptography.* **Mayers, Dominic.** 3, 2001, Journal of the ACM (JACM), Vol. 48, pp. 351-406. ISSN 0004-5411.

67. **Papanikolaou, Nikolaos K.** *Techniques for design and validation for quantum protocols.* Warwick : s.n., 2004. Mémoire de Master.

68. *QMC: A Model Checker for Quantum Systems.* **Gay, Simon J., Nagarajan, Rajagopal et Papanikolaou, Nikolaos.** [éd.] Springer Berlin Heidelberg. 2008. In International Conference on Computer Aided Verification. pp. 543-547.

69. **Papanikolaou, Nikolaos K.** *Model Checking Quantum Protocols.* Warwick : s.n., 2008/2009. Thèse de Doctorat.

70. *Analysis of the security of BB84 by model Checking.* **Elboukhari, Mohamed, Azizi, Mostafa et Azizi, Abdelmalek.** 2, Avril 2010, International Journal of Network Security & Its Applications, Vol. 2, pp. 87-98.

71. *Verification of quantum cryptography protocols by model checking.* **Elboukhari, Mohamed, Azizi, Mostafa et Azizi, Abdelmalek.** 4, Octobre 2010, International Journal of Network Security & Its Applications, Vol. 2, pp. 43-53. DOI : 10.5121/ijnsa.2010.2404.

72. *Quantum cryptography using any two nonorthogonal states.* **Bennett, Charles. H.** 21, 25 Mai 1992, Physical Review Letters, Vol. 68, p. 3121.

73. *A Survey of Quantum Key Distribution Protocols.* **Javed, Mobin et Aziz, Khurram.** [éd.] ACM. s.l. : ACM, 2009. Proceedings of the 7th International Conference on Frontiers of Information Technology. 978-1-60558-642-7/09/12.

74. *Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations.* **Scarani, Valerio, et al.** 05, 2004, Physical review letters, Vol. 92, p. 057901.

75. *Cryptography from Quantum Mechanical Viewpoint.* **Minal, Lopes et Nisha, Sarwade.** 2, Juin 2014, International Journal on Cryptography and Information Security, Vol. 4, pp. 13-25. DOI: 10.5121/ijcis.2014.4202.

76. *Quantum key distribution in the classical authenticated key exchange framework.* **Mosca, Michele, Stebila, Douglas et Berkan, Ustaoglu.** [éd.] Springer Berlin Heidelberg. 2013. In International Workshop on Post-Quantum Cryptography. pp. 136-154.

77. *Simple Proof of Security of the BB84 Quantum Key Distribution Protocol.* **Peter W, Shor et Preskill, John.** 2, 2000, Physical review letters, Vol. 85, p. 441.

78. *Automated Verification of Quantum Protocols by Equivalence Checking.* **Ardeshir-Larijani, Ebrahim, Gay, Simon J. et Nagarajan, Rajagopal.** 2013. arXiv preprint arXiv:1312.5951.

79. **Bjesse, Per.** What is formal verification? [éd.] ACM. *ACM SIGDA Newsletter.* 15 Decembre 2005, Vol. 35, 24.

80. **Parker, Dave.** PRISM-People. *www.prismmodelchecker.org.* [En ligne] [Citation : 15 Septembre 2016.] http://www.prismmodelchecker.org/people.php.

81. —. The PRISM Language: Introduction. *http://www.prismmodelchecker.org.* [En ligne] 03 Decembre 2010. [Citation : 27 Septembre 2016.] http://www.prismmodelchecker.org/manual/ThePRISMLanguage/Introduction.

82. **Patrick.** 8 messages codés qui restent à déchiffrer. *http://www.axolot.info/.* [En ligne] 11 Septembre 2011. [Citation : 09 Obtobre 2016.] http://www.axolot.info/?p=1092&cpage=1.

83. **Peter, Wayner.** *Disappering cryptograpgy:Information hiding: steganogray and watermarking.* Burlington : Morgan Kaufman, 2009. 978-0-12-374479-1.

84. **Djellab, Rima.** Cryptographie quantique. Batna, Algérie : s.n., 2009. Mémoire de Magistère.

85. **Drechsler, Rolf.** *Advanced Formal Verification.* s.l. : KLUWER ACADEMIC PUBLISHERS, 2004. 1-4020-7721-1.

86. *Study on Secret Sharing Schemes (SSS) and Their applications.* **Al Ebri, Noura, Baek, Joonsang et Yeob, Yeun Chan.** Abu Dhabi : s.n., 2011. 6th International conference on internet technology and secured transactions. pp. 40-45. 978-1-908320-00-1/11.

87. *Quantum Cryptography: Uncertainty in the Service of privacy.* **Bennett, Charles H.** 1992, Science, Vol. 257, pp. 752-3. ISSN 0036-8075.

88. *Communicating quantum processes.* **J. Gay, Simon et Nagarajan, Rajagopal.** 2004. QPL. pp. 91-107.

89. *PRISM: Propabilistic Symbolic Model Checker.* **Marta, Kwaitkoska, Gethin, Norman et David, Parker.** [éd.] P. Harrison, J, Bradley and U. Harder T.Field. Berlin : Springer, 2002. Computer Performance Evaluation TOOLS 2002. Vol. 2324, pp. 200-204.

90. *On the Performance of Group Key Agreement.* **Yair, Amir, Yongdae, Kim et Tsudik, Gene.** 3, New York : ACM, August 2004, ACM Transaction on Information and System Security, Vol. 7, pp. 457-488. Doi 10.1145/1015040.1015045.

91. **Dave, Parker, Gethin, Norman et Marta, Kwiatkowska.** Introduction. *www.prismmodelchecker.org.* [En ligne] [Citation : 31 Mai 2016.] http://www.prismmodelchecker.org/manual/ThePRISMLanguage/Introduction.

92. —. Manual. *www.prismmodelchecker.org.* [En ligne] [Citation : 13 October 2015.] http://www.prismmodelchecker.org/manual.

93. *A Method for Obtaining Digital Signatures and Public-key Cryptosystems.* **Ronald, Rivest, Adi, Shamir et Adleman, Leonard.** 2, 1978, Communications of the ACM, Vol. 21, pp. 120-126.

94. *Hybrid encryption/decryption technique using new public key and symmetric key algorithm.* **Prakash, Kuppuswamy et Saeed, Q.Y. Al-Khalidi.** [éd.] Inderscience. 4, 2016, International Journal of Information and Computer Security, Vol. 6, pp. 372 - 382. DOI: http://dx.doi.org/10.1504/IJICS.2014.068103.

95. les 7 merveilles de la mecanique quantique. *sciencetonnante.wordpress.com.* [En ligne] https://sciencetonnante.wordpress.com/2013/09/30/les-7-merveilles-de-la-mecanique-quantique/.

Résumé

Quelque soit l'environnement dans lequel se déroule une communication, la sécurité reste un pilier nécessaire pour le bon déroulement de cette communication. Afin d'assurer un certain niveau de sécurité, la confidentialité est un élément essentiel à assurer. Ce sont les techniques de cryptographie qui l'assurent. La pierre angulaire des techniques cryptographiques est la gestion de clé. Plusieurs protocoles ont été proposés pour assurer une bonne gestion de clé dans un groupe, mais leur sécurité se base sur la complexité computationnelle. La distribution de clé quantique, par abus dite cryptographie quantique, est une primitive cryptographique basée sur les lois de la mécanique quantique, permettant d'établir une clé secrète commune entre les traditionnels correspondants Alice et Bob. Dans cette thèse, nous nous intéressons à la problématique de la distribution de clé dans groupe. Nous proposons pour cela une solution basée sur la distribution de clé quantique, ce qui permet d'exploiter cette même solution dans le contexte de groupe. Nous avançons par la suite la preuve formelle de la solution proposée en utilisant le vérificateur formel PRISM.

Mots clé :

Cryptographie, Cryptographie quantique, BB84, Distribution de clé, Distribution de clé quantique, Quantique, Sécurité.

Abstract

Whatever the environment in which communication takes place, security remains a necessary pillar for the success of this communication. To ensure a certain level of security, confidentiality is essential to ensure. These are the cryptographic techniques that provide it. The cornerstone of cryptographic techniques is the key management. Several protocols have been proposed to ensure good key management in a band, but their security is based on computational complexity. The quantum key distribution, by abuse called quantum cryptography, is a cryptographic primitive based on the laws of quantum mechanics, to establish a shared secret key between the traditional corresponding Alice and Bob. In this thesis, we address the problem of group key distribution. For this we propose a solution based on quantum key distribution, which allows the use of this solution in the group context. We argue later formal proof of the proposed solution using the PRISM formal model checker.

Key Words:

Cryptography, BB84, Key distribution, Quantum cryptography, Quantum key distribution, Quantum, Security.