Un schéma d'authentification pour les protocoles de Distribution Quantique de Clef

Minh-Dung DANG (1,2)

***Abstract*—L'échange de clef entre les utilisateurs légitimes, employant les protocoles de *Distribution Quantique de Clef* (QKD - "Quantum Key Distribution"), est parfaitement sécurisé grâce aux propriétés de la physique quantique. Cependant, les protocoles de QKD purs ne garantissent pas l'authentification et sont donc vulnérables contre les *attaques par le milieu*. Il faut éliminer cette intrusion par les schémas d'authentification. Pour atteindre une sécurité élevée, il a été proposé d'implémenter des schémas d'authentification utilisant les *clefs jetables* ("one-time key"). Malheureusement, les schémas de ce type se prêtent aux attaques de *déni de service* (DoS - "Denial of Service") : l'espion peut épuiser entièrement le stock des clefs prépositionnées en perturbant la transmission.**

Dans cet article, nous proposons un schéma d'authentification pour les protocoles de QKD, en exposant le cas de BB84. Notre schéma utilise aussi les clefs secrètes, mais ne respecte pas le principe des clefs jetables et serait sécurisé contre les attaques de DoS.

***Index Terms*— quantum cryptography, quantum key distribution, identity verification, authentication.**

I. Introduction

La *distribution de clef* et la *mise en accord de clef* sont les protocoles importants de la Cryptographie. On en a besoin afin d'échanger des secrets nécessaires entre les utilisateurs dans des protocoles cryptographiques symétriques comme le chiffrement ou l'authentification.

En résolvant ces problèmes, les cryptograpes modernes, dont Diffie-Hellman sont les pionniers, ont donné lieu à des systèmes cryptographiques *à clef publique*. En principe, nous pouvons échanger des clefs secrètes en utilisant les systèmes à clef publique. Pourtant, ces protocoles se basent sur les hypothèses non-prouvées comme la complexité de la factorisation, des logarithmes discrètes, ... Leur sécurité était récemment mise en question, notamment dès que Shor a trouvé un bon algorithme de factorisation des entiers pour les ordinateurs quantiques.

La Cryptographie Quantique a fait partie du domaine en soumettant des versions quantiques de mise en accord de clef, nommées par un nom incorrect mais maintenant familier à toute personne dans le domaine : la Distribution Quantique de Clef (QKD). Incomparable aux protocoles d'échange de clef classiques, les protocoles quantiques sont basés sur les propiétés imbattables de la physique quantique et ont été prouvés inconditionnellement sûrs [1].

Malheureusement, les protocoles de QKD ne fournissent pas eux-mêmes l'authentification, et sont par conséquent facilement cassés par une menace naïve : les attaques par le milieu. Pour empêcher cette intrusion, certains schémas d'authentification classiques [2] aussi bien que quantiques [3] ont été proposés pour protéger les protocoles de QKD.

Comme la sécurité inconditionnelle est le but primordial de la cryptographie quantique, l'approche à clef jetable est forcément utilisée. Cela exige évidemment certaines clefs secrètes prépositionnées entre les utilisateurs. Pourtant, les clefs jetables permettent les attaques de déni de service : *comme chaque clef est utilisée une seule fois, l'espion peut perturber la communication pour épuiser toutes les clefs d'authentification qui ont été coûteusement échangées.*

Dans cet article, nous proposons un schéma d'authentification pour les protocoles de QKD. Ce schéma permet d'utiliser une clef secrète pour plusieurs fois, et pourrait en conséquence être résistant aux attaques de DoS. La section II aborde le contexte qui aboutit à notre travail. L'analyse de la sécurité des clefs sur les langages non-redondants est exposée dans la section III. Et dans la section IV, nous présentons notre contribution, exploitant les résultats de la section III, à la construction de notre schéma d'authentification.

II. Préliminaires

A. Cryptosystèmes à clef publique et à clef secrète

Les cryptosystèmes de la cryptographie moderne sont divisés en deux catégories : ceux à clef publique et ceux à clef secrète.

1) Les systèmes à clef publique: Nommés aussi *systèmes asymétriques.* Chaque utilisateur est associé à une paire de clefs (k_{pri}, k_{pub}) : k_{pri} est la clef privée à l'utilisateur, et la clef k_{pub} est publiée.

Un expéditeur, nommé Alice, veut envoyer un message confidentiel à un destinataire, nommé Bob. Alice va chercher la clef publique k_{pub} de Bob ; chiffre le message avec la clef ; et envoie à Bob qui peut faire le déchiffrement avec sa clef privé k_{pri} pour récupérer le message.

L'avantage principal des systèmes à clef publique est qu'ils ne demandent pas une distribution sécurisée des clefs. Cependant, il est exigé que les clefs publiques soient authentifiées, normalement certifiées par une partie honnête et connue, pour empêcher les attaques par le milieu : un espion, nommé Eve, crée une paire de clefs k'_{pri}, k'_{pub} ; publie k'_{pub} sous le nom de Bob, et peut découvrir les messages envoyés à celui-ci.

La possibilité d'identifier chaque utilisateur rend aux systèmes à clef publique un autre usage : la signature numérique. Un utilisateur peut créer une signature sur un message en applicant un calcul au message et à sa clef privée. Tout le monde peut vérifier la signature avec sa clef publique. Comme chaque

(1) *GET-ENST, 46 rue Barrault, 75634 Paris Cedex 13, France.*
(2) *AUF-IFI, 42 pho Ta Quang Buu, Hanoi, Vietnam.*
Courriel : dang@infres.enst.fr

paire de clefs est assignée à une personne unique, l'émetteur ne peut pas refuser le fait qu'il a envoyé le message [4].

Par ailleurs, la sécurité des systèmes à clef publique est basée sur des hypothèses non-prouvées du domaine de la complexité de calcul. Certains problèmes, proclamés difficiles comme la factorisation des entiers (système RSA), les logarithmes discrets (système ElGamal), les courbes elliptiques (système des Courbes Elliptiques), ... sont utilisés pour construire des systèmes à clef publique [4]. Ces derniers sont potentiellement rompus par des avances mathématiques, de nouveaux algorithmes et nouvelles technologies de calcul tel que les ordinateurs quantiques.

2) Les systèmes à clef secrète: Nommés aussi *systèmes symétriques.* Un clef secrète est partagée entre Alice et Bob qui s'entendent sur deux algorithmes de chiffrement et de déchiffrement. Pour envoyer un message, Alice utilise l'algorithme de chiffrement qui combine le message avec la clef ; et à la réception, Bob utilise l'algorithme de déchiffrement qui extrait le message original à partir du message chiffré.

Alors que les systèmes à clef publique se basent sur les hypothèses de la complexité de calcul, ceux à clef secrète utilisent les résultats de la théorie de l'information. Grâce à celle-ci, certains systèmes sont reconnus sécurisés contre les ennemis ayant une capacité de calcul illimitée, par exemple le chiffrement de Vernam que nous allons mentionner plus tard.

Pourtant, les systèmes à clef secrète exigent qu'Alice et Bob partagent discrètement des clefs. C'est en effet le rôle des protocoles d'échange de clef.

B. Sécurité inconditionnelle

La notion de la sécurité inconditionnelle contre les ennemis ayant une capacité de calcul illimitée était premièrement étudiée par Shannon [5], [4], qui utilisait la définition de l'entropie. Si nous avons un variable aléatoire X qui prend les valeurs $\{x_1, x2, \ldots, x_n\}$ avec les probabilités $\{p_1, p_2, \ldots, p_n\}$ respectivement, son entropie est définie par

$$H(X) = -\sum_{i=1}^{n} p_i * log(p_i) \qquad (II.1)$$

Cette somme atteint la value maximale quand X prend les valeurs $x_1, ..., x_n$ avec la même probabilité, ie. $p_1 = p_2 = ... = p_n$, et

$$H(X)_{max} = log|X| = log(n) \qquad (II.2)$$

L'entropie conditionnelle, *qui mesure la quantité moyenne d'information d'un variable aléatoire X, révélée par un autre variable Y*, s'écrit

$$H(X/Y) = \sum_{x \in X, y \in Y} p(x,y) * log(p(x/y)) \qquad (II.3)$$

Normalement, un cryptosystème est caractérisé par les trois variables

- M : ensemble des messages en clair (plaintexts).
- K : ensemble des clefs.
- C : ensemble des messages chiffrés (ciphertexts).

Un système est dit inconditionnellement sécurisé si, après avoir intercepté le message chiffré, l'espion n'a pas davantage d'information sur le message en clair correspondant. Ou, autrement dit, la probabilité *a posteriori* que le message en clair prend une valeur m, étant donné que le message chiffré soit connu, est égale à celle à celle *a priori* que le message en clair prend la valeur m:

$$\forall m \in M, \forall c \in C, p(m/c) = p(m) \qquad (II.4)$$

On peut en déduire que

$$H(M/C) = H(M) \qquad (II.5)$$

La théorie de Shannon a une remarque importante : si un cryptosystème (M, K, C) est inconditionnellement sécurisé, alors

$$|K| \geq |C| \geq |M| \qquad (II.6)$$

Cela veut dire qu'un système parfaitement sécurisé consomme au moins un nombre de clefs aussi grand que celui des messages. Dans le cas optimal, $|K| = |C| = |M|$, nous avons le chiffrement de Vernam:

> Pour communiquer chaque message m, Alice et Bob utilisent une séquence de bits aléatoires - k - de la même longueur comme clef secrète. Alice combine les deux, produisant un message chiffré $c_i = m_i \oplus k_i$ où $\oplus$ est l'addition modulo 2. Bob extrait le message par le même algorithme $m_i = c_i \oplus k_i$. Chaque clef k est utilisée une seule fois.

Nous insistons que les clefs doivent être utilisées une seule fois. Si Alice et Bob utilisent une même clef pour plusieurs messages, Eve peut se baser sur les statiques du langage L des messages en claire M pour découvrir la clef. Nous faisons ici un résumé des études de Shannon sur la révélation statistique de la clef, appelée *"key equivocation"*.

L'entropie du langage L est définie par l'entropie moyenne dans toute combinaison des messages :

$$H_L = lim_{n \to \infty} \frac{H(M^n)}{n} \qquad (II.7)$$

qui sera égale à celle de M dans le cas où L est un langage aléatoire, et $H(L) = H(M) = log|M|$. La redondance du langage est donc :

$$R_L = 1 - \frac{H_L}{log|M|} \qquad (II.8)$$

Étant donné un message chiffré y de n messages en claire par une même clef i.e. $y \in C^n$, nous définissons l'ensemble des clefs possibles pour produire y:

$$K(y) = \{k \in K : \exists x \in M^n, e_k(x) = y\} \qquad (II.9)$$

où e_k est la fonction de chiffrement associée à la clef k. Parmi ces clefs, il y a la seule clef k est bonne. Le nombre de fausses clefs est donc $K(y) - 1$. Et le nombre moyen de fausses clefs pour tous les messages chiffrés $y \in C^n$ est

$$\overline{s}_n = \sum_{y \in C^n} p(y)(|K(y)| - 1) \qquad (II.10)$$

$$= \left(\sum_{y \in C^n} p(y)|K(y)| \right) - 1 \qquad (II.11)$$

Quand $|C| = |M|$ et les clefs sont choisies avec une même probabilité, ie. $H(K) = log|K|$, on a prouvé que :

$$\overline{s}_n \geq \frac{|K|}{|M|^{nR_L}} - 1 \qquad (II.12)$$

Dans le cas où $R_L > 0$, il y a une valeur de n, appelé *"unicity distance"* et dénoté n_0, qui rend le membre droit de l'inéquation $II.12$ à zéro. On a :

$$n_0 \approx \frac{log|K|}{R_L log|M|} \qquad (II.13)$$

Eve a alors une chance pour réduire le nombre de fausses clefs à zéro après avoir intercepté plus de n_0 messages chiffrés avec une même clef.

C. Distribution Quantique de Clef

La première idée d'utiliser les états quantiques pour discrètement transmettre de l'information est à Wiesner [6], exploitant les principes de l'incertitude de la physique quantique :

- *On ne peut reconnaitre un système quantique que par les mesures; et une mesure incompatible va perturber fondamentalement le système en le transformant dans un des états propres de la mesure.*
- *On ne peut pas cloner un système quantique sauf quand il est dans un état bien déterminé.*

Inspiré des idées de Wiesner, Bennet et Brassard ont créé les premiers protocoles cryptographiques quantiques : *l'échange de clef* et *le tirage à pile ou face à distance* [7], parmi lesquels le protocole d'échange de clef, nommé BB84, a été prouvé sécurisé contre toutes les attaques permises par les lois quantiques [1], [8].

L'information quantique utilise les systèmes à deux états, représentés par un espace Hilbert à deux dimensions, pour coder les bits quantiques - *qubits* [9]. Conformément à une base orthorgonale $\{|0\rangle, |1\rangle\}$ choisie, un bit qubit est représenté par un verteur unitaire $a|0\rangle + b|1\rangle$ où $|a|^2 + |b|^2 = 1$. Une observable est caractisée par une matrice, décomposable en une somme des projections sur ses états propres $|v_i\rangle$: $A = \sum_i a_i |v_i\rangle\langle v_i|$, où a_i sont des valeurs propres correspondantes, et nécessairement réelles. La mesure de A d'un système dans l'état $|\psi\rangle$ donne a_i comme résultat avec la probabilité $|\langle v_i|\psi\rangle|^2$. Et après la mesure qui donne a_i comme résultat, le système sortant est dans un état déterminé - l'état propre $|v_i\rangle$ normé. Il est ainsi profondément perturbé par la la mesure de A sauf que $|\psi\rangle$ soit un état propre de A. Par exemple la mesure de l'observable $\sigma = (|0\rangle\langle 0| - |1\rangle\langle 1|)$ d'un système dans l'état $a|0\rangle + b|1\rangle$ produit $+1$, sortant $|0\rangle$ avec la probabilité $|a|^2$; et produit -1, sortant $|1\rangle$ avec la probabilité $|b|^2$.

Les bits classiques $0, 1$ peuvent être codés par les qubits orthogonaux - états propres d'une observable. Par exemple les états $\{|0\rangle, |1\rangle\}$ sont distinguables par l'observable $\sigma = (|0\rangle\langle 0| - |1\rangle\langle 1|)$. Il est important qu'il est possible d'utiliser une base orthogonale B quelconque, par exemple $\{a|0\rangle + b|1\rangle, b|0\rangle - a|1\rangle\}$ avec a, b réels. Dans ce cas là, on peut dénoter $|0_B\rangle = a|0\rangle + b|1\rangle, |1_B\rangle = b|0\rangle - a|1\rangle$, et la nouvelle observable σ de cette base s'écrit $\sigma_B = (|0_B\rangle\langle 0_B| - |1_B\rangle\langle 1_B|)$.

Le protocole BB84 utilise quatre états quantiques pour coder les informations - deux paires d'états propres de deux bases conjuguées. Il est ainsi nommé par un autre nom - *QKD à 4 états*. Deux bases X et Y sont dites conjuguées si la mesure de l'observable σ_Y d'un système dans un des états propres de X - $|0_X\rangle, |1_X\rangle$ - produit $+1$, sortant $|0_Y\rangle$, ou -1, sortant $|1_Y\rangle$ avec la même probabilité ; et réciproquement.

En réalité, la polarisation des photons peut être utilisée pour coder les qubits. Deux polarisations orthogonales, par exemple horisontale (0^o) et verticale (90^o) dans la base rectangulaire, représentent deux qubits $|0\rangle$ et $|1\rangle$, et un qubit $a|0\rangle + b|1\rangle$ (a, b réels) est représenté par la polarisation d'un angle α où $cos(\alpha) = a$. Deux bases conjuguées sont utilisées dans BB84 : la base rectangulaire ($\bigoplus$) et et la base diagonale ($\bigotimes$). La base rectangulaire est composée de deux états : la polarisation de 0^o et celle de 90^o ; la base diagonale est composée de la polarisation de 45^o et de celle de 135^o.

Le protocole utilise un canal quantique pour envoyer les états quantiques et un canal classique pour les communications traditionnelles. On peut décrire le protocole comme dans le schéma suivant :

Protocole BB84

1) L'émetteur, Alice, prépare une séquence de bits aléatoires et code chaque bit par l'état propre correspondant dans une base tirée aléatoirement parmi $\{\bigoplus, \bigotimes\}$ - $x \in \{|0_{\bigoplus}\rangle, |1_{\bigoplus}\rangle, |0_{\bigotimes}\rangle, |1_{\bigotimes}\rangle\}$.
2) Alice envoie ces états à Bob par le canal quantique.
3) Bob utilise une base, aléatoirement tirée de $\{\bigoplus, \bigotimes\}$ pour mesurer chaque état arrivé, et produit une séquence de bits aléatoires. Il annonce la séquence des bases utilisées à Alice.
4) Alice annonce les bases non-assorties à Bob et ils jettent les bits correspondants. Alice et Bob possèdent maintenant deux séquences de bits x_a, x_b qui seraient identiques s'il n'y avait pas d'erreurs, et ils peuvent les utiliser comme clef secrète pour envoyer les informations confidentielles.
5) Alice et Bob comparent une portion de bits, extraits de x_a, x_b, pour détecter Eve. Le nombre de bits comparés doit être suffisamment grand pour assurer que les erreurs causées par présence d'Eve soient détectables. S'il n'y a pas d'erreurs, ou x_a, x_b sont "probablement" identiques, ils peuvent les utiliser comme clef secrète pour envoyer les informations confidentielles. S'il y a des erreurs, ils jettent x_a, x_b et réessaient une autre session.

Si un espion, Eve, veut écouter la transmission, elle doit mesurer les états quantiques avec les bases forcément aléatoires car elle ne connnait pas les bases utilisées par Alice et Bob. Elle risque donc de laisser les erreurs détectables par le dernier étape du protocole. Les calculs statistiques montrent qu'avec chaque mesure aléatoire, Eve a une probabilité $1/4$ de changer le résultat de Bob, et donc de laisser un bit erroné.

En pratique, faute à l'imperfection des matériels, il y a toujours des erreurs durant la transmission quantique. Alice et Bob doivent tolérer un taux d'erreurs dans l'étape de détection de présence de Eve. Ils doivent implémenter un protocole de cor-

rection - *"bit reconciliation"* - pour corriger des bits erronés, et un mécanisme d'amplification de sécurité - *"privacy amplification"* pour élimier les connaissances acquises par Eve sur la clef [10].

Les protocoles de QKD reçoivent encore davantage de contributions des cryptographes ainsi que des physiciens. Il faut mentionner le protocole à deux états, inventé par Bennett [11], utilisant deux états non-orthogonaux au lieu de quatre dans BB84. Une autre famille venait de l'idée de Ekert, utilisant des photons intriqués [12] : deux photons dans un état intriqué - état de Bell - sont séparés et envoyés à Alice et Bob ; comme la mesure locale de chaque utilisateur affecte la mesure de l'autre dans le cas ils utilsent la même base de mesure, ils peuvent en profiter pour échanger des clefs secrètes. Les premiers appareils implémentés de QKD ont été élaborés en 1991, et pouvaient transférer des clef au travers une distance de $32cm$ [13], [14]. Aujourd'hui, on a déjà des dispositifs à franchir d'une centaine de kilomètres, et prêts à être utilisés [15], [16]. Quelques projets ont été aussi lancés pour la conception des réseaux quantiques à grande échelle [17], [18].

D. Authentification pour les protocoles de QKD

Les protocoles de QKD sont inconditionnellement sécurisés mais ne garantissent pas l'authentification. L'espionne Eve peut toujours se mettre au milieu, se présente à Alice sous le nom de Bob et à Bob sous le nom d'Alice pour établir deux clefs, k_a avec Alice et k_b avec Bob. Elle peut donc déchiffrer les messages envoyés par Alice avec k_a ; les lire ; les chiffrer avec k_b et renvoyer à Bob, et faire l'inverse avec les messages envoyés par Bob sans être détectée par ces pauvres.

C'est pourquoi il faut intégrer un mécanisme d'authentification pour empêcher cet intrus.

1) Authentification classique: Une idée simple est d'implémenter un schéma d'authentification classique audessus des protocoles de QKD [2]. Après l'exécution de QKD pour échanger une clef, on considère cette clef comme un message à authentifier, et on peut alors appliquer les techniques d'authentification de messages. Cela est fait par la création d'une étiquette d'authentification - MAC ("Message Authentification Code"), qui sera vérifiable grâce aux secrets : clef publique où clef publique.

Il y a toujours deux familles : celle à clef publique et celle à clef secrète.

- *Authentification à clef publique* : L'authentification à clef publique est réalisé par la signature numérique. Elle hérite également de la vulnérabilité potentielle des systèmes à clef publique, cf. section II-A.1.
- *Authentification à clef secrète* : Cette approche utilise normalement des fonctions de hachage sécurisé, utilisant une clef secrète. Elle demande évidemment des clefs prépositionnées entre les utilisateurs.

Pour assurer une sécurité élevée, on préfère les protocoles d'authentification à clef secrète. Par inconvénient, outre la construction des fonctions de hachage inconditionnellement sécurisées [19], les protocoles à clef secrète hautement sécurisés exigent que les clefs doivent être utilisées une seule fois (c.f. section II-B). Cette contrainte des protocoles à clef jetable permet les attaques de DoS : Eve perturbe le canal pendant une durée nécessaire pour épuiser entièrement le stock des clefs d'authentification.

2) Authentification quantique: Certains schémas quantiques sont aussi proposés pour aider à vérifier l'identité des participants dans les protocoles de QKD [3]. Ces schémas utilisent les clefs secrètes pour introduire des informations d'authentification dans les communications des protocoles de QKD. Pour assurer leur sécurité, ces schémas respectent aussi le principe des clefs jetables sont ainsi vulnérables contre les attaques de DoS.

III. SÉCURITÉ DE CLEF AVEC LES MESSAGES EN CLAIR ALÉATOIRES

Rappelons-nous la révélation statistique de clef, abordée dans la section II-B : comme les cryptosystèmes marchent normalement sur les langages naturels dont la redondance $R_L > 0$, si on utilise une clef pour plusieurs messages, Eve a la possibilité de découvrir la clef après avoir intercepté plus de n_0 messages chiffrés. n_0 est calculé par l'équation $II.13$.

Ce n'est pas le cas si on l'applique aux messages aléatoires :

$$H_L = \lim_{n\to\infty} \frac{H(M^n)}{n} = H(M) = log|M| \qquad (III.1)$$

Maintenant, le langage est non-redondant

$$R_L = 1 - \frac{H_L}{log|M|} = 0 \qquad (III.2)$$

et le nombre de fausses clefs ne réduit pas quand le nombre de messages chiffrés interceptés augmente :

$$\overline{s}_n \geq \frac{|K|}{|M|^{nR_L}} - 1 = |K| - 1 \qquad (III.3)$$

Ainsi, avec un langage non-redondant, une même clef peut être utilisée avec un nombre infini de messages sans être révélée.

Cela aboutit à une remarque intéressante

Considérons l'ensemble M des messages aléatoires de longueur $n * l$, nous avons :

$$H(L) = H(M) = n * l$$

Étant donné une clef aléatoire k de longueur l et un message $m \in M$, nous pouvons couper m en segments de l bits et les chiffrer avec le même clef k sans révéler aucune information sur la clef.

$$H(K/C) = H(K) = l$$

Pourtant, le message m n'est pas sécurisé :

$$\begin{aligned}
H(M/C) &= H(M) + H(C/M) - H(C) \text{ (voir [5])} \\
&= H(C/M) \text{ (comme } H(M) = H(C) = n * l\text{)} \\
&= -\sum_{m\in M, c\in C} p(m,c) * log(p(c/m)) \\
&= -\sum_{m\in M, k\in K} p(m,k) * log(p(k/m)) \\
&= -\sum_{m\in M, k\in K} p(m) * p(k) * log(p(k)) \\
&= -\sum_{k\in K} p(k) * log(p(k)) * (\sum_{m\in M} p(m)) \\
&= -\sum_{k\in K} p(k) * log(p(k)) = H(K) = l \\
&< H(M) = n * l
\end{aligned}$$

IV. Un schéma d'authentification pour QKD

Comme les bases utilisées par Alice et Bob sont aléatores, nous pouvons profiter des propriétés trouvées, cf. la section III, pour introduire les étiquettes d'authentification dans les protocoles de QKD.

Nous écrivons dès maintenant $k \oplus m$ pour dénoter le chiffrement d'un message long m en utilisant la concaténation de n copies de k où $|m| = n * |k|$.

Supposons que Alice et Bob partage une séquence de bits aléatoires de longueur k comme clef secrète, nous décrivons ici un protocole de QKD, basé sur BB84, authentifié par la clef.

A. Première version

1) Protocole:

Schéma d'authentification simple

1) L'émetteur, Alice, prépare une séquence de bits aléatoires et code chaque bit par l'état propre correspondant dans une base tirée aléatoirement parmi $\{\bigoplus, \bigotimes\}$ - $x \in \{|0_{\bigoplus}\rangle, |1_{\bigoplus}\rangle, |0_{\bigotimes}\rangle, |1_{\bigotimes}\rangle\}$.
2) Alice envoie ces états à Bob par le canal quantique.
3) Bob utilise une base, aléatoirement tirée de $\{\bigoplus, \bigotimes\}$ pour mesurer chaque état arrivé, et produit une séquence de bits aléatoires. Il annonce la séquence des bases chiffrées - $b_b \oplus k$, à Alice.
4) Alice déchiffre le message pour découvrir les bases utilisées par Bob.
5) Alice annonce les bases non-assorties à Bob et ils jettent les bits correspondants. Alice et Bob possèdent maintenant deux séquences de bits x_a, x_b qui seraient identiques s'il n'y avait pas d'erreurs, et ils peuvent les utiliser comme clef secrète pour envoyer les informations confidentielles.
6) Alice et Bob comparent une portion de bits, extraits de x_a, x_b, pour détecter Eve. Le nombre de bits comparés doit être suffisamment grand pour assurer que les erreurs causées par présence d'Eve soient détectables. S'il n'y a pas d'erreurs, ou x_a, x_b sont "probablement" identiques, ils peuvent les utiliser comme clef secrète pour envoyer les informations confidentielles. S'il y a des erreurs, ils jettent x_a, x_b et réessaient une autre session.

2) Analyse de sécurité: On réalise que la clef échangée est inconditionnellement sécurisée car ce protocole n'est pas moins sécurisé que BB84, cf. II-C.

Si Eve veut réaliser une attaque par le milieu, par exemple elle remplace Bob pour tricher avec Alice. Elle doit envoyer, dans l'étape 3, ses bases b_e chiffrées par une chaîne quelconque k_e, et envoyer $b_e \oplus k_e$ à Alice. Comme k_e n'a aucune importance, on peut choisir une chaîne de bit 1 et Alice reçoit donc b_e. Alice considère donc $b_b = b_e \oplus k$, et annonce les bases non-assorties i.e $a = b_a \oplus b_b = b_a \oplus b_e \oplus k$ à Eve. Alice garde les bits aux positions i où $a[i] = 0$, et alors soit k une chaîne aléatoire, $b_a[i] \neq b_e[i]$ avec la probabilité $1/2$. Les bits des résultats d'Alice et d'Eve - x_a, x_e - se diffèrent donc avec une probabilité de $1/4$. Ce taux d'erreur serait détecté par la dernière étape.

L'observation passive d'Eve sur le canal classique ne lui révèle pas d'informations sur la clef d'authentification k - cf. III. Pourtant, Eve peut faire plus que cela, en réalisant des attaques de déni de service, pour découvrir k. En fait, les positions où Alice garde les bits codés sont annoncées. Ensuite, Alice publie certains bits de résultat pour détecter les erreurs. Alors si Eve fait la mesure de quelques états quantiques, aux positions i, elle a une chance de découvrir les bases d'Alice (Bob). En effet, si une mesure de l'état i cause une erreur, détectée par Alice (Bob), Eve est révélé que sa base est fausse et $b_a[i] = \overline{b_e}[i]$ et peut calculer $k[i] = a[i]$. Alors, si Eve intercepte l états aux positions $i_1, ..., i_l$, sur lesquels Alice garde en moyen $l/2$ résultats, Alice annonce $p * \frac{l}{2}$ bits de résultats où p est la portion de bits annoncés pour la détection d'erreur. *Il y a donc en moyenne $\frac{1}{4} * p * \frac{l}{2}$ bits erronés, grâce auxquels Eve peut arriver à découvrir les bits correspondant de la clef k.*

On contaste que ce simple protocole peut nous aider à authentifier Alice et Bob. Il est seulement sécurisé si l'on utilise les clefs jetables, et donc se prête aux attaques de déni de service. Si on veut utiliser une même clef plusieurs fois, Eve peut répéter le schéma ci-dessus pour découvrir la clef.

Or, nous pouvons éliminier l'attaque ci-dessus en cachant en plus les positions où Alice et Bob gardent les résultats. Cela aboutit à un schéma qui permet d'utiliser une clef d'authentification pour plusieurs fois.

B. Le schéma final

1) Protocole: Alice et Bob divise la clef partagée initiale k en deux clefs secrète k_a, k_b de même longueurs.

Schéma d'authentification final

1) L'émetteur, Alice, prépare une séquence de bits aléatoires et code chaque bit par l'état propre correspondant dans une base tirée aléatoirement parmi $\{\bigoplus, \bigotimes\}$ - $x \in \{|0_{\bigoplus}\rangle, |1_{\bigoplus}\rangle, |0_{\bigotimes}\rangle, |1_{\bigotimes}\rangle\}$.
2) Alice envoie ces états à Bob par le canal quantique.
3) Bob utilise une base, aléatoirement tirée de $\{\bigoplus, \bigotimes\}$ pour mesurer chaque état arrivé, et produit une séquence de bits aléatoires. Il annonce la séquence des bases chiffrées - $b_b \oplus k_b$, à Alice.
4) Alice envoie la séquence des bases de préparation, chiffrée - $b_a \oplus k_a$, à Bob.
5) Alice et Bob déchiffrent les messages et trouvent les bases non-assorties et ils jettent les bits correspondants. Alice et Bob possèdent maintenant deux séquences de bits x_a, x_b qui seraient identiques s'il n'y avait pas d'erreurs.
6) Alice et Bob comparent une portion de bits, extraits de x_a, x_b, pour détecter Eve. Le nombre de bits comparés doit être suffisamment grand pour assurer que les erreurs causées par présence d'Eve soient détectables. S'il n'y a pas d'erreurs, ou x_a, x_b sont "probablement" identiques, ils peuvent les utiliser comme clef secrète pour envoyer les informations confidentielles. S'il y a des erreurs, ils jettent x_a, x_b et réessaient une autre session.

2) Analyse de sécurité: La sécurité de la clef échangée est évidemment assurée contre toute observation. La clef d'authentification est aussi protégée quand Eve écoute le canal classique, cf. section IV-A.2.

Eve ne peut également pas réaliser les attaques par le milieu. En plus, on peut dire que ce schéma est plus fort que celui simple présenté dans la section IV-A. Par exemple nous étudions le cas où Eve remplace Bob pour tricher avec Alice. Elle doit envoyer, dans l'étape 3, ses bases b_e chiffrées par une chaîne quelconque k_e, et envoyer $b_e \oplus k_e$ à Alice. Comme k_e n'a aucune importance, on peut choisir une chaîne de bits 1 et Alice reçoit donc b_e. Alice envoie $b_a \oplus k_a$ à Eve. Alice considère donc $b_b = b_e \oplus k_b$, calcule les bases non-assorties $a = b_a \oplus b_b = b_a \oplus b_e \oplus k_b$ et garde les bits aux positions i où $a[i] = 0$. Connaissant $b_a \oplus k_a, b_e$, Eve ne peut pas trouver a, ie. $(b_a \oplus k_a) = a \oplus b_e \oplus (k_a \oplus k_b)$, ou autrement dit, a est chiffré par $k_a \oplus k_b$. Vu les études dans la section III, on a $H(a) = H(k_a \oplus k_b) = \frac{|k|}{2}$ où $|k|$ est la longueur de k. Eve devrait forcément choisir une chaîne aléatoire comme résultat x_e, par exemple en prenant la moitié des résultats aux positions aléatoires. x_a et x_e sont quasiment indépendants, même de longueurs différentes. Les bits extraits de x_a, x_e pour être comparés à la dernière étape diffèrent donc avec une probabilité de $1/2$. Ce taux d'erreur serait plus facilement détectable.

Si Eve veut découvrir la clef d'authentification, les résultats de mesure ne l'aident pas beaucoup car les positions où Alice et Bob gardent leurs résultats sont cachées, par $k_a \oplus k_b$. Quand Alice (Bob) annone un bit de x_a (x_b), Eve ne sait pas il est codé par quel état quantique, avec une incertitude $H(a) = \frac{|k|}{2}$, et ne peut donc pas découvrir la base correspondante pour déduire k_a (k_b).

Eve peut seulement faire des tests exhaustifs - réussite/échec - pour diminuer l'entropie de k_a, k_b. Elle peut

- Réaliser des attaques par le milieu en essayant toutes clefs possibles.
- Réaliser des attaques de type *interception/renvoi* : Eve intercepte quelques états quantiques avec les bases choisies par elle même. En se basant sur le fait que les mesures laissent des erreurs ou pas, Eve peut réduire leur entropie, et ainsi l'entropie des clefs d'authentification.

Ces deux attaques sont équivalentes et *faibles* : chaque test ne révèle pas beaucoup d'information sur les clefs d'authentification à Eve, ou autrement dit, leur entropie diminue faiblement. Eve doit ainsi faire un nombre énorme de tests, de l'ordre de $2^{|k|}$, pour découvrir entièrement les clefs k_a, k_b, ou une portion significative de leurs bits, pour passer l'authentification. D'ailleurs, chaque test doit être réalisé *en ligne*. Et par conséquent, *ces attaques ne dépendent pas de la capacité de calcul d'Eve.*

V. Comparaisons

Nous pouvons donc ajouter notre protocole dans la liste des schémas d'authentification pour QKD. La liste se compose donc de trois éléments principaux : le schéma classique utilisant les fonctions de hachage de Wegman-Carter [19], dénoté schéma 1 ; le schéma quantique de Zeng-Zhang [3], schéma 2 ; et notre schéma, schéma 3. Nous voudrions exposer des comparaisons entre ces schémas, prenant compte de quelques critères :

Implémentation
: On constate que l'implémentation du schéma 1 est plus compliquée, ayant besoin une classe des fonctions de hachage. Alors que les deux autres n'utilisent que de simples opérations de l'addition modulo 2.

Efficacité
: L'efficacité est mesurée par l'utilisation de la clef d'authentification. C'est facile de remarquer que le schéma 1 est le plus efficace. En effet, la différence d'un seul bit de la clef est détectable, cela implique qu'Eve doit connaitre la clef en entier pour pouvoir passer le protocole. Viens en deuxième rang le schéma 3 : l'attaque par le milieu laisse un taux d'erreurs de 50% contre 25% dans le schéma 2.

Sécurité
: En principe, les schémas $1, 2$ respectent le principe des clefs jetables et sont donc inconditionnelement sûrs, tandis que le schéma 3 ne l'est pas. Bien évidemment, le protocole 3 peut utiliser les clefs jetables pour atteindre la sécurité parfaite. Pourtant, les attaques de DoS suppriment cet avantage. Or, le protocole 2 ne peut pas utiliser une clef pour plusieurs sessions parce que la clef d'authentification est entièrement révélée à la fin de chaque session. Si l'on veut utilser une clef pour le schéma 1, on se prêtre aux attaques dont le message en clair est connu ("known-plaintext attack") : par exemple Eve communique avec Alice pour partager une clef m et Alice envoie ensuite un code d'authentification à Eve $c = h(m, k)$; Eve connait donc m et c. Eve peut réaliser des calculs *hors*

ligne avec sa capacité de calcul illimitée pour retrouver k. Par contre, le schéma 3 ne permet à Eve que de faire des tests *en ligne* qui ne dépendent pas de sa capacité de calcul.

VI. Conclusions

L'intention de communiquer de façon sécurisée entre les deux interlocuteurs qui ne se connaissent pas semble impossible. En fait, une troisième personne peut toujours se mettre au milieu pour tricher. Par conséquent, les deux parties légitimes doivent se connaître, et peuvent authentifier la conversation. Cela peut se réaliser quand les deux partagent une clef secrète, ou chaque partie a une valeur qui peut être publiquement reconnue - clef publique.

Les protocoles de QKD purs ne fournissent pas l'authentification. Ils n'identifient pas les utilisateurs. Il faut donc intégrer les mécanismes d'authentification pour les sécuriser, soit par l'approche à clef publique, soit par celle à clef secrète. Pour obtenir une sécurité inconditionnelle, on devrait utiliser les clefs secrètes en respectant la contrainte que chaque clef est utilisée une seule fois. Cela fait l'objet des attaques de DoS épuisant les clefs secrètes partagées. La contribution principale de cet article est de résoudre ce problème. La seule attaque possible sur notre protocole est une recherche exhaustive de clef, en ligne et interactive, qui ne dépend pas de la capacité de calcul de l'ennemi. En conséquence, cette contribution peut rendre les protocoles de QKD plus pratiques, et peut aider à des efforts de porter ces miracles au marché.

Pourtant, le travail n'est pas encore complet. La sécurité du protocole a été justifiée de façon heuristique et informelle. Il nous demande des preuves rigoureuses, justifiant une révélation d'information *suffisamment faible* sur la clef d'authentification durant les tests mentionnés dans la section IV-B.2, pour assurer la sécurité.

References

[1] D. Mayers, "Unconditional security in quantum cryptography," *Journal of the ACM*, vol. 48, no. 3, pp. 351 – 406, 2001.

[2] C. Elliott, D. Pearson, and G. Troxel, "Quantum cryptography in practice," in *Proceedings of ACM SIGCOMM*, 2003, pp. 227 – 238.

[3] G. Zeng and W. Zhang, "Identity verification in quantum key distribution," *Physical Review A*, vol. 61, issue 2, article 022303, 2000.

[4] D. Stinson, *Cryptography - Theory and Practice*, P. K.H. Rosen, Ed. CRC Press, 1995.

[5] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28-4, pp. 656–715, 1949.

[6] S. Wiesner, "Conjugate coding," *ACM SIGACT News*, vol. 15, no. 1, pp. 78 – 88, 1983.

[7] C. Bennet and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, 1984, pp. 175–179.

[8] P. Shor and J. Preskill, "Simple proof of security of the bb84 quantum key distribution protocol," *Physical Review Letters*, vol. 85, pp. 441 – 444, 2000, arXiv e-print quant-ph/0003004.

[9] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, C. U. Press, Ed. Cambridge Univ. Press, 2004.

[10] C. Bennett, F. Bessette, G. Bassard, L. Salvail, and J. Smolin, "Experimental quantum cryptography," *Journal of Cryptology*, vol. 5, no. 1, pp. 3 – 28, 1992.

[11] C. Bennett, "Quantum cryptography using any two nonorthogonal states," *Physical Review Letters*, vol. 68, no. 21, pp. 3121–3124, 1992.

[12] A. Ekert, "Quantum cryptography based on bell's theorem," *Physical Review Letters*, vol. 67, no. 6, pp. 661 – 663, 1991.

[13] G. Brassard, "A bibliography of quantum cryptography," 1993, updated by C. Crépeau - url=http://www.cs.mcgill.ca/~crepeau/CRYPTO/Biblio-QC.html.

[14] G. Brassard and C. Crépeau, "Cryptology column - 25 years of quantum cryptography," *ACM SIGACT News*, vol. 27, no. 3, pp. 13 – 24, 1996.

[15] D. Stucki, N. Gisin, O. Guinnard, G. Ribordy, and H. Zbinden, "Quantum key distribution over 67 km with a plug & play system," *New Journal of Physics*, vol. 4, pp. 41.1–41.8, 2002. [Online]. Available: http://www.iop.org/EJ/article/1367-2630/4/1/341/nj2141.html

[16] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Reviews of Modern Physics*, vol. 74, pp. 145 – 190, 2002, arXiv eprint quant-ph/0101098.

[17] D. C. Monyk, "Development of a global network for secure communication based on quantum cryptography," SECOQC Project Proposal, April 2003.

[18] C. Elliott, "Building the quantum network," *New Journal of Physics*, vol. 4, pp. 46.1 – 46.12, 2002. [Online]. Available: http://www.iop.org/EJ/article/1367-2630/4/1/346/nj2146.html

[19] M. Wegman and L. Carter, "New hash functions and their use in authentication and set equality," *Journal of Computer and System Sciences*, vol. 22, pp. 265 – 279, 1981.