page bidon

Introduction

Traditionellement la cryptographie était réservée aux applications militaires et diplomatiques. De nos jours, avec l'avènement des techniques modernes, on lui trouve des applications partout où la sécurité est mise en cause.

Ce mémoire vise deux objectifs bien distincts. Premièrement, il démontre la complexité des transactions qu'il est possible de réaliser avec les techniques de cryptographie moderne. Deuxièmement, on y trouve de nouveaux outils cryptographiques pouvant être utilisés dans des contextes différents de celui proposé.

Le problème auquel ce mémoire apporte une solution est celui de jouer au poker avec des adversaires qu'il n'est pas possible de rencontrer, de façon à ce que personne ne puisse tricher. L'historique de ce problème est présenté au chapitre 1. Le chapitre suivant explique le problème dans tous ses détails: un ensemble de contraintes est présenté spécifiant le but à atteindre. Au chapitre 3 se trouvent les notions élémentaires qu'il faut posséder afin de comprendre les protocoles des chapitres suivants.

Une première solution partielle est suggérée au chapitre 4. Cette solution est essentiellement basée sur des techniques cryptographiques existantes. La solution complète au problème posé se trouve au chapitre 5. On y trouve également des nouveaux protocoles cryptographiques d'usage général dont DISTOR (divulgation de secret tout ou rien).

Jusqu'à quel point peut-on simuler une véritable partie de poker? Avec quel degré de sécurité la partie va-t-elle se dérouler? Quelle puissance de calcul devra être déployée par chaque participant? Le protocole est-il sûr même contre ses concepteurs? Toutes ces questions et bien d'autres ont leur réponse dans ce mémoire.

Si vous êtes d'une nature nerveuse ou expressive et avez renoncé à faire fortune en jouant au poker, car vous n'aviez pas de "poker face", voici la solution à vos problèmes: *un protocole de poker à l'aveugle permettant la confidentialité de la stratégie*.

1. Historique

N individus veulent jouer au poker. Il leur est malheureusement impossible de se rencontrer. Ils ont, par contre, chacun un ordinateur à leur disposition leur permettant de communiquer entre eux et rendant possible certains calculs mathématiques compliqués. Est-il possible qu'ils établissent un protocole leur permettant d'arriver à leur fin? Cette question a été initialement posée par Robert W. Floyd et a déjà été étudiée par plusieurs chercheurs [SRA, GM1, FM, BF, Yu].

Bien entendu, le problème est très simple à résoudre si les joueurs se font mutuellement confiance. Car dans un tel cas on peut programmer les ordinateurs pour décider aléatoirement du brassage du paquet de cartes et ensuite pour distribuer les cartes tel que prescrit par la brasse obtenue. Mais évidemment, personne n'a confiance en ses adversaires. Chacun pense que ses opposants vont tricher aussitôt que l'occasion se présentera, et qu'ils ne se gêneraient pas non plus pour faire tricher leur ordinateur.

Si l'on considère le problème d'un point de vue absolu, c'est-à-dire que chacun des joueurs dispose d'un temps de calcul arbitrairement grand avant chaque action, alors un protocole qui garantit l'unicité de chaque carte et qui ne permet à son adversaire de savoir quoi que ce soit d'utile sur son jeu, n'est pas possible dans le cas $N=2$. La preuve de cet impossibilité est exhibée dans l'article "Mental Poker" de Shamir, Rivest et Adleman [SRA]. Cette preuve s'étend à N quelconque si l'on veut empêcher une coalition de $N-1$ joueurs de tricher contre le joueur isolé. Par contre, on trouve également dans cet article une solution pratique au problème. Paradoxe?

Non, la subtilité réside dans le fait que leur protocole suppose que les joueurs disposent d'un temps de calcul limité, que leur machine a une puissance de calcul raisonnable et que certains problèmes de théorie des nombres sont "difficiles" à résoudre. Dans ce contexte, "difficile" à résoudre veut dire que le problème demande beaucoup plus de temps à résoudre qu'il ne leur est permis d'utiliser sur quelque machine dont ils disposent. Disons qu'en général un problème est considéré difficile, par exemple, si le meilleur algorithme connu nécessite quelques millions d'années sur le meilleur ordinateur du monde pour en résoudre un exemplaire.

Avec cette hypothèse, ils arrivent à construire un protocole ayant certaines des propriétés souhaitables. Notons l'unicité des cartes, c'est-à-dire qu'on garantit que chaque carte apparaît une et une seule fois, et une certaine confidentialité de son jeu. En plus ce protocole se généralise simplement afin de permettre des parties à plus de 2 joueurs. Malheureusement, Lipton [Li] a montré que la confidentialité des cartes n'est pas totale. C'est-à-dire que certaines informations peuvent être calculées sur les cartes de ses adversaires ou même celles restées dans le paquet, en un temps raisonnable. D'autres lacunes de ce protocole sont soulevées dans [Co].

Une solution nouvelle a vu le jour, suite au progrès des techniques cryptographiques. Goldwasser et Micali proposent dans [GM1] un nouveau système d'encryption et une de ses applications: un protocole de poker à deux joueurs cachant toute information partielle sur les cartes. Par contre, un problème surgit. Il n'est pas clair que l'on puisse étendre ce protocole à plus de deux joueurs.

A ce point, plusieurs équipes se lancent à l'assaut du problème à plusieurs joueurs. Trois résultats font d'abord surface; un premier par une équipe hongroise formée de Barany et Füredi qui décrivent un protocole permettant à un nombre de joueurs supérieur à 2 de réaliser une partie sans pouvoir tricher même si on leur accorde un temps de calcul arbitrairement grand [BF]. C'est un progrès impressionnant. Par contre, comme on le sait, puisqu'un tel protocole n'est pas possible pour 2 joueurs, leur méthode ne s'applique pas pour $N=2$ et de plus elle n'admet pas que des joueurs puissent former des coalitions en s'échangeant des informations secrètes à l'insu de leurs adversaires. Mais dans le contexte où l'on tente d'établir le jeu, rien ne garantit que des joueurs ne vont pas se donner un coup de téléphone au milieu de la partie et s'échanger des informations sur l'état du jeu. Il serait catastrophique qu'un joueur perde systématiquement parce que ses adversaires se sont échangé des informations. Pire encore, s'ils pouvaient apprendre son jeu à tout coup, ils pourraient le laisser gagner ou perdre à volonté et ainsi lui donner la conviction que la partie se déroule normalement.

Dans le cas du protocole des hongrois, il est peine perdue d'essayer de régler ce problème, car il est inhérent à l'hypothèse de la puissance de calcul infinie. Yung a également soumis un protocol permettant de jouer au poker à plusieurs mais cette fois avec les mêmes hypothèses que SRA [Yu]. Malheureusement dans ce cas aussi, la catastrophe se produit si les joueurs forment une coalition. Ceux qui, en fait, ont soulevé le problème sont Fortune et Merrit et ils ont proposé à leur tour une solution où, cette fois, il n'est d'aucune utilité de former des coalitions, excepté le fait que les membres d'une telle formation vont s'apprendre mutuellement le contenu de leur jeu respectif [FM]. Cette situation est la meilleure que l'on puisse espérer, car rien ne peut empêcher des joueurs pouvant communiquer entre eux secrètement de se dire le contenu de leur jeu. On parle dans ce cas d'effet minimal des coalitions.

Mais encore une fois, la solution proposée a une lacune majeure : le protocole présume l'existence d'une entité en qui tout le monde a confiance et qui a pour but de brasser les cartes.

C'est à ce point que je me suis penché sur le problème pour la première fois. La première chose qu'il faudrait mettre au clair est l'ensemble des propriétés souhaitables que l'on voudrait obtenir pour un protocole.

2. Contraintes à respecter

Pour qu'un protocol puisse vraiment simuler la réalité, il faut qu'il respecte les contraintes établies par le jeu tel que pratiqué habituellement. Nous supposons ici que le lecteur est familier avec le jeu de poker [Po]. Les contraintes à respecter sont les suivantes:

- Absence d'un parti honnête
- Unicité des cartes
- Distribution uniforme des jeux possibles
- Impossibilité d'influencer la distribution des cartes
- Détection des tricheurs avec haute probabilité
- Confidentialité des cartes
- Effet minimal des coalitions
- Confidentialité complète de la stratégie des joueurs.

1) Absence d'un parti honnête:

L'honnêteté est une vertu qu'il est difficile de faire transparaître à travers un appareil téléphonique ou un réseau d'ordinateurs. Dans le contexte où l'on tente d'établir la partie, il est peu probable que les joueurs soient prêts à se faire confiance, particulièrement s'ils ne se connaissent pas.

2) Unicité des cartes:

Chaque carte doit apparaître une et une seule fois dans le paquet simulé. Il doit y avoir un moyen de détecter un joueur qui tenterait, en ne respectant pas son protocole, d'éliminer ou de répéter une carte.

3,4) Distribution uniforme des jeux possibles et impossibilité d'influencer la distribution des cartes:

Chaque jeu possible doit être accessible à tous les joueurs avec la distribution de probabilité uniforme. Il est essentiel que la distribution de probabilité des cartes ne puisse être influencée par un joueur ou par un groupe de joueurs. Chaque joueur devrait avoir droit à tous les jeux possibles avec probabilité égale, quelques soient les interventions de ses adversaires.

5) Détection des tricheurs avec haute probabilité:

Toute tentative de triche devrait être détectée avec forte probabilité. Il serait souhaitable que cette probabilité puisse être réduite à volonté de façon à ce que les joueurs établissent cette valeur en fonction de la confiance mutuelle qu'ils se portent. Idéalement, la probabilité de tricher sans être détecté devrait décroître exponentiellement quand le travail à effectuer croît polynomialement.

6) Confidentialité des cartes:

La confidentialité complète des cartes implique qu'il soit impossible de calculer une quelconque information sur une carte qui ne soit pas parmi celles légitimement connues par un joueur, en un temps raisonnable.

7) Effet minimal des coalitions:

Comme on l'a déjà vu, le mieux que l'on puisse espérer est que des joueurs formant une coalition ne puissent rien apprendre de plus que l'ensemble des cartes connues par au moins un membre de la coalition.

8) Confidentialité complète de la stratégie des joueurs:

La confidentialité de la stratégie est un concept général qui se manifeste de plusieurs façons dans le jeu de poker. Par exemple, les joueurs perdants ont le privilège de ne pas montrer leur jeu à la fin de la partie. Ceci permet à un joueur, par exemple, de ne pas laisser savoir aux autres qu'en fait il a pris une mauvaise décision, ou encore qu'il a misé gros sur un jeu tout à fait ordinaire (c'est-à-dire: il a bluffé). Plusieurs éléments font partie de la stratégie du poker. Pour un joueur professionel, le gain ne dépend pas tellement du hasard mais surtout de sa stratégie.

Un autre exemple se produit lorsqu'un joueur dévoile son jeu dans le but de remporter la partie. A ce moment, il n'est pas acceptable que les joueurs sachent l'origine des cartes présentées. C'est-à-dire de savoir lesquelles proviennent de la donne initiale et lesquelles de la seconde. Par exemple, si j'ai reçu d'abord le jeu A et que j'écarte les deux premières cartes et en demande deux nouvelles pour en arriver à la situation du jeu B

alors il est important que les autres ne sachent pas lesquelles de ces cartes sont les nouvelles arrivées. Dans cet exemple, ils sauraient que mon jeu n'est devenu intéressant seulement qu'après avoir reçu mes cartes à la seconde donne.

De tous les protocoles mentionnés jusqu'ici aucun ne satisfait plus de 6 des 8 contraintes suggérées. Ce que propose le reste de ce mémoire est d'abord une solution qui satisfait les 7 premières exigences (chap. 4). Celle-ci permettra, entre autre, de définir des concepts et de se familiariser avec les méthodes utilisées. Ensuite, vient une solution complète au problème, satisfaisant les 8 contraintes proposées. Cette solution beaucoup plus complexe demandera un certain préambule présenté dans les sections (5.1, 5.2.3). On verra en cours de route de nombreux sous-problèmes à résoudre et leur solution (sections 5.1.1, 5.1.2.1, 5.1.2.2). Mais voyons d'abord quelques notions préliminaires nécessaires à la compréhension des protocoles qui suivront.

3. Notions préliminaires

3.1. Théorie des nombres

Soient $\mathbb{Z}=\{...,-4,-3,-2,-1,0,1,2,3,4,...\}$ l'ensemble des entiers relatifs et $\mathbb{N}=\{1,2,3,4,...\}$ l'ensemble des entiers positifs. Soient $a,b,c,d\in\mathbb{Z}$ et $m,n\in\mathbb{N}$. On dit que

Définition 1:

a divise b (noté $a \mid b$) si $\frac{b}{a}\in\mathbb{Z}$

Définition 2:

a est un facteur de b si $a \mid b$

Définition 3:

a est congru à b modulo n (noté $a \equiv b \pmod{n}$) si $n \mid (b-a)$

Définition 4:

$n>1$ est premier si $\{m \in \mathbb{N} \mid m \mid n\}=\{1,n\}$

Définition 5:

$n>1$ est composé s'il n'est pas premier

Définition 6:

m est le plus grand commun diviseur de a et b (noté $m=(a,b)$) si $m \mid a$ et $m \mid b$ et $n>m \implies n \nmid a$ ou $n \nmid b$.

Afin d'éviter les confusions, définissons la fonction **mod** $:\mathbb{Z}\times\mathbb{N}\rightarrow\mathbb{N}\cup\{0\}$ de la façon suivante:

Définition 7:

a **mod** $n = min\{m \mid m\geq 0$ et $a \equiv m \ (mod\ n)\}$

Ainsi aucune ambiguité n'est possible entre la fonction **mod** et la relation

$(mod\ \ n)$.

Définition 8: $\mathbb{Z}_n^* = \{m \mid 0<m<n \text{ et } (n,m)=1\}$.

Définition 9:

$x \in \mathbb{Z}_n^*$ est un résidu quadratique modulo n s'il existe $y \in \mathbb{Z}_n^*$ tel que $x \equiv y^2 (mod\ n)$. Sinon x est un non-résidu quadratique.

Définition 10:

$RQ_n = \{x \in \mathbb{Z}_n^* \mid x \text{ est un résidu quadratique modulo } n\}$

$NRQ_n = \{x \in \mathbb{Z}_n^* \mid x \text{ est un non-résidu quadratique modulo } n\}$

Soit p un nombre premier. Alors pour chaque $x \in \mathbb{Z}_p^*$ on définit:

Définition 11:

Le symbole de Legendre $\left(\frac{x}{p}\right) = \begin{cases} +1 & \text{si } x \in RQ_p \\ -1 & \text{sinon } (x \in NRQ_p) \end{cases}$

De façon générale on définit le symbole de Jacobi $\left(\frac{x}{n}\right)$ de x par rapport à n comme le produit des symboles de Legendre des diviseurs premiers de n avec répétition si il y a lieu. Si $n = p_1 \times p_2 \times \cdots \times p_k$ avec chaque p_i un nombre premier, alors on aura $\left(\frac{x}{n}\right) = \left(\frac{x}{p_1}\right) \times \left(\frac{x}{p_2}\right) \times \cdots \times \left(\frac{x}{p_k}\right)$.

Clairement, selon cette définition $\left(\frac{x}{n}\right) = \pm 1$ quel que soit $x \in \mathbb{Z}_n^*$. Soit $X \subseteq \mathbb{Z}_n^*$, on dénote $X[+1] = \{x \in X \mid \left(\frac{x}{n}\right) = +1\}$ et similairement pour $X[-1]$.

Remarquons que les symboles de Jacobi et de Legendre peuvent être calculés efficacement de la façon suivante [RSA]:

$$\left(\frac{x}{n}\right) = \begin{cases} +1 & \text{si } x=1 \\ \left(\frac{n \ \mathbf{mod}\ x}{x}\right)(-1)^{\frac{(n^2-1)(x^2-1)}{4}} & \text{si } x \text{ est impair} \\ \left(\frac{x/2}{n}\right)(-1)^{\frac{(n^2-1)}{8}} & \text{si } x \text{ est pair} \end{cases}$$

Ces définitions sont celles que nous utiliserons tout au long de ce mémoire. A certains moments il nous arrivera d'énoncer certains théorèmes exposant des faits connus de la théorie des nombres utilisant ces définitions. Il pourra être utile au lecteur peu expérimenté en théorie des nombres de se référer à cette section afin de se remémorer les définitions utilisées. Pour plus de détails ou pour les preuves de théorèmes consulter [NZ].

3.2. Complexité du calcul

La sécurité des protocoles exposés dans ce mémoire est essentiellement basée sur la théorie de la complexité du calcul, particulièrement sur la difficulté présumé de résoudre certains problèmes de théorie des nombres. Voici donc un bref exposé des principes de cette science et de ses relations avec les fonctions arithmétiques utilisées dans cet ouvrage.

La complexité du calcul est une science qui évalue la quantité de ressources (temps et espace mémoire en général) nécessaires à la résolution d'un problème. Il semble que tous les problèmes de la nature ne soient pas nés égaux. Certains problèmes ont des solutions efficaces, d'autres pas. Dans ce contexte nous dirons qu'un problème est intraitable s'il n'existe aucune méthode efficace pour le solutionner. Malheureusement, on ne connait actuellement pas de preuve démontrant qu'aucune solution efficace n'existe pour un problème dont on peut vérifier les solutions rapidement ($\mathbf{P} \neq \mathbf{NP}?$). Ceci est dû au fait que la complexité est une science toute jeune . Souvent par contre, on accepte comme argument convaincant le fait que des siècles de recherche n'ont produit aucune solution efficace. Considérons par exemple le problème de la factorisation des entiers. Depuis l'antiquité grecque, les mathématiciens cherchent une technique efficace pour factoriser les entiers. Même à l'heure actuelle, aucun algorithme sur aucun ordinateur ne peut factoriser un nombre de plus de 200 chiffres décimaux en un temps raisonnable. Plusieurs millions d'années de calculs seraient nécessaires pour effectuer un tel travail selon les méthodes actuelles et les améliorations technologiques anticipées. Pourtant aucun argument connu ne nous permet d'affirmer qu'une telle chose est impossible. Peut-être que quelqu'un trouvera un jour une solution permettant de résoudre le problème rapidement. Il est même posible qu'une telle solution soit déjà connue mais gardée jalousement secrète!

Le problème que nous supposerons intraitable n'est pas prouvé difficile à résoudre. Heureusement, dans ce cas particulier, l'incapacité historique à le solutionner efficacement nous permet de présumer qu'il soit intraitable. Nous considérerons le problème qui suit.

Soit n un entier dont la factorisation est inconnue. Prenons un x quelconque dans $\mathbb{Z}_n^*[+1]$. Il s'agit de déterminer si $x \in RQ_n$ ou $x \in NRQ_n$. Présentement les seules techniques permettant de résoudre ce problème ont besoin de la factorisation de n. Dénotons D_k l'ensemble des nombres n'ayant que deux facteurs premiers de k bits chacun. Ces nombres sont considérés comme les plus difficiles à factoriser. On présumera que la factorisation des grands nombres dans D_k est intraitable.

Nous dénoterons HRQ l'hypothèse selon laquelle le problème de décider la résiduosité quadratique des nombres de $\mathbb{Z}_n^*[+1]$ pour un nombre n assez grand dans D_k est intraitable. Evidemment certains exemplaires de ce problème sont faciles à résoudre. Par exemple, si $x<n$ est un carré dans les entiers, alors il est un résidu quadratique modulo n. Mais ces nombres sont tellement peu nombreux qu'on peut les négliger. Nous présumons que sous HRQ la probabilité de trouver un exemplaire sur lequel on puisse facilement obtenir la résiduosité quadratique, est exponentiellement faible par rapport à k.

Ce problème de décision est bien connu des théoriciens des nombres pour sa difficulté. Il constitue un des quatres principaux problèmes algorithmiques trouvés dans "Disquisitiones Arithmeticae" (1801) de Gauss [Ga]. Une solution efficace à ce problème entraînerait une solution efficace pour d'autres problèmes ouverts de théorie des nombres, tel le problème de décider si un nombre composé possède 2 ou 3 facteurs premiers [GM2,Ad].

La véritable raison pour s'intéresser à ce problème dans cet ouvrage est une propriété extrêmement remarquable: étant donnés n et y un quelconque non-résidu quadratique modulo n, il est très facile à quiconque de générer des nombres qui soient des résidus ou des non-résidus modulo n, à son choix. Par contre il est difficile, à partir de ces nombres, de décider s'ils sont des résidus ou non, quand la factorisation de n n'est pas connue.

De plus, il existe une clef permettant de résoudre aisément les exemplaires du problème. Dans ce cas, la clef est la factorisation de n. Si quelqu'un connaît cette factorisation alors il peut à volonté déterminer si $x \in RQ_n$ ou $x \in NRQ_n$. Par contre, ceux qui ne connaissent pas la factorisation ne peuvent accomplir cette tâche en un temps raisonnable (sous HRQ).

3.3. Technique d'attestation.

A travers les protocoles proposés, on utilisera les propriétés du problème cité à la section précédente comme méthode d'attestation. Une méthode d'attestation permet de rendre publique une valeur déterminant uniquement une variable donnée. Mais sans la "clef" connue uniquement par le propriétaire de la variable, il est difficile d'en déterminer la valeur. Formellement, une méthode d'attestation est un quintuplet $<X,Y,R,F,f>$ où X, Y et R sont trois ensembles, F est une fonction de $X \times R \rightarrow Y$ et f est une fonction de $Y \rightarrow X$ tels que:

$$\forall x \in X, \forall r \in R \quad f(F(x,r)) = x$$

$F(x,r)$ étant facile à calculer mais $f(y)$ étant intraitable. C'est-à-dire qu'il est facile de calculer $F(x,r)$ étant donné x et r mais difficile de trouver x étant donné $F(x,r)$.

Commençons par un exemple. Supposons qu'Alice et Bob veulent tirer à pile ou face. Chacun d'eux choisit un bit secret. La valeur tirée sera pile si les deux bits sont identiques et face sinon. Evidemment, si l'un des deux déclare son bit d'abord, l'autre n'aura qu'à choisir subséquemment le bit qui lui convient afin de produire pile ou face. Voici une solution: Alice choisit deux grands nombres premiers p,q au hasard (disons de 200 chiffres décimaux de long). Alice calcule $n=pq$ et informe Bob de la valeur de n. Ensuite, Alice choisit au hasard un nombre x qui est un résidu quadratique dans $\mathbb{Z}_n^*[+1]$ si son bit secret est 0 et un non-résidu quadratique dans $\mathbb{Z}_n^*[+1]$ si son bit est 1. Alice annonce x à Bob. Sous HRQ, il n'est pas possible pour Bob de déterminer en un temps raisonnable si x est un résidu ou non, donc de déterminer si le bit d'Alice est 0 ou 1. Par contre il est maintenant impossible pour Alice de changer la valeur de son bit car la valeur de x détermine uniquement celle-ci. Bob n'a plus qu'à annoncer son bit à Alice. Maintenant que chacun connaît le bit de Bob, Alice va donner à Bob les facteurs de n de façon à ce que Bob puisse constater si x est un résidu ou non, et du même coup apprendre quel était le bit secret d'Alice. Il ne reste qu'à décider du résultat final selon que les bits sont identiques ou non.

Il reste plusieurs détails à éclaircir dans ce protocole. Comment Alice peut-elle choisir des grands nombres premiers au hasard? Comment peut-elle choisir un résidu ou un non-résidu quadratique modulo n au hasard? Et en quoi la factorisation de n permet-elle à Bob de déterminer si x est un résidu ou non?

D'abord il est possible de décider si un nombre est premier rapidement en utilisant un test de primalité probabiliste comme celui de Rabin [Ra1]. Pour choisir un nombre premier d'une certaine taille (nombre de bit) k au hasard il suffit d'itérer le processus de choisir un nombre quelconque r de taille k et ensuite vérifier s'il est premier. D'après un théorème de la théorie des nombres, il faudra essayer en moyenne $k \ln 2$ valeur de r avant d'en trouver une qui soit un nombre premier. Alice peut donc choisir p et q selon cette méthode.

Pour déterminer si un nombre est un résidu quadratique ou un non-résidu quadratique dans $\mathbb{Z}_n^*[+1]$ on peut utiliser le théorème suivant:

Théorème 1:

$$RQ_n[+1] = RQ_n = \mathbb{Z}_p^*[+1] \cap \mathbb{Z}_q^*[+1] \quad \text{et}$$
$$NRQ_n[+1] = \mathbb{Z}_p^*[-1] \cap \mathbb{Z}_q^*[-1]$$

Afin de déterminer si $x \in \mathbb{Z}_p^*[\pm 1]$, il suffit d'utiliser la formule récursive donnée à la section 3.1 (similairement pour q). Si on utilise également ce deuxième fait:

Théorème 2:

$$\#RQ_n = \#NRQ_n[+1] = \frac{\#\mathbb{Z}_n^*}{4} = \frac{(p-1)(q-1)}{4}$$

alors il suffit de choisir des nombres a au hasard dans $\mathbb{Z}_n^*$ jusqu'à ce que $\left[\frac{a}{p}\right]$ et $\left[\frac{a}{q}\right]$ satisfassent les conditions voulues. Avec une très grande probabilité, on trouvera un tel nombre en quelques essais.

Remarquons qu'il aurait été possible également de produire des résidus quadratiques simplement en choisissant un nombre a au hasard dans $\mathbb{Z}_n^*$ et en calculant $x = a^2 \ \mathbf{mod}\ n$. On peut montrer que tous les carrés modulo n peuvent être produits selon cette méthode avec une distribution uniforme. Par contre, pour produire un non-résidu quadratique dans $\mathbb{Z}_n^*[+1]$, il faut faire un peu plus de travail. Mais en général, une fois qu'on dispose d'un premier non-résidu dans $\mathbb{Z}_n^*[+1]$, on peut produire tous les autres en multipliant celui que l'on possède par les résidus quadratiques de $\mathbb{Z}_n^*$.

Soit $y \in NRQ_n[+1]$.

Théorème 3:

$$NRQ_n[+1] = \{xy \ \mathbf{mod}\ n \ \mid \ x \in RQ_n\}$$

Remarquons que puisque $\#RQ_n = \#NRQ_n[+1]$ alors

$$\begin{aligned} RQ_n &\rightarrow RNQ_n[+1] \\ x &\mapsto xy \ \mathbf{mod}\ n \end{aligned}$$

est une bijection.

Ceci nous amène donc à faire la constatation que la production d'exemplaires du problème de la résiduosité quadratique est facile dès que l'on dispose d'un non-résidu quadratique dans $\mathbb{Z}_n^*[+1]$. C'est-à-dire qu'il faut faire une multiplication modulo n pour un résidu quadratique et deux pour un résidu non-quadratique. Ceci constitue une propriété essentielle qui nous a fait choisir ce problème comme méthode d'attestation des données. Dans ce cas nous choisissons $X = \{0,1\}$, $R = \mathbb{Z}_n^*$, $Y = \mathbb{Z}_n^*[+1]$, $F(x,r) = r^2 y^x \ \mathbf{mod}\ n$ et $f(u) = \begin{cases} 0 & \text{si } u \in RQ_n \\ 1 & \text{sinon } (u \in NRQ_n) \end{cases}$.

Comme dans l'exemple du tirage à pile ou face, on peut en général établir un parallèle entre la valeur binaire 0 et les résidus quadratiques ainsi qu'entre la valeur 1 et les non-résidus quadratiques. Si l'on veut attester une chaîne binaire par des entiers de $\mathbb{Z}_n^*[+1]$, on n'a qu'à choisir aléatoirement des résidus quadratiques pour attester les 0 et des non-résidus pour les 1. Remarquez que cette opération peut être effectuée par n'importe qui connaissant n et un quelconque non-résidu quadratique modulo n car la technique de génération mentionnée précédemment ne suppose nullement que celui qui l'exécute connaisse la factorisation de n. Etant donné $y \in NRQ_n[+1]$ on dira que $\hat{b} \in \mathbb{Z}_n^*[+1]$ atteste $b \in \{0,1\}$ si $\exists r \in \mathbb{Z}_n^*$ tel que $\hat{b} \equiv r^2 y^b \ (mod\ n)$ et on parlera d'*ouvrir* $\hat{b}$ lorqu'on révèlera un tel r.

Rappelons qu'il n'est pas possible (sous HRQ) à quiconque ne connaissant pas la factorisation de n de déterminer si un nombre dans $\mathbb{Z}_n^*[+1]$ est un résidu quadratique ou non modulo n. De plus, si l'on dispose d'un sous-ensemble de nombres dans $\mathbb{Z}_n^*[+1]$ alors il n'est pas possible d'établir une corrélation entre leurs résiduosités quadratiques. Il n'est pas possible, par exemple, de décider si deux nombres pris au hasard dans $\mathbb{Z}_n^*[+1]$ sont de même résiduosité ou de résiduosité inverse. Cette propriété est à l'origine de l'intérêt que portent les cryptographes à ce problème [GM2]. L'encryption probabiliste est la technique selon laquelle Alice peut envoyer à Bob un message secret en représentant chaque bit 0 de son message par un résidu quadratique modulo n choisi au hasard et chaque bit 1 par un non-résidu modulo n. Ici, n est fourni par Bob qui en connait la factorisation. Bob met également à la disposition d'Alice un non-résidu quadratique qu'elle utilisera pour en produire d'autres. Bien qu'on pourrait croire le contraire, ce non-résidue quadratique ne donne aucun avantage à Alice afin de décider la résiduosité quadratique des nombres de $\mathbb{Z}_n^*[+1]$ (voir [GM2]). Bob pourra enfin prendre connaissance du message d'Alice en décidant si les nombres employés sont des résidus modulo n ou non grâce à la factorisation que lui seul connait. Sous HRQ ceci garantit qu'aucune information ne peut être extraite par un tiers de la séquence de nombres dans $\mathbb{Z}_n^*[+1]$ qu'Alice a transmise à Bob.

Une autre propriété exceptionelle de $\mathbb{Z}_n^*[+1]$ est liée à sa structure algébrique. Le parallèle entre 0 et les résidus quadratiques et entre 1 et les non-résidus quadratiques s'étend davantage puisqu'on peut même effectuer certaines opérations binaires sur les nombres de $\mathbb{Z}_n^*[+1]$ correspondant à des opérations binaires sur $\{0,1\}$. C'est ainsi que le produit modulo n dans $\mathbb{Z}_n^*[+1]$ correspond au ou-exclusif sur $\{0,1\}$. Plus précisément, si b_1 et b_2 sont deux bits que l'on atteste par $\hat{b}_1$ et $\hat{b}_2$ dans $\mathbb{Z}_n^*[+1]$ alors $\hat{b}_1 \times \hat{b}_2 \ \mathbf{mod}\ n$ est une attestation de $b_1 \oplus b_2$. Cette propriété sera essentielle dans les protocoles qui suivent et nous verrons à ce moment de quelle façon.

C'est à cause de toutes ces raisons que j'ai choisi la représentation des données inspirée par la technique d'encryption probabiliste. Cette représentation sert principalement de méthode d'attestation puisque la représentation de chaque bit en un résidu ou un non-résidu le détermine uniquement. De plus, sous HRQ, la valeur du bit restera secrète aussi longtemps que celui qui connaît la factorisation de n le voudra.

4. Un premier protocole.

4.1. Préparation

Supposons que $J_1, J_2, \ldots, J_N$ soient les N joueurs voulant participer à la partie de poker. Supposons sans perte de généralité que les joueurs soient ordonnés selon $1,2,3,\ldots,N$. Considérons une correspondance entre le paquet de 52 cartes et l'ensemble $\{1,2,\ldots,52\}$. Tout au long de ce mémoire, il ne sera jamais question explicitement des cartes mais seulement de leur numéro établi par cette correspondance.

Contrairement à une véritable partie de poker, il n'existe pas de brasseur dans mes protocoles. A chaque brasse, tous les participants vont être impliqués dans l'opération qui mélange le paquet de carte. Intuitivement, chaque joueur J_i va choisir une permutation π_i de $\{1,2,\ldots,52\}$. Le paquet brassé sera la permutation composée $\pi_N \pi_{N-1} \cdots \pi_1$. Pour obtenir une carte, un joueur va choisir un nombre k dans $\{1,2,\ldots,52\}$ et sa carte sera alors $\pi_N \pi_{N-1} \cdots \pi_1(k)$ (i.e. $\pi_N(\pi_{N-1}(\cdots(\pi_1(k))\cdots))$). Evidemment, si l'on veut que le jeu des joueurs reste secrèt il faudra que les permutations π_i restent secrètes aussi. De plus, il doit être impossible de les modifier après le début de la partie. Bien entendu, les joueurs devront lire certaines entrées des permutations secrètes de leurs adversaires afin de connaître leurs cartes. C'est l'implantation de cette opération qui constitue l'essentiel de ce protocole. Afin de voir clairement le but des sections suivantes, voici le protocol global vers lequel nous nous acheminons:

PROTOCOLE 1 Poker (version 1)

(1) Préparation
(2) Lecture des cartes
(3) Ouverture des jeux
(4) Identification des tricheurs

Chaque joueur J_i devra d'abord créer sa propre méthode d'attestation. Pour ce faire, il va générer un couple de grands nombres premiers p_i, q_i et calculer leur produit n_i. Ensuite par tâtonnement il devra trouver un quelconque non-résidu quadratique y_i dans $\mathbb{Z}_{n_i}^*[+1]$. Ces deux valeurs devront être devoilées à ses adversaires afin de leur permettre de les utiliser de la façon explicitée plus bas. Ensuite, J_i génère une permutation π_i qui sera utilisée pour déterminer le brassage des cartes. Afin de fixer sa permutation, J_i va l'attester à l'aide de n_i. Considérons les nombres dans $\{1,2,...,52\}$ comme des chaînes binaires de six bits chacune (000001_2 à 110100_2). Soit $b_{i,j,k}$ le $j^{\text{ième}}$ bit du nombre $\pi_i(k)$. J_i va présenter à ses adversaires la valeur de $b_{i,j,k}$ à l'aide d'un résidu quadratique modulo n_i si ce bit est 0 et d'un non-résidu avec symbole de Jacobi +1 sinon. Il attestera ce bit par

$$\hat{b}_{i,j,k} = x_{i,j,k}^2 y_i^{b_{i,j,k}} \ \mathbf{mod}\ n_i$$

pour un $x_{i,j,k}$ choisi au hasard dans $\mathbb{Z}_{n_i}^*$. La valeur de $\hat{b}_{i,j,k}$ une fois rendue publique va déterminer uniquement celle de $b_{i,j,k}$, laquelle (sous HRQ) ne sera connue des autres joueurs qu'avec la collaboration de J_i.

PROTOCOLE 2 Préparation

(1) POUR chaque $i \in \{1,...,N\}$ FAIRE
(1.1) J_i FAIT
(1.1.1) choisir p_i, q_i des grands nombres premiers distincts, de même longueur.
(1.1.2) trouver par essai et erreur y_i tel que $\left(\frac{y_i}{p_i}\right) = \left(\frac{y_i}{q_i}\right) = -1$.
(1.1.3) dévoiler publiquement $n_i = p_i q_i$ et y_i.
(1.1.4) choisir π_i une permutation de $\{1,2,...,52\}$.
(1.1.5) POUR chaque $j \in \{1,...,6\}$ et chaque $k \in \{1,2,...,52\}$
(1.1.5.1) $b_{i,j,k} \leftarrow j^{\text{ième}}$ bit de $\pi_i(k)$.
(1.1.5.2) choisir aléatoirement $x_{i,j,k} \in \mathbb{Z}_{n_i}^*$.
(1.1.5.3) $\hat{b}_{i,j,k} \leftarrow x_{i,j,k}^2 y_i^{b_{i,j,k}} \ \mathbf{mod}\ n_i$.
(1.1.5.4) dévoiler publiquement $\hat{b}_{i,j,k}$.
(1.1.6) FIN
(1.2) FIN
(2) FIN
(3) POUR chaque $k \in \{1,2,...,52\}$, marquer k "libre".

4.2. Lecture des cartes

Une fois ce protocole complété, il sera possible de choisir des cartes. Pour ce faire, lorsqu'un joueur J_i veut obtenir une carte il choisit une valeur k "libre" dans $\{1,2,...,52\}$ et sa carte sera $\pi_N \pi_{N-1} \cdots \pi_1(k)$. On dira à l'avenir que k est l'identificateur de cette carte. J_i ne peut prédire sa carte car il connaît seulement π_i. Ceci demeure vrai même s'il est en coalition avec tous les autres joueurs sauf un. Ceci implique qu'il devra apprendre $\pi_1(k)$, $\pi_2\pi_1(k)$, ..., $\pi_N \pi_{N-1} \cdots \pi_1(k)$ grâce à ses adversaires. Puisque J_i ne fait confiance à aucun d'entre eux, s'il veut que $\pi_N \pi_{N-1} \cdots \pi_1(k)$ reste secret il devra garder secret $\pi_{N-1} \cdots \pi_1(k)$, $\pi_{N-2} \cdots \pi_1(k)$, ..., $\pi_i \cdots \pi_1(k)$ car n'importe laquelle de ces valeurs permettrait à certains de ses adversaires, s'ils collaborent entre eux, de déterminer sa carte. Par contre $\pi_{i-1} \cdots \pi_1(k)$, $\pi_{i-2} \cdots \pi_1(k)$, ..., $\pi_1(k)$ peuvent être devoilées publiquement puisque π_i sera appliquée au résultat et que cette permutation est connue uniquement de J_i. Donc $\pi_i \pi_{i-1} \cdots \pi_1(k)$ sera secrète même si son argument ne l'est pas. Mais alors, comment J_i va-t-il réussir à découvrir $\pi_{i+1} \cdots \pi_1(k)$, ..., $\pi_{N-1} \cdots \pi_1(k)$, $\pi_N \pi_{N-1} \cdots \pi_1(k)$ sans dévoiler ces valeurs à ses adversaires?

Pour lire une entrée de la permutation d'un de ses adversaires J_d sans que ce dernier ne sache laquelle, J_i procède de la façon suivante. Supposons que J_i veuille lire l'entrée $\pi_d(k)$ de la permutation appartenant à J_d. Il choisit six bits $m_1, m_2, \ldots, m_6$ au hasard et choisit également $w_1, w_2, \ldots, w_6$ au hasard dans $\mathbb{Z}_{n_d}^*$. Ensuite il calcule Q_l de la façon suivante pour $l \in \{1,2,3,4,5,6\}$:

$$Q_l = \hat{b}_{d,l,k} w_l^2 y_d^{m_l} \textbf{ mod } n_d$$

et demande à J_d si Q_l est un résidu quadratique modulo n_d ou non. Si Q_l est un résidu quadratique alors $b_{d,l,k} = m_l$ et $b_{d,l,k} = \neg m_l$ sinon.

Théorème 4:

Si Q_l est un résidu quadratique alors $b_{d,l,k} = m_l$ et $b_{d,l,k} = \neg m_l$ sinon.

Preuve:

Comme on le sait, le produit sur $\mathbb{Z}_n^*[+1]$ correspond au ou-exclusif sur $\{0,1\}$. On conclut du fait que $Q_l = \hat{b}_{d,l,k} w_l^2 y_d^{m_l} \textbf{ mod } n_d$, que Q_l est le produit d'une attestation de $b_{d,l,k}$ et d'une de m_l. Donc que Q_l est une attestation de $b_{d,l,k} \leftarrow m_l$. On en conclut directement le résultat.

C.Q.F.D.

Ceci permet à J_i de lire un bit de la permutation de J_d mais encore faut-il que cette opération garde secret ce bit aux yeux des autres joueurs, incluant J_d. C'est-à-dire qu'il ne doit pas être possible d'apprendre quel bit a été lu à partir de Q_l. Une condition suffisante à ce résultat est que la probabilité de produire la question Q_l soit la même, quel que soit le bit que l'on tente de lire.

Soit $x \in \mathbb{Z}^*_{n_d}[+1]$.

Théorème 5:

$$prob(Q_l = x \mid Q_l \text{ sert à déterminer } b_{d,l,k}) = \frac{2}{\# \mathbb{Z}^*_{n_d}}.$$

Preuve: On a que

$$\#\{(w,m) \mid \hat{b}_{d,l,k} w^2 y_d^m \equiv x \pmod{n_d}\} = \#\{(w,m) \mid w^2 \equiv x \hat{b}^{-1}_{d,l,k} y_d^{-m} \pmod{n_d}\}$$

Sachant que la congruence de droite n'admet une solution que pour la valeur de m faisant de la partie droite de la congruence un carré modulo n_d et sachant qu'en général il existe exactement quatre racines carrées à une équation quadratique modulo n_d, on conclut:

$$\#\{(w,m) \mid \hat{b}_{d,l,k} w^2 y_d^m \equiv x \pmod{n_d}\} = 4$$

Donc

$$prob(Q_l = x \mid Q_l \text{ sert à déterminer } b_{d,l,k}) = \frac{\#\{(w,m) \mid \hat{b}_{d,l,k} w^2 y_d^m \equiv x \pmod{n_d}\}}{\#\{(w,m) \mid w \in \mathbb{Z}^*_{n_d} \text{ et } m \in \{0,1\}\}}$$

$$= \frac{4}{2 \times \# \mathbb{Z}^*_{n_d}} = \frac{2}{\# \mathbb{Z}^*_{n_d}}$$

C.Q.F.D.

On constate que la méthode suggérée permet à J_i de lire n'importe quel bit de la permutation de J_d à la condition que ce dernier collabore, et ce sans que personne ne puisse déterminer lequel de ces bits a été lu. Donc J_d a un contrôle sur la quantité de bits qu'il donne à J_i puisqu'il doit intervenir chaque fois que celui-ci en demande un. Par contre, à première vue, J_d n'a aucun contrôle sur quels bits ont été lus par J_i; il peut en lire la bonne quantité mais pas nécessairement ceux auxquels il a droit. Ceci permettrait à J_i, par exemple, de lire les cartes d'un de ses adversaires, au prix de ne pas connaître les siennes. Cette situation peut être profitable dans le cas où le tricheur donne à un autre joueur les informations qu'il acquiert illégalement. Ce dernier pourra profiter du résultat des actes de son associé. Nous verrons comment un tel tricheur sera identifié dans la prochaine sous-section. Pour le moment voici le protocole formel pour lire des cartes:

PROTOCOLE 3 lecture des cartes.

(1) POUR chaque $i \in \{1,...,N\}$ FAIRE
(1.1) REPETER selon les règles du jeu
(1.1.1) lecture d'une carte(i).
(1.2) FIN
(2) FIN

PROTOCOLE 3.1 lecture d'une carte(i).

(1) J_i FAIT
(1.1) choisir $k \in \{1,2,...,52\}$ "libre".
(1.2) marquer k "utilisée".
(1.3) $c \leftarrow k$.
(1.4) POUR $d \leftarrow 1$ JUSQU'A $i-1$ FAIRE
(1.4.1) demander à J_d la valeur de $\pi_d(c)$.
(1.4.2) J_d dévoile publiquement $\pi_d(c)$.
(1.4.3) $c \leftarrow \pi_d(c)$.
(1.5) FIN
(1.6) $c \leftarrow \pi_i(c)$.
(1.7) POUR $d \leftarrow i+1$ JUSQU'A N FAIRE
(1.7.1) obtenir $\pi_d(c)$ en utilisant lecture d'une entrée(i,d,c).
(1.7.2) $c \leftarrow \pi_d(c)$.
(1.8) FIN
(1.9) $carte \leftarrow c$.
(2) FIN

PROTOCOLE 3.1.1 lecture d'une entrée(i,d,c).

(1) J_i FAIT

(1.1) POUR chaque $l \in \{1,...,6\}$ FAIRE

(1.1.1) obtenir $entrée_l$ en utilisant lecture d'un bit(i,d,c,l).

(1.2) FIN

(1.3) $entrée \leftarrow \sum_{l=1}^{6} 2^{l-1} \times entrée_l$.

(2) FIN

PROTOCOLE 3.1.1.1 lecture d'un bit(i,d,c,l).

(1) J_i FAIT

(1.1) choisir $w_l \in \mathbb{Z}_{n_d}^*$ et $m_l \in \{0,1\}$.

(1.2) $Q_l \leftarrow \hat{b}_{d,l,c} w_l^2 y_d^{m_l}$ **mod** n_d

(1.3) demander à J_d si Q_l est dans RQ_{n_d} ou NRQ_{n_d}.

(1.4) J_d répond décision(Q_l,d) à la question de J_i.

(1.5) $r \leftarrow$décision(Q_l,d).

(1.6) $b_{d,l,c} \leftarrow m_l \oplus r$.

(1.7) $bit \leftarrow b_{d,l,c}$.

(2) FIN

FONCTION décision(x,d)

(1) SI $x \in RQ_{n_d}$ ALORS RETOURNER 0 SINON RETOURNER 1.

4.3. Déroulement de la partie.

Comme le veut la règle au Poker, après une première distribution de 5 cartes par joueur, chacun a la possibilité d'écarter une partie de son jeu incluant au maximum 4 cartes et d'obtenir de nouvelles cartes du paquet. Dans cette première version du protocole de poker, pour écarter une carte il suffit de la désigner par son identificateur et de l'indiquer "écartée". Ensuite, une nouvelle carte peut être obtenue en répétant le protocole de lecture de cartes.

Le reste de la partie ne nécessite pas d'opérations spéciales. Aucune attention particulière n'est portée à la portion de la partie où les joueurs font leurs mises. Il faut cependant se rendre compte que si les joueurs ne se font vraiment pas confiance, il faudra faire appel à un système d'identification qui permet de vérifier l'origine des messages échangés. Par exemple, si les joueurs disputent leur partie sur un réseau d'ordinateurs, il sera important qu'on puisse identifier un joueur ayant envoyé le message: "Je mise 1000 dollars" sans se tromper pour qu'il ne puisse pas prétendre plus tard que le message a été émis par quelqu'un d'autre que lui. Le problème d'identification a longuement été étudié par plusieurs chercheurs du monde de la cryptographie. Pour connaître des techniques d'identification on peut consulter [B], [GMRi], [GMY], [Ra2] ,[RSA] et [WC].

4.4. Ouverture des jeux.

A la fin de la partie, une fois les mises terminées, il faut déterminer le gagnant. Pour ce faire, les joueurs ayant misé le plus fort et n'ayant pas abandonné lors des mises peuvent remporter la partie. Le gagnant est celui qui produira le jeu le plus fort selon l'ordre établi par les règles du jeu. Chacun va révéler son jeu aux autres et devra ensuite prouver que ses cartes correspondent bien à des identificateurs qu'il a choisis lors de la distribution des cartes. Il fera de même pour celles qu'il a écartées. Afin de convaincre les autres que les cartes qu'ils prétendent avoir sont bien les leurs, les joueurs vont suivre le protocole de détection des tricheurs.

4.5. Détection des tricheurs.

Afin de s'assurer que personne n'a triché, il est nécessaire de vérifier que les opérations de lecture de bits ont été effectuées correctement. C'est-à-dire que chacun n'a lu que des bits auxquels il avait droit. Il faut donc concevoir une preuve démontrant que chaque opération a bel et bien servi à lire le bit pour lequel on prétend l'avoir utilisée. Supposons qu'un joueur J_i affirme avoir posé la question Q_l à J_d dans le but de lire le bit $b_{d,l,k}$. Supposons également qu'en réalité cette question avait pour but de lire un bit f qui constitue une fonction des bits que lui présente J_d afin d'obtenir de l'information sur d'autres valeurs que $b_{d,l,k}$. Montrons qu'alors J_i sera pris au piège si on lui demande de révéler w_l et m_l tels que $Q_l = \hat{b}_{d,l,k} w_l^2 y_d^{m_l}$ **mod** n_d et que tous pourront se rendre compte qu'il a triché.

Théorème 6:

Si la réponse à Q_l contient de l'information sur d'autres bits que $b_{d,l,k}$ alors J_i ne pourra pas trouver w_l et m_l tels que $Q_l = \hat{b}_{d,l,k} w_l^2 y_d^{m_l}$ **mod** n_d.

Preuve:

Supposons au contraire que J_i puisse trouver avec une probabilité non-négligeable une valeur f dépendant d'autres bits que $b_{d,l,k}$ et que la réponse à Q_l donne de l'information sur la valeur de f. Si en plus J_i peut trouver w_l et m_l tels que définis alors il apprendra en plus $b_{d,l,k}$ car $\hat{b}_{d,l,k} = Q_l w_l^{-2} y_d^{-m_l}$ **mod** n_d.

Dans ce cas J_i aurait pu trouver de l'information sur la valeur $b_{d,l,k}$ lui-même, sans l'intervention de J_d. Voici comment il aurait pu simuler les actions de J_d.

•J_i génère $b'_{u,v}$ pour $1 \leq u \leq 6$ et $1 \leq v \leq 52$ et des attestations correspondantes $\hat{b}'_{u,v}$ en se servant de y_d.

•Il remplace $\hat{b}'_{l,k}$ par $\hat{b}_{d,l,k}$.

•Il forme une question Q_l comme il l'aurait fait avec J_d.

Avec la même probabilité que précédemment, J_i peut trouver f', dépendant d'autres bits parmi les $b'_{u,v}$, telle que la réponse à Q_l donne de l'information sur la valeur de f'. Comme précédemment on présume qu'il puisse trouver w_l et m_l tels que définis. Il faut noter que J_i connaît tous les $b'_{u,v}$. Il peut donc calculer f' exactement. De ce fait il peut apprendre lequel de $Q_l \in RQ_n$ ou $Q_l \in NRQ_n[+1]$ est le plus probable, car f' depend de Q_l. Finalement, puisqu'en plus J_i connaît m_l, il lui est possible de déterminer la valeur la plus probable de $b_{d,l,k}$, sans l'aide de J_d. Ceci contredit HRQ.

C.Q.F.D.

En conclusion, si un joueur veut prouver aux autres qu'il a lu un bit $b_{d,l,k}$, il n'a qu'à exhiber w_l et m_l tels que $Q_l = \hat{b}_{d,l,k} w_l^2 y_d^{m_l}$ **mod** n_d et ceci constitue une preuve indiscutable qu'il a lu ce bit et rien d'autre avec la question Q_l. Si par contre, il n'est pas capable de révéler ces valeurs alors on peut présumer qu'il a triché. Malheureusement ces preuves ont l'inconvénient de forcer chaque joueur à révéler toutes ses cartes, ce qui ne permet aucune confidentialité des stratégies et décisions prises par les joueurs.

Si on permettait aux joueurs de ne pas prouver qu'ils ont respecté la technique proposée dans le PROTOCOLE 3 et qu'ils l'ont utilisée pour obtenir de l'information auquel ils avaient droit, alors il leur serait facile de tricher. Par exemple si le jeu se joue à trois joueurs, supposons que J_1 est honnête, et que J_2 et J_3 collaborent afin de tricher J_1 et le détrousser. A chaque partie J_2 pourrait lire les cartes de J_1 et les dévoiler à J_3 et ensuite se retirer sous prétexte qu'il n'a pas un bon jeu. Ensuite J_3 serait en mesure de décider avant les mises si son jeu est meilleur que celui de J_1 ou non, et ainsi profiter de cette information. Il est donc essentiel pour que ce protocole fonctionne que les joueurs prouvent leur honnêteté.

Une autre façon de tricher serait pour le $N^{\text{ième}}$ joueur d'utiliser autre chose qu'une permutation pour π_N, par exemple une permutation dans laquelle la valeur correspondant à l'as de pique serait remplacée par une copie de la valeur du 2 de coeur. Dans ce cas, si personne ne se plaint que deux 2 de coeur sont en circulation, il ne serait pas pris et il saura pertinemment qu'aucun as de pique ne peut être obtenu. Ce simple fait peut constituer une information précieuse. On demandera donc aussi aux joueurs de donner toutes les valeurs $x_{i,j,k}$ (d'ouvrir $\hat{b}_{i,j,k}$) utilisées pour l'attestation de leur permutation afin d'en vérifier la validité. Du même coup, ces $x_{i,j,k}$ permettront de constater que chacun a répondu correctement aux questions qu'on lui a posées. C'est-à-dire que si J_i pose la question $Q_l = \hat{b}_{d,l,k} w_l^2 y_d^{m_l}$ **mod** n_d à J_d, alors J_d aurait dû répondre un bit r tel que $r = m_l \oplus b_{d,l,k}$. Autrement, il a triché et ceci sera mis en évidence quand $x_{d,l,k}$ sera dévoilé.

Il faut néanmoins prendre soin de ne pas révéler les $x_{d,l,k}$ trop tôt. Ils ne doivent être divulgués qu'après les w_l et m_l mentionnés plus tôt. Car autrement il serait très facile aux tricheurs connaissant $x_{d,l,k}$ de calculer les w_l et m_l qui leur manquent. Finalement, chaque joueur révèlera p_i et q_i afin de permettre à ses adversaires de constater la validité de n_i et y_i.

Malgré tout il se peut, si plusieurs joueurs trichent en même temps, que certains ne soient pas détectés mais ceci est inévitable. Prenons par exemple le cas où J_i et J_{i+1} ont triché tous les deux et que J_i refuse de collaborer à la fin et est décrété tricheur pour son manque de coopération, alors J_i peut communiquer des informations à J_{i+1} pouvant le sortir du pétrin et lui permettre de prouver son "innocence".

Donc en résumé, les étapes à suivre sont les suivantes:

PROTOCOLE 4 détection des tricheurs

(1) POUR chaque $i \in \{1,...,N\}$ FAIRE
(1.1) J_i FAIT
(1.1.1) POUR chaque $k \in \{1,2,...,52\}$ lui appartenant FAIRE
(1.1.1.1) $c \leftarrow \pi_i \pi_{i-1} \cdots \pi_1(k)$.
(1.1.1.2) dévoiler publiquement c.
(1.1.1.3) POUR $d \leftarrow i+1$ JUSQU'A N FAIRE
(1.1.1.3.1) $c \leftarrow \pi_d(c)$.
(1.1.1.3.2) POUR chaque $l \in \{1,...,6\}$ FAIRE
(1.1.1.3.2.1) dévoiler w_l et m_l tels que $Q_l = \hat{b}_{d,l,c} w_l^2 y_d^{m_l}$ **mod** n_d
où Q_l est la question qui a été utilisée pour lire $b_{d,l,c}$.
(1.1.1.3.3) FIN
(1.1.1.4) FIN
(1.1.2) FIN
(1.2) FIN
(2) FIN
(3) POUR chaque $i \in \{1,...,N\}$ FAIRE
(3.1) J_i FAIT
(3.1.3) POUR chaque $k \in \{1,2,...,52\}$ FAIRE
(3.1.3.1) dévoiler publiquement $\pi_i(k)$.
(3.1.3.2) POUR chaque $j \in \{1,...,6\}$ FAIRE
(3.1.3.2.1) dévoiler publiquement $x_{i,j,k}$ (ouvrir $\hat{b}_{i,j,k}$)
(3.1.3.3) FIN
(3.1.4) FIN
(3.1.5) révéler p_i et q_i
(3.2) FIN
(4) FIN

4.6. Sécurité contre les coalitions

Malgré certains inconvénients déjà évidents de ce protocole, celui-ci franchit quand même une étape importante dans le sens de la sécurité contre les coalitions. Cette sécurité provient du fait que chaque joueur est impliqué personnellement dans le brassage des cartes. Tout joueur parfaitemet honnête peut être certain que personne ne va, sans se compromettre, obtenir d'information sur son jeu ou sur les cartes restées dans le paquet. Même si certains ou tous ses opposants s'échangent des informations à son insu, ils ne peuvent rien découvrir de plus que leurs jeux mis en commun. Bien entendu, ceci est le mieux que l'on puisse espérer.

Par contre, former une coalition comporte un certain risque! Supposons que J_i réussisse à convaincre tous ses adversaires de former une coalition avec lui sans qu'ils ne sachent mutuellement qu'ils se sont tous alliés au même. Alors J_i aura accès à suffisamment d'information pour déterminer exactement tous les jeux et les cartes dans le paquet. Pire encore, si chacun de ses alliés lui dit sa permutation, alors il peut même choisir les cartes qu'il aura dans son jeu ! En conclusion, si quelqu'un vous offre de former une coalition, soyez certain que vous pouvez lui faire confiance. Aussi, méfiez-vous d'un allié qui vous demanderait votre permutation.

4.7. Lacunes à combler.

Pour terminer ce chapitre, voici en résumé les choses qu'il faudra mettre au point dans les prochains chapitres, afin d'arriver au but ultime: un protocole permettant de cacher complètement la stratégie. Premièrement, il faudra mettre au point une technique permettant de prouver que les joueurs font usage d'une permutation sans avoir à la révéler. Il faudra aussi pouvoir convaincre ses opposants que les valeurs lues par un joueur lui revenaient. Finalement, il sera essentiel de réaliser une technique permettant de prouver que les réponses aux questions posées par les joueurs étaient correctes, et ceci sans dévoiler les valeurs $x_{i,j,k}$ utilisées pour attester leurs permutations secrètes.

5. Un protocole complet

5.1. Sous-problèmes de base

Le premier sous-problème auquel nous nous intéresserons est celui de prouver à ses adversaires que ce qu'un joueur prétend être une permutation en est vraiment une. De façon similaire au protocole du chapitre 4, dans le protocole final, chaque joueur devra générer une permutation de {1,2,...,52} afin de déterminer le brassage des cartes. Aussi, afin de fixer définitivement cette permutation, il devra l'attester de la même façon que précédemment. Par contre, ces permutations ne seront jamais révélées "en clair" aux autres joueurs. Il serait donc possible qu'un joueur utilise autre chose qu'une permutation, alors qu'il prétends en utiliser une.

Comme au chapitre 4, chaque joueur J_i va d'abord générer p_i, q_i, n_i et y_i, puis va rendre publiques ces deux dernières. Il générera ensuite π_i, une permutation de {1,2,...,52}, et l'attestera en présentant à ses adversaires les $\hat{b}_{i,j,k}=x_{i,j,k}^2 y_i^{b_{i,j,k}}$ **mod** n_i pour $x_{i,j,k}$ choisi au hasard dans $\mathbb{Z}_{n_i}^*$ et $b_{i,j,k}$ le $j^{\text{ième}}$ bit du nombre $\pi_i(k)$.

Maintenant J_i devra prouver que les $\hat{b}_{i,j,k}$ attestent une permutation de {1,2,...,52}.

5.1.1. Preuve de la validité d'une permutation.

Pour résoudre ce problème, nous allons étudier le problème plus général de prouver que deux chaînes attestées α et β sont mutuellement une permutation l'une de l'autre. Supposons qu'Alice possède α et β, respectivement attestées par $\hat{\alpha}_i=x_i^2 y^{\alpha_i}$ **mod** n et $\hat{\beta}_i=x'^2_i y^{\beta_i}$ **mod** n pour des x_i et x'_i choisis au hasard dans $\mathbb{Z}_n^*$,

α_i étant le $j^{\text{ième}}$ bit de α et β_i étant le $j^{\text{ième}}$ bit de β. Soit l la longueur de α et de β. Alice voudrait prouver à Bob que les chaînes attestées par les $\hat{\alpha}_i$ et $\hat{\beta}_i$ sont constituées exactement des mêmes bits mais possiblement dans un ordre différent, sans pour autant révéler quelqu'autre information que ce soit sur α et β. Voici comment procéder pour en arriver à ce résultat.

Pour commencer, rappelons un fait énoncé à la section 3.3: le produit sur $\mathbb{Z}_n^*$ correspond au $\oplus$ sur $\{0,1\}$. Donc une première solution serait de permuter les bits de β pour qu'ils correspondent à ceux de α puis de montrer qu'ils sont identiques. Supposons que σ est une permutation de $\{1,2,...,l\}$ tel que $\alpha_i=\beta_{\sigma(i)}$ pour chaque $i \in \{1,2,...,l\}$. Pour prouver l'existence de σ sans donner de connaissances sur α ou β, il suffit de prouver que $\hat{\alpha}_i$ et $\hat{\beta}_{\sigma(i)}$ attestent le même bit. Par la propriété précédente, $\alpha_i=\beta_{\sigma(i)}$ si et seulement si $\hat{\alpha}_i \times \hat{\beta}_{\sigma(i)}$ est un résidu quadratique. Donc Alice n'a qu'à donner à Bob des nombres r_i tels que $r_i^2 \equiv \hat{\alpha}_i \times \hat{\beta}_{\sigma(i)} \ (mod\ n)$ afin qu'il puisse vérifier que α_i et $\beta_{\sigma(i)}$ sont identiques. Remarquons que si α et β ne sont pas mutuellement une permutation l'une de l'autre, alors Alice ne pourra jamais convaincre Bob du contraire avec ce protocole. Il n'est pas possible de tricher.

Cette méthode fonctionne très bien, mais elle suppose qu'Alice donne à Bob la permutation σ. Ceci constitue une relation entre α et β. C'est-à-dire qu'une certaine quantité d'information sur α et β a été donnée à Bob. Afin d'éliminer cette information supplémentaire, il faudrait réussir à ne pas donner σ, mais quand même convaincre Bob de son existence. Pour ceci, Alice et Bob vont d'abord se mettre d'accord sur un paramètre de sécurité s. Alice va convaincre Bob de l'existence de σ avec une probabilité de 2^{-s} de réussite si σ n'existe pas. Ceci provient du fait qu'une triche de la part d'Alice définie un sous-ensemble de $\{1,2,...,s\}$ et qu'elle pourra convaincre Bob à tords seulement si il choisit ce même sous ensemble par hasard.

Alice choisit s permutations $\rho_1,\rho_2,\cdots,\rho_s$ de $\{1,2,...,l\}$. Ensuite pour chaque $u \in \{1,2,...,s\}$, elle définit $\gamma_u(i)=\alpha_{\rho_u(i)}$ et atteste de ces valeurs par $\hat{\gamma}_u(i)$. Maintenant, Bob va choisir U, un des 2^s sous-ensembles possibles de $\{1,2,...,s\}$. Alice va prouver à Bob que γ_u est une permutation de α de la façon suggérée plus haut, pour chaque $u \in U$. De la même façon elle va montrer à Bob que γ_u est une permutation de β, pour les $u \notin U$. Remarquez que jamais Alice ne prouve à Bob que ρ_u est une permutation de α et de β, car ceci aurait pour effet de permettre à Bob de deviner σ.

Supposons que α et β ne soient pas mutuellement une permutation l'une de l'autre. La probabilité de succès du protocole précédent est alors de 2^{-s}, car ceci veut dire qu'Alice avait choisi γ_u comme permutation de α, pour chaque $u \in U$ et $\gamma_{u'}$

comme permutation de β, pour chaque $u' \notin U$. Sinon elle n'aurait pas pu prouver que les γ_u et les $\gamma_{u'}$ sont des permutations de α et de β, respectivement. Ceci voudrait dire qu'Alice avait choisi un sous-ensemble V de $\{1,2,...,s\}$ avant que Bob ne déclare U, et qu'elle avait décidé de fixer γ_v comme permutation de α ou de β selon que v soit dans V ou non. Ce n'est qu'avec probabilité 2^{-s} que Bob va choisir U identique à V et se laisser convaincre que β est une permutation des bits de α quand ce n'est pas le cas. Il existe, en effet, 2^s sous-ensembles possibles de $\{1,2,...,s\}$ et tous sauf V font faire que Bob va se rendre compte de la supercherie.

La même idée peut être étendue à des chaînes secrètes constituées d'éléments plus complexes que des bits de la façon suivante.

Soient μ et ν deux vecteurs de t chaînes de l bits. Dénotons $\mu_{i,j}$ (resp. $\nu_{i,j}$) le $j^{\text{ième}}$ bit de la $i^{\text{ième}}$ chaîne de μ (resp. ν). Supposons que $\hat{\mu}_{i,j}$ (resp. $\hat{\nu}_{i,j}$) $\in \mathbb{Z}_n^*$ soit une attestation de $\mu_{i,j}$ (resp. $\nu_{i,j}$). Alice voudrait prouver à Bob qu'il existe σ tel que $\mu_{i,j} = \nu_{\sigma(i),j}$ pour $1 \le j \le l$.

PROTOCOLE 5 Permutation

Alice:

(1) choisit $\rho_1, \rho_2, ..., \rho_s$ des permutations de $\{1,2,...,t\}$

(2) choisit aléatoirement $r_{i,j,k}$ afin de former $\varepsilon_{i,j,k} = r_{i,j,k}^2 \hat{\mu}_{\rho_k(i),j}$ **mod** n, pour $1 \le i \le t$, $1 \le j \le l$ et $1 \le k \le s$.

(3) rend publique $\varepsilon_{i,j,k}$, pour $1 \le i \le t$, $1 \le j \le l$ et $1 \le k \le s$.

Bob:

(4) choisit $U \subseteq \{1,2,...,s\}$ et l'annonce à Alice.

Alice:

(5) POUR chaque $u \in U$ révèle ρ_u et pour $1 \le i \le t$, $1 \le j \le l$
$r'_{i,j,u}$ tels que $r'^{2}_{i,j,u} \hat{\mu}_{\rho_u(i),j} \equiv \varepsilon_{i,j,u} \pmod{n}$

(6) POUR chaque $u \notin U$ révèle $\rho_u(\sigma)$ et pour $1 \le i \le t$, $1 \le j \le l$
$r'_{i,j,u}$ tels que $r'^{2}_{i,j,u} \hat{\nu}_{\rho_u(\sigma(i)),j} \equiv \varepsilon_{i,j,u} \pmod{n}$

Nous verrons plus tard que ce protocole est un élément central dans la construction du protocole global. Il sera utilisé à plusieurs reprises et dans plusieurs situations. Dans le cas qui nous intéressait jusqu'à présent, il suffit de choisir $\hat{\beta}$ telle que $\beta_i = i$ et d'utiliser le protocole précédent pour prouver que α et β sont mutuellement une permutation l'une de l'autre, puis d'ouvrir β, afin de montrer que $\beta_i = i$ pour $i \in \{1,2,...,52\}$.

Si pour une raison ou une autre Bob n'est pas une personne mais plusieurs, (comme ce l'est dans notre cas car J_i veut convaincre tous ses adversaires) alors il faut que l'intervention de Bob à l'étape (4) soit le résultat d'une décision commune des personnes impliquées. Si elles ne se font pas confiance, il leur est toujours possible de se distribuer la décision. Par exemple dans notre cas, les $N-1$ joueurs identifiés par Bob doivent choisir $U \subseteq \{1,2,...,s\}$, mais il est possible qu'un seul d'entre eux soit honnête et que les autres soient des complices de J_i et que tous ensemble tentent de convaincre le seul joueur honnête de la validité de la permutation de J_i. Bien entendu, la solution directe à ce problème est de répéter le protocole avec chaque joueur, à tour de rôle. Mais il est plus économique de faire le protocole une seule fois, mais en prenant soin de tirer U au hasard plutôt que de le choisir. Ceci ne change en rien la probabilité de ne pas détecter que J_i triche. Pour tirer U au hasard, il suffit de tirer s bits aléatoires $b_1, b_2, \cdots, b_s$ et de mettre i dans U si et seulement si $b_i = 1$. Pour tirer des bits au hasard à plusieurs, il suffit de généraliser le protocole de tirage au sort de la section 3.3 de la façon suivante.

PROTOCOLE 5.1 Tirage au hasard d'un bit

(1) POUR chaque $i \in \{1,2,...,N\}$

(1.1) J_i choisit aléatoirement $B_i \in \{0,1\}$

(1.2) J_i choisit aléatoirement $r_i \in \mathbb{Z}_{n_i}^*$

(1.3) J_i annonce publiquement $r_i^2 y_i^{B_i}$ **mod** n_i

(2) FIN

(3) POUR chaque $i \in \{1,2,...,N\}$

(3.1) J_i révèle B_i et r_i

(4) FIN

(5) $bit \leftarrow \bigoplus_{i=1}^{N} B_i$

5.1.2. Validité des paramètres n et y

On peut remarquer qu'à la section précédente on suppose que J_i choisit honnêtement les paramètres n_i et y_i. Qu'advient-il si n_i n'a pas la forme appropriée ou si y_i n'est pas un non-résidu quadratique modulo n_i ? Il est peu probable que l'on puisse étudier toutes les conséquences possibles de cette triche. Notons néanmoins qu'un danger potentiel se cache derrière ce geste (c'est pourquoi il devra être évité). Nous avons donc intérêt à prévoir un mécanisme permettant de prouver que n_i a exactement deux facteurs premiers, d'une part, et un autre capable de prouver que y_i est un non-résidu quadratique modulo n_i, d'autre part. Et ce, sans compromettre la sécurité du système d'attestation obtenu de n_i et y_i. En particulier, l'application de ce mécanisme ne doit pas faciliter une éventuelle tentative de factoriser n_i. Ce problème a été étudié et partiellement résolu dans [GHY] et [GMRa].

5.1.2.1. Preuve que $y \in NRQ_n$

Supposons qu'Alice veuille prouver à Bob que y est un non-résidu quadratique modulo n. Pour se faire convaincre, Bob choisit aléatoirement un nombre r, $1<r<n$. Ensuite, Bob demande aléatoirement à Alice de lui révéler la résiduosité quadratique de $x=r^2 \textbf{ mod } n$ ou de $x=r^2y \textbf{ mod } n$. Bob sait parfaitement la réponse qu'Alice devrait lui donner, puisque le premier est un résidu quadratique et que le second n'en est pas un, à condition que y soit lui-même un non-résidu quadratique.

Si y est un non-résidu quadratique alors Alice peut distinguer entre les deux choix que Bob peut utiliser car elle sait décider de la résiduosité quadratique modulo n. Dans ce cas, elle donnera sûrement à Bob la réponse qu'il attend. Par contre, si y est un résidu quadratique alors Alice ne pourra pas décider si x est de la forme r^2 ou r^2y car les deux sont des résidus quadratiques. Donc Alice a une chance sur 2 de se tromper en répondant à Bob car elle ne sait pas la vraie réponse. Maintenant, si Bob est satisfait de la réponse qu'Alice lui a fournie, il lui révèle r afin de lui prouver qu'il savait la réponse à l'avance. Cette étape est nécessaire car sinon Bob pourrait utiliser ce protocole pour apprendre la résiduosité quadratique de nombres de son choix, ce qui constitue un danger potentiel pour Alice. Par ailleurs, si Alice donne une mauvaise réponse alors Bob arrête la discussion car Alice a triché.

Si ce protocole est répété s fois alors Alice convaincra Bob que y est un non-résidu quadratique seulement avec probabilitité 2^{-s} si y ne l'est pas.

PROTOCOLE 6 NON-Résidu(y)

(1) POUR $k \in \{1,2,...,s\}$
(1.1) J_j choisit aléatoirement $r_k \in \mathbb{Z}_{n_i}^*$ et $b_k \in \{0,1\}$.
(1.2) J_j demande à J_i de lui révéler la résiduosité quadratique de $r_k^2 y_i^{b_k}$.
(1.3) J_i répond à la demande de J_j
(1.4) SI J_j est satisfait de la réponse de J_i
(1.4.1) ALORS J_j dévoile r_k
(1.4.2) SINON J_j arrête le protocole car J_i triche
(1.5) FIN

Evidemment, ce protocole ne peut nullement compromettre la sécurité du système d'attestation d'Alice car Bob connaît d'avance les réponses qu'elle va lui donner. C'est-à-dire qu'il n'apprend rien qu'il ne peut calculer lui-même! Donc Bob ne gagne aucun avantage pouvant réduire la sécurité des données d'Alice.

5.1.2.2. Comment prouver que $n = p^a q^b$

Jusqu'ici on a toujours présumé que n avait la forme $n = pq$. Il est possible de démontrer que toutes les propriétés arithmétiques qui nous intéressent dans les nombres de cette forme, sont aussi valides si $n = p^a q^b$ avec a ou b impair (ou les deux, sous-entendu). Etant donné que les nombres de la forme $p^a q^b$ peuvent être plus faciles à factoriser que ceux de la forme pq, personne n'a avantage à choisir $n = p^a q^b$ car il risque seulement de réduire la sécurité de sa méthode d'attestation. Il suffit donc de démontrer que n a la forme $p^a q^b$ avec a ou b impair.

Si Bob veut vérifier que le n d'Alice a la forme cherchée, il vérifie d'abord que si $n = p^a q^b$ alors a ou b est impair. Pour ceci il suffit de constater que n n'est pas un carré parfait dans $\mathbb{N}$. Pour vérifier ceci, Bob calcule $\sqrt{n}$ et constate que ce n'est pas un entier. Ensuite, il lui faut vérifier que n a exactement deux facteurs premiers. Il peut d'abord se convaincre que n n'est pas premier en utilisant un test probabiliste de composition tel que ceux de [Ra1] ou [SS]. Il reste à montrer que n a au plus deux facteurs premiers. Le théorème suivant est la base de la preuve qu'Alice va fournir à Bob:

Soit $\nu(x)$ le nombre de facteurs premiers de x.

Théorème 7 [GHY]:

si $\nu(n)=2$ alors $\#RQ_n = \#\mathbb{Z}_n^*[+1]/2$

et si $\nu(n)>2$ alors $\#RQ_n \leq \#\mathbb{Z}_n^*[+1]/4$

Il en découle également que pour $x_1,x_2,\ldots,x_s \in \mathbb{Z}_n^*[+1]$ choisis aléatoirement et $R_s = \{x_i \mid 1 \leq i \leq s \text{ et } x_i \in RQ_n\}$:

Théorème 8 [GHY]: il existe $k \in \mathbb{R}, k>0$ tel que

si $\nu(n)>2$ alors $prob(\#R_s \geq \frac{3s}{8}) \leq 2^{-ks}$

et si $\nu(n)=2$ alors $prob(\#R_s < \frac{3s}{8}) \leq 2^{-ks}$

Pour convaincre Bob que $\nu(n)=2$, Alice va montrer que parmi s nombres aléatoires $x_1,x_2,\ldots,x_s \in \mathbb{Z}_n^*[+1]$, au moins $\frac{3s}{8}$ de ceux-ci sont des résidus quadratiques modulo n. Par contre, pour ne pas donner d'avantage à Bob, Alice ne va sûrement pas lui laisser choisir les x_i car Bob pourrait alors connaître la résiduosité quadratique de s nombres de son choix. Evidemment, si Bob laissait Alice choisir les x_i, elle pourrait en choisir au moins $\frac{3s}{8}$ qui soient des carrés modulo n, même si $\nu(n)>2$. Afin de produire les x_i aléatoirement sans que ni Bob, ni Alice ne puissent les déterminer à l'avance, il faudra qu'ils les forment conjointement. Pour ceci, ils n'auront qu'à tirer $lg\ n$ bits aléatoires avec le protocole de la section 5.1 et à considérer la concaténation des bits obtenus comme un nombre exprimé en binaire. Si le nombre obtenu est dans $\mathbb{Z}_n^*[+1]$ il pourra être utilisé comme l'un des x_i. Heureusement, ceci se produit avec probabilité au moins $\frac{1}{2}$. Quand Alice veut montrer à Bob qu'un x_i particulier est un résidu quadratique, elle n'a qu'à exhiber r_i tel que $r_i^2 \equiv x_i \pmod{n}$. Le protocole suivant résume cette méthode:

PROTOCOLE 7 Bonne Forme(n)

(0) $total \leftarrow 0$
(1) POUR $k \in \{1,2,...,s\}$
(1.1) REPETER
(1.1.1) POUR $u \in \{1,2,...,\lceil lg\ n_i \rceil\}$
(1.1.1.1) les joueurs choisissent aléatoirement b_u (PROTOCOLE 5.1)
(1.1.2) FIN
(1.1.3) $r_k \leftarrow \sum_{u=1}^{\lceil lg\ n_i \rceil} b_u 2^{u-1}$
(1.2) JUSQUE $\left[\frac{r_k}{n_i}\right] = +1$
(1.3) SI $r_k \in RQ_{n_i}$ ALORS
(1.3.1) J_i exhibe a_k tel que $r_k \equiv a_k^2 \ (mod\ n_i)$
(1.3.2) $total \leftarrow total + 1$
(2) FIN
(3) SI $total \geq \frac{3s}{8}$ ALORS accepter SINON refuser

5.1.3. Initialisation

Voici donc le protocole que les joueurs doivent suivre afin de préparer la partie:

PROTOCOLE 8 Préparation (partie A)
(1) POUR $i \in \{1,2,...,N\}$
(1.1) J_i FAIT
(1.1.1) choisit deux grands nombres premiers de même taille: p_i et q_i
(1.1.2) rend public $n_i = p_i q_i$
(1.1.3) prouve à tous que $\nu(n_i) = 2$ à l'aide du PROTOCOLE 7
(1.1.4) choisit $y_i \in NRQ_{n_i}[+1]$ aléatoirement
(1.1.5) POUR $j \in \{1,2,...,N\}$, $i \neq j$
(1.1.5.1) prouve à J_j que $y_i \in NRQ_{n_i}[+1]$ à l'aide du PROTOCOLE 6
(1.1.6) FIN
(1.2) FIN
(2) FIN

5.2. Méthode de lecture des cartes

Un joueur qui utilise le protocole 3.1 de lecture (section 4.2), peut tricher en lisant à la ligne 1.7.1 une valeur de $\pi_d(c')$ pour $c' \neq c$, qui ne lui appartient pas. Dans le protocole initial ce problème est résolu en forçant les joueurs à prouver à la fin de la partie qu'ils ont lu des entrées leur revenant. Par contre, pour cela, ils devaient dévoiler les entrées en question à leurs adversaires, c'est-à-dire leurs cartes, du même coup. Ce qui empêche la confidentialité de la stratégie. Afin de comprendre la nouvelle façon de résoudre ce problème sans avoir à révéler sa stratégie, nous allons d'abord identifier les actions que peut prendre un tricheur qui mettent en danger la sécurité de ses adversaires.

5.2.1. Triches tolérables

Identifions d'abord les actions dites tolérables que peut exécuter un tricheur. Une action est considérée tolérable si elle ne donne aucun avantage au joueur l'exécutant ou à un de ses associés dans le cas où plusieurs collaborent. En d'autres termes, un joueur pourra dévier de son protocole sans être détecté, à la seule condition qu'il ne puisse en tirer avantage. Evidemment, porter une telle action est inutile, particulièrement quand elle comporte des risques d'être détectée!

Une action de ce genre serait, pour un tricheur, de lire des entrées dans les permutations de ses adversaires lorqu'elles ne lui appartiennent pas. Ceci lui permettrait possiblement d'apprendre des cartes appartenant à ses adversaires. A première vue, cette action ne semble pas tolérable, mais nous verrons qu'avec le nouveau protocole elle le deviendra. Le premier inconvénient est que le nombre d'entrées auxquelles un joueur a droit est fixe et équivaut au nombre de cartes dont il dispose. Ce qui veut dire que pour poser un tel geste il devra sacrifier une de ses cartes, car il perdra, par cette action malhonnête, la seule chance qu'il avait de la lire. Néanmoins, s'il a l'intention de fermer son jeu à la fin du match, il pourra étudier la stratégie de ses adversaires à son gré.

Deuxièmement, s'il triche de façon anarchique et qu'il lit des entrées quelconques chez ses adversaires, alors il se peut qu'il n'apprenne rien du tout. Car la relation reliant les entrées de la permutation du joueur J_i à celles de J_{i+1} est bien plus importante que les valeurs elles-même. Par exemple, si un tricheur décide de lire systématiquement la première entrée de la permutation de chaque joueur, les informations obtenues seront inutiles car en général il sera incapable de dire qui possède la carte qu'il aura lue. En conclusion, si un joueur décide de tricher il devra le faire d'une façon très précise et nous nous intéresserons à empêcher ce type particulier d'attaque.

Nous verrons plus loin qu'un joueur J_i ne peut pas tricher dans les permutations des joueurs qui le précèdent ($J_k, 1 \leq k < i$) car les entrées qu'il y lit seront connues de tous. Evidemment, il ne peut "tricher" dans sa propre permutation. Donc si un joueur triche de cette façon, il doit le faire dans la permutation des joueurs le succédant ($J_k, i < k \leq N$).

Si par malchance, il triche et obtient une carte qui n'appartient à aucun autre joueur à la fin de la partie, alors nul avantage ne peut être tiré de cette carte car il aurait pu l'obtenir légalement avec la même probabilité, sans changer le jeu de ses adversaires. De ce fait, cette triche est tolorée, car elle n'est nuisible qu'à celui qui la pratique. Dans le cas qui nous intéresse, en plus, il perd la possibilité d'utiliser une de ses cartes.

Par contre, le fait de lire une carte appartenant à quelqu'un d'autre doit absolument être détecté. C'est justement pour cela qu'il a fallu modifier la représentation des cartes.

5.2.2. Nouvelle représentation des cartes

Précédemment, le brassage des cartes était obtenu par le choix de chaque joueur J_i d'une permutation π_i. Ceci reste vrai dans le second protocole, mais de plus, J_i devra choisir $i-1$ fonctions $\tau_{i,j}:\{1,2,\ldots,52\} \rightarrow \{0,1\}^s$, $1 \leq j < i$ (s étant un paramètre de sécurité à déterminer). Les chaînes de bits $\tau_{i,j}(1), \tau_{i,j}(2), \cdots, \tau_{i,j}(52)$ que l'on nomme les traces de π_i, permettront à J_i de vérifier l'honnêteté de J_j. On dénotera $\tau_{i,j}(f(x))$ par $\tau_{i,j}(f(x))$ afin de simplifer les expressions. Maintenant, pour lire une carte, correspondant à un identificateur k, J_i devra apprendre $\pi_1(k)$, $\pi_2\pi_1(k)$, ..., $\pi_N\pi_{N-1}\cdots\pi_1(k)$ comme précédemment, mais aussi $\tau_{i+1,i}\pi_i\pi_{i-1}\cdots\pi_1(k)$ $\tau_{i+2,i}\pi_{i+1}\pi_i\cdots\pi_1(k)$, ..., $\tau_{N,i}\pi_{N-1}\pi_{N-2}\cdots\pi_1(k)$. Nous verrons au chapitre suivant comment ces chaînes de bits vont être utilisées pour la détection des tricheurs.

Quoi qu'il en soit, il est d'une importance capitale qu'un joueur J_j obtienne précisément $\tau_{i,j}(c)$ en supplément à $\pi_i(c)$ quand il lit cette valeur dans la permutation de J_i, car $\tau_{i,j}(c)$ servira de preuve qu'il a lu une entrée $\pi_i(c)$ n'appartenant pas à J_i. Si J_j pouvait obtenir $\tau_{i,j}(c')$, pour $c' \neq c$ lors de la lecture de $\pi_i(c)$, alors il pourrait plus tard prouver illégalement qu'il n'a pas lu d'entrées appartenant à J_i. Par exemple, si $\pi_i(c')$ n'appartient pas à J_i et que $\pi_i(c)$ lui revient, alors J_j pourrait lire $\pi_i(c)$ et prouver qu'il n'a pas triché !!

Donc ce que nous voulons est un mécanisme de lecture qui permette de lier $\tau_{i,j}(c)$ à $\pi_i(c)$ pour chaque $c \in \{1,2,...,52\}$.

5.2.3. DISTOR

Supposons qu'Alice dispose de t secrets, chacun étant composé de L bits, et qu'elle consente à en divulger un et un seul à Bob. Dénotons $S_{i,j}$ le $j^{\text{ième}}$ bit du $i^{\text{ième}}$ secret d'Alice. Alice va pouvoir arriver à son but grâce à un protocole de divulgation de secrets tout ou rien (DISTOR) qu'elle utilisera avec Bob. Voici un tel protocole.

Intuitivement, Alice et Bob procéderont comme suit:

- Alice va attester ses secrets.
- Bob va générer des "questions", chacune étant destinée à un secret d'Alice.
- Alice peut répondre à une question sans pouvoir comprendre à quel secret cette question fait référence.
- Bob peut déduire la valeur d'un secret à partir de la réponse d'Alice à la question correspondante.
- Bob va prouver à Alice que ses questions sont honnêtes (que chacune des questions s'adresse à un et un seul secret) en produisant s séries permutées de questions équivalentes et en fournissant pour chacune des séries, au choix d'Alice, soit une preuve de l'équivalence de la série aux questions initiales, soit une démonstration de l'honnêteté de la série.

Voici comment ce mécanisme est implanté:

Pour commencer Alice choisit une méthode d'attestation tel que décrit précédemment. Supposons que n et y soient ses paramètres publics. Alice prouve à Bob que ses paramètres sont valides, comme à la section 5.1.2. Enfin, Alice atteste les bits de ses secrets par des $\hat{S}_{i,j} = r_{i,j}^2 y^{S_{i,j}}$ **mod** n avec $r_{i,j}$ choisi au hasard dans $\mathbb{Z}_n^*$.

Bob, pour sa part, va maintenant construire t séries de L questions, chaque série de questions devant s'adresser à un secret particulier d'Alice. Bob choisit une permutation ρ de $\{1,2,...,t\}$ et construit ses questions $Q_{i,j}=\hat{S}_{\rho(i),j}m_{i,j}$ avec des $m_{i,j}$ aléatoires dans $\mathbb{Z}_n^*[+1]$ mais dont Bob connaît la résiduosité quadratique. Pour obtenir de tels $m_{i,j}$ il peut choisir $a_{i,j}$ uniformément dans $\mathbb{Z}_n^*$ et $b_{i,j}$ dans $\{0,1\}$ pour former $m_{i,j}=a_{i,j}^2 y^{b_{i,j}}$ qui sera un nombre choisi uniformément dans $\mathbb{Z}_n^*[+1]$ et dont la résiduosité quadratique est déterminée par $b_{i,j}$. Bob annonce ses questions à Alice.

Remarquons que le fait que les $m_{i,j}$ soient issus d'une distribution uniforme implique que les $Q_{i,j}$ seront aussi uniformément distribués. C'est-à-dire qu'Alice est incapable de déterminer auquel de ses secrets est destinée la question $Q_{i,j}$. Par ailleurs, si Alice révèle à Bob la résiduosité quadratique de $Q_{i,j}$ alors Bob peut en déduire celle de $\hat{S}_{\rho(i),j}$ car il sait celle de $m_{i,j}$. Donc pour donner un de ses secrets, Alice n'a qu'à révéler le caractère quadratique de tous les $Q_{i,j}$ pour un i choisi par Bob.

Jusqu'ici rien ne garantit à Alice que Bob a vraiment formé ses séries de questions correctement. Il pourrait fort bien avoir des questions s'adressant à des secrets distincts dans la même série, ce qui est justement ce que nous voulons éviter. Donc Bob doit prouver son honnêteté à Alice. Pour ceci il choisit s permutations $\rho_1,\rho_2,\ldots,\rho_s$ de $\{1,2,...,t\}$. Il produit s nouveaux groupes de t séries de L questions, comme précédemment. Par contre, cette fois, il forme les questions $Q_{i',j,k}=\hat{S}_{\rho_k(i),j}m_{i',j,k}$ pour $1\le i\le t, 1\le j\le L, 1\le k\le s$. Bob annonce les $t\times L\times s$ questions à Alice.

En retour, Alice choisit $U\subseteq\{1,2,...,s\}$ et le dit à Bob. Maintenant Bob va montrer que pour $u\notin U$ les questions $Q_{i',j,u}$ forment un groupe correct. Afin de prouver cette affirmation, Bob révèle ρ_u ainsi que des $a_{i',j,u}$ et $b_{i',j,u}$ tels que $m_{i',j,u}=a_{i',j,u}^2 y^{b_{i',j,u}}$, $\forall i\forall j$. Alice peut ensuite vérifier que $Q_{i',j,u}=\hat{S}_{\rho_u(i),j}a_{i',j,u}^2 y^{b_{i',j,u}}$ et ainsi constater que chaque série de questions correspondent bien à un et un seul secret. Ensuite, pour $u\in U$, Bob révèle $\rho_u(\rho^{-1})$ ainsi que des $a_{i'',j,u}$ et $b_{i'',j,u}$ tels que $m_{i',j,u}m_{\rho_u(\rho^{-1}(i)),j}=a_{i'',j,u}^2 y^{b_{i'',j,u}}$. Alice peut ensuite vérifier que $Q_{i',j,u}=Q_{\rho_u(\rho^{-1}(i)),j}a_{i'',j,u}^2 y^{b_{i'',j,u}}$ et ainsi constater que chaque série de questions du groupe u correspondent bien à une et une seule série de questions du groupe initial.

PROTOCOLE 9 Préparation de DISTOR

Bob:

(1) choisit ρ une permutation de $\{1,2,...,t\}$

(2) POUR $1 \le i \le t$, $1 \le j \le L$

(2.1) choisit $a_{i,j} \in \mathbb{Z}_n^*$ et $b_{i,j}$ aléatoirement

(2.2) $Q_{i,j} \leftarrow \hat{S}_{\rho(i),j}\, a_{i,j}^2 y^{b_{i,j}}$ **mod** n

(2.3) annonce $Q_{i,j}$ à Alice

(3) FIN

(4) POUR $1 \le k \le s$

(4.1) choisit ρ_k une permutation de $\{1,2,...,t\}$

(4.2) POUR $1 \le i \le t$, $1 \le j \le L$

(4.2.1) choisit $a'_{i,j,k} \in \mathbb{Z}_n^*$ et $b'_{i,j,k}$ aléatoirement

(4.2.2) $Q'_{i,j,k} \leftarrow \hat{S}_{\rho_k(i),j}\, a'^2_{i,j,k} y^{b'_{i,j,k}}$ **mod** n

(4.2.3) annonce $Q'_{i,j,k}$ à Alice

(4.3) FIN

(5) FIN

Alice:

(6) choisit $U \subseteq \{1,2,...,s\}$ et l'annonce à Bob

Bob:

(7) POUR chaque $u \notin U$

(7.1) révèle ρ_u ainsi que $a'_{i,j,u}$ et $b'_{i,j,u}$ $,1 \le i \le t, 1 \le j \le L$

(8) FIN

(9) POUR chaque $u \in U$

(9.1) révèle $\rho_u(\rho^{-1})$ ainsi que $a''_{i,j}$ et $b''_{i,j}$ $,1 \le i \le t, 1 \le j \le L$

tels que $Q_{i,j} = Q'_{\rho_u(\rho^{-1}(i)),j,u}\, a''^2_{i,j} y^{b''_{i,j}}$ **mod** n

(10) FIN

De cet échange, Alice pourra déduire que les questions $Q_{i,j}$ de Bob sont honnêtes car il n'avait qu'une chance sur 2^s de réussir le protocole précédent s'il avait triché. La seule chance que Bob aurait de tricher serait qu'il choisisse les $Q_{i,j}$ de façon malhonnête ainsi que les $Q'_{i,j,v}$, $v \in V$ pour un $V \subseteq \{1,2,...,s\}$ de son choix et que les questions restantes soient honnêtes: $Q_{i',j,v}$, $v \notin V$. Puis qu'Alice choisisse exactement $U=V$, ce qui se produit avec probabilité 2^{-s}.

Maintenant à chaque fois que Bob veut obtenir le secret $\rho(i)$ d'Alice, il lui indique $i \in \{1,2,...,t\}$, ce à quoi Alice répond, en lui dévoilant la résiduosité quadratique de ses $Q_{i,j}$, $1 \le j \le L$. Finalement pour que Bob soit convaincu qu'Alice lui donne des réponses correctes, Alice va lui en donner la preuve. Si elle prétend que $Q_{i,j}$ est un résidu quadratique, elle lui dévoile R tel que $Q_{i,j} \equiv R^2 \ (mod\ n)$ et tel que $Q_{i,j}y \equiv R^2 \ (mod\ n)$ dans le cas contraire.

PROTOCOLE 10 Lecture d'un secret (DISTOR)

Bob:

(1) choisit $i \in \{1,2,...,t\}$.

(2) annonce $\rho^{-1}(i)$ à Alice.

(3) POUR $1 \le j \le L$

Alice:

(3.1) $\beta_j \leftarrow \begin{cases} 0 \text{ si } Q_{\rho^{-1}(i),j} \text{ est un résidu quadratique} \\ 1 \text{ sinon.} \end{cases}$

(3.2) SI $\beta_j = 0$

(3.2.1) ALORS trouve r_j tel que $r_j^2 \equiv Q_{\rho^{-1}(i),j} \ (mod\ n)$

(3.2.2) SINON trouve r_j tel que $r_j^2 \equiv Q_{\rho^{-1}(i),j}\,y \ (mod\ n)$.

(3.3) retourne β_j et r_j à Bob.

Bob:

(3.4) $b_{i,j} \leftarrow \beta_j \oplus m_{\rho^{-1}(i),j}$

(4) FIN

5.2.4. Application de DISTOR au poker

Dans le cas qui nous intéresse, DISTOR sera utilisé entre chaque couple de joueurs (J_i, J_j) avec $i > j$. J_i prendra le rôle d'Alice et J_j celui de Bob. DISTOR sera utilisé avec les 52 secrets $< \pi_i(1), \tau_{i,j}(1) >$, $< \pi_i(2), \tau_{i,j}(2) >$, ..., $< \pi_i(52), \tau_{i,j}(52) >$. Chacun des J_j, $j < i$ possèdera à la fin du protocole de préparation de DISTOR, un groupe de 52 séries de $6+s$ questions qui seront prouvées correctes à J_i.

PROTOCOLE 11 Préparation (partie B)

(1) POUR $i \in \{1,2,...,N\}$
(1.1) J_i choisit π_i une permutation de $\{1,2,...,52\}$
(1.2) J_i atteste π_i par $\hat{\pi}_i$
(1.3) J_i utilise le PROTOCOLE 5 et prouve
à ses adversaires que π_i est une permutation de $\{1,2,...,52\}$
(1.4) POUR $1 \le j < i$
(1.4.1) J_i choisit $\tau_{i,j} : \{1,2,...,52\} \to \{0,1\}^s$
(1.4.2) J_i atteste $\tau_{i,j}(u)$ par $\hat{\tau}_{i,j,u}$ pour $1 \le u \le 52$
(1.4.3) utiliser Preparation de DISTOR avec Alice=J_i, Bob=J_j,
$\hat{S}_{u,v} = \hat{\pi}_{i,u,v}$ et $\hat{S}_{u,w+6} = \hat{\tau}_{i,j,u,w}$, pour $1 \le u \le 52$, $1 \le v \le 6$ et $1 \le w \le s$
(1.5) FIN
(2) FIN

Au début, chaque $k \in \{1,2,...,52\}$ est marquée libre. Pour lire une carte, J_i va d'abord choisir un k libre. Puis J_i demande successivement à J_1, J_2, ...,J_{i-1} les valeurs $\pi_1(k)$, $\pi_2\pi_1(k)$, ..., $\pi_{i-1} \cdots \pi_2\pi_1(k)$, ce qu'ils font en ouvrant $\hat{\pi}_1(k)$, $\hat{\pi}_2(\pi_1(k))$, ..., $\hat{\pi}_{i-1}(\cdots \pi_2\pi_1(k))$ afin de prouver leur réponse. Ensuite J_i calcule $\pi_i\pi_{i-1} \cdots \pi_2\pi_1(k)$ secrètement. Enfin, il se procure les valeurs manquantes $\pi_{i+1}\pi_i \cdots \pi_2\pi_1(k)$, $\pi_{i+2}\pi_{i+1} \cdots \pi_2\pi_1(k)$, ..., $\pi_N\pi_{N-1} \cdots \pi_2\pi_1(k)$ en utilisant Lecture d'un secret (DISTOR) avec J_{i+1}, J_{i+2}, ...,J_N.

PROTOCOLE 12 Lecture d'une carte(i)

(1) J_i choisit $k \in \{1,2,...,52\}$ libre et la marque "utilisée"
(2) $c \leftarrow k$
(3) POUR $j = 1$ A $i-1$
(3.1) J_i obtient publiquement $\pi_j(c)$ de J_j
(3.2) $c \leftarrow \pi_j(c)$
(4) FIN
(5) J_i calcule secrètement $c \leftarrow \pi_i(c)$
(6) POUR j = i+1 A N
(6.1) J_i obtient $<\pi_j(c),\tau_{j,i}(c)>$ en utilisant le PROTOCOLE 10 avec Alice=J_j et Bob=J_i
(6.2) J_i pose secrètement $c \leftarrow \pi_j(c)$
(7) FIN
(8) CARTE$\leftarrow c$

5.3. Détection des tricheurs grâce aux traces.

Soit X un ensemble et f une fonction dont le domaine contient X, on dénote $f(X) = \{y \mid \exists x \in X \ [y=f(x)]\}$. Dénotons également par $\psi_{i,j}(k)$ le couple $<k,\tau_{i,j}(k)>$.

Soit $H_{i,j} \subseteq \{1,2,...,52\}$, l'ensemble des entrées que possède J_j dans la permutation π_i (les entrées qu'il a lues avec DISTOR). Par extension de cette définition, $H_{0,j}$ est l'ensemble des identificateurs que J_j a choisis afin d'obtenir des cartes. Similairement, soit $T_{i,j} \subseteq \psi_{i,j}(\{1,2,...,52\})$ l'ensemble des indices et des chaînes que possède J_j dans $\tau_{i,j}(\{1,2,...,52\})$, pour $1 \le j \le N$ et $j < i \le N$.

Quand un joueur J_j est honnête alors on a que

$$H_{i,j} = \pi_i(H_{i-1,j}), \quad 1 \le i \le N.$$

Etant donné que dans le protocole de lecture des cartes les valeurs lues dans les permutations des joueurs précédant J_j sont connues de tous, alors les valeurs de $H_{0,j}, H_{1,j}, ..., H_{j-1,j}$ sont aussi connues publiquement. Donc, comme on l'a souligné précédemment, si J_j triche alors

$$\exists i_0 > j,\ H_{i_0,j} \neq \pi_{i_0}(H_{i_0-1,j}) \text{ et } \forall i < i_0,\ H_{i,j} = \pi_i(H_{i-1,j}).$$

Soit $i \geq i_0$. Il faut maintenant considérer deux possibilités concernant la triche de J_j aux dépends de J_i, que l'on suppose honnête:

1) $H_{i,j} \cap H_{i,i} = \emptyset$.

Dans ce cas J_j ne pourra jamais savoir qu'une carte qu'il a lue illégalement appartenait à J_i. Tout au plus, J_j peut espérer dans ce cas, obtenir des cartes n'appartenant à personne, des cartes dont il ne connaît pas le propriétaire ou des cartes appartenant à quelqu'un d'autre que J_i. Ceci est sans danger pour J_i dans tous les cas. Le jeu et la stratégie de J_i ne sont pas mises en danger.

2) $H_{i,j} \cap H_{i,i} \neq \emptyset$.

Dans ce second cas J_j peut apprendre des cartes appartenant à J_i. C'est donc dans cette situation que les traces seront utilisées afin que J_i puisse le détecter. A n'importe quel moment de la partie, particulièrement avant de poser un geste stratégique, J_i peut décider de tester l'honnêteté de ses adversaires. Etudions le cas de J_j.

Puisque $H_{i,j}$ a été lu avec DISTOR, J_j connaît nécessairement aussi $T_{i,j}$. Afin de compromettre J_j, J_i va d'abord lui demander de rendre publique une version attestée des indices et des traces $T_{i,j}$ qu'il connaît. A ce point J_j a deux choix, (*) il peut attester les valeurs qu'il connaît comme lui demande J_i ou (**) attester à la place des indices et des chaînes quelconques de son choix.

Par ailleurs, tout le monde connaît la position des entrées que J_i possède dans sa propre permutation, c'est-à-dire que $H_{i-1,i}$ est connu publiquement. J_i annonce publiquement les valeurs $\tau_{i,j}$ qu'il avait attestées plus tôt dans la partie et indique lesquelles lui appartiennent. Il peut prouver que ces valeurs sont bien celles qu'il avait attestées précédemment en ouvrant $\hat{\tau}_{i,j}$.

Maintenant, J_j doit prouver que les chaînes qu'il vient d'attester font partie de celles n'appartenant pas à J_i. Dénotons $\#_k$ la cardinalité de $H_{i,k}$. Pour arriver à cette fin, J_j doit d'abord attester les $52-\#_i-\#_j$ indices et chaînes qui théoriquement n'appartiennent ni à J_i, ni à lui. Ensuite il utilise le protocole de la section 5.1 afin de prouver que les $52-\#_i$ indices et chaînes qu'il a attestées forment une permutation des $52-\#_i$ n'appartenant pas à J_i. Il prouve du même coup que ses $\#_j$ chaînes étaient parmi les 52 offertes par J_i et qu'elles n'étaient pas parmi celles lui appartenant. Si J_j

a utilisé l'alternative (**) tel que décrit plus haut, il n'a qu'une probabilité de 2^{-s} de produire des chaînes provenant vraiment des 52 qu'offrait J_i. Aussi, s'il a utilisé l'alternative (*), alors il aura attesté une chaîne appartenant à J_i parmi ses $\#_j$. Ceci rendra improbable le fait de prouver que ses $52-\#_i$ chaînes forment une permutation de celles n'appartenant pas à J_i. Donc quoique fasse J_j, il a une chance exponentiellement faible de ne pas être détecté par J_i.

Normalement, J_i engagera ce protocole de détection de tricheurs avec chacun des adversaires le précédant à chaque fois qu'il voudra faire une opération pouvant révéler une partie de sa stratégie. Par exemple, avant d'annoncer le nombre de cartes qu'il veut changer ou avant de miser sur son jeu. Néanmoins, il est essentiel de pouvoir continuer la partie en toute sécurité donc de rétablir une nouvelle fonction $\tau_{i,j}$ afin de pouvoir refaire de nouveau le protocole de détection plus tard si nécessaire.

PROTOCOLE 13 Détection des Tricheurs

(1) POUR $1 \le j \le N-1$
(1.1) POUR $j+1 \le i \le N$
(1.1.1) J_j atteste $T_{i,j}$ par $\hat{T}_{i,j}$
(1.1.2) J_i annonce $\tau_{i,j}(k)$, $1 \le k \le 52$
(1.1.3) J_j calcule secrètement $T'_{i,j} \leftarrow \psi_{i,j}(\{1,2,...,52\} - H_{i-1,i}) - T_{i,j}$
(1.1.4) J_j atteste $T'_{i,j}$ par $\hat{T}'_{i,j}$
(1.1.5) J_j prouve que $\hat{T}_{i,j} \cup \hat{T}'_{i,j}$ atteste une permutation
de $\psi_{i,j}(\{1,2,...,52\} - H_{i-1,i})$ en utiliant le PROTOCOLE 5.
(1.2) FIN
(2) FIN

5.4. Reconstruction des traces

Après avoir vérifié que ses adversaires ont été honnêtes avec lui, J_i va rétablir avec chacun d'eux, une nouvelle fonction lié à π_i. Pour ceci, J_i génère 52 chaînes (pas necéssairement distinctes des précédentes) et les atteste comme il l'avait fait avec les valeurs précédentes de la fonction. Puis il recommence la préparation de DISTOR afin de permettre à J_j de produire 52 séries de questions sur les $<\pi_i(x),\tau_{i,j}(x)>$ pour $1\leq x\leq 52$. Afin de permettre à J_i de savoir lesquelles des séries de questions ont été utilisées jusqu'à ce point dans la partie, J_j devra modifier uniquement la partie de ses 52 séries qui s'applique aux chaînes $\tau_{i,j}(x)$ et conserver intacte la partie destinée aux $\pi_i(x)$. J_i et J_j doivent ensuite répéter la partie du protocole DISTOR qui convainc J_i que les séries de questions de J_j sont correctes. Et finalement, J_i répond de nouveau aux questions auxquelles il avait répondu précédemment dans la partie afin de donner à J_j les nouvelles valeurs de $\tau_{i,j}(x)$ lui appartenant.

PROTOCOLE 14 Reconstruction des traces

(1) POUR $i \in \{1,2,...,N\}$
(1.1) POUR $1\leq j<i$
(1.1.1) J_i choisit $\tau_{i,j}:\{1,2,...,52\}\rightarrow\{0,1\}^s$
(1.1.2) J_i atteste $\tau_{i,j}(k)$ par $\hat{\tau}_{i,j,k}$ pour $1\leq k\leq 52$
(1.1.3) utiliser Preparation de DISTOR† avec Alice=J_i, Bob=J_j,
$\hat{S}_{u,v}=\hat{\pi}_{i,u,v}$ et $\hat{S}_{u,w+6}=\hat{\tau}_{i,j,u,w}$, pour $1\leq u\leq 52$, $1\leq v\leq 6$ et $1\leq w\leq s$
(1.2) FIN
(2) FIN

5.5. Ouverture et écart d'une carte

Il nous faut maintenant discuter de la façon utilisée pour ouvrir ou écarter une carte. A première vue, on pourrait croire qu'il suffit de nommer la carte par son identificateur k et de dire que l'on veux l'écarter ou de révéler $\pi_1(k)$, $\pi_2\pi_1(k)$, ..., $\pi_N\pi_{N-1}\cdots\pi_1(k)$ et aussi $\tau_{i+1,i}\pi_i\pi_{i-1}\cdots\pi_1(k)$ $\tau_{i+2,i}\pi_{i+1}\pi_i\cdots\pi_1(k)$, ..., $\tau_{N,i}\pi_{N-1}\pi_{N-2}\cdots\pi_1(k)$ si l'on veux l'ouvrir. Mais il faut se rendre compte qu'une telle méthode révèle beaucoup d'information sur la stratégie des joueurs.

Comme il est expliqué au chapitre 2, le fait d'apprendre l'origine des cartes, comme c'est le cas ici, peut être une information pertinente pour les adversaires. Ceci permet, entre autres, aux joueurs de savoir qu'un joueur a commis une erreur de stratégie, d'apprendre les préférences d'un joueur ou de savoir à quel point il est

† ρ et $a_{u,v}$ pour $1\leq v\leq 6$, doivent être identiques à ceux utilisés précédemment.

entreprenant. Pour régler ce problème, chaque joueur J_i va gérer publiquement un ensemble M_i déterminant les cartes dans son jeu et E_i des cartes qu'il a écartées. Ces M_i et E_i sont des sous-ensembles de $H_{i,i}$.

Initialement $M_i = E_i = \emptyset$. A chaque fois que J_i prend une nouvelle carte identifiée par k, il utilise l'attestation qu'il possède pour la valeur $\pi_i \pi_{i-1} \cdots \pi_1(k)$ et la déclare membre de M_i. Au moment d'ouvrir ou d'écarter une carte il va d'abord embrouiller son ensemble M_i afin de faire perdre l'origine de ses cartes à ses adversaires. Pour ceci J_i génère de nouvelles attestations pour les éléments de M_i et les présente dans un ordre permuté à ses adversaires. Ensuite il prouve que les nouvelles attestations forment une permutation des anciennes, à l'aide du protocole de la section 5.1. Ceci convainc ses adversaires que les valeurs attestées sont les mêmes mais ils perdent toute information sur leur origine!

S'il veut écarter une carte, il prend maintenant son attestation qui est membre de M_i et la déclare membre de E_i. Il retire également cette attestation de M_i. Si par ailleurs, il veut ouvrir une carte, alors il prend la valeur de M_i qui correspond à la carte qu'il veut ouvrir et révèle la valeur qu'elle atteste, disons c. Il exhibe finalement à ses adversaires les valeurs $\pi_{i+1}(c)$, $\pi_{i+2}\pi_{i+1}(c)$, ..., $\pi_N \pi_{N-1} \cdots \pi_{i+1}(c)$ et aussi $\tau_{i+1,i}(c)$, $\tau_{i+2,i}\pi_{i+1}(c)$, ..., $\tau_{N,i}\pi_{N-1}\pi_{N-2} \cdots \pi_{i+1}(c)$ afin de leur prouver qu'il a été honnête de c à la fin.

PROTOCOLE 15 Ecart(i,k)

(1) J_i place l'entrée de $\hat{M}_i$ correspondant à $\pi_i \pi_{i-1} \cdots \pi_1(k)$
dans $\hat{E}_i$ et la retire de $\hat{M}_i$.

PROTOCOLE 16 Ouverture(i,k)

(1) $c \leftarrow \pi_i \pi_{i-1} \cdots \pi_1(k)$
(2) J_i dévoile c
(3) J_i ouvre l'entrée de $\hat{M}_i$ correspondant à c.
(4) POUR $j = i+1$ A N
(4.1) J_i dévoile $\pi_j(c)$ et $\tau_{j,i}(c)$
(4.2) J_j ouvre $\pi_j(c)$ et $\tau_{j,i}(c)$.
(4.3) $c \leftarrow \pi_j(c)$
(5) FIN

5.6. Structure globale

Voici enfin le protocole global permettant de combiner tous ceux déjà expliqués. Malgré qu'il soit moins formel que les autres, il contient néanmoins l'essentiel permettant de jouer la partie en toute sécurité. Bien que le fait ne soit pas mentionné explicitement, un joueur qui serait pris en défaut ou qui se verrait dans l'impossibilité d'exécuter son protocole sera expulsé et les joueurs restants devront recommencer la partie à cause de la présence du tricheur.

PROTOCOLE 17 Poker (version 2)

(1) chaque joueur J_i utilise Preparation(i) (partie A & B)

(2) REPETER

(2.1) chaque joueur J_i obtient ses cartes grâce
au PROTOCOLE 12 Lecture d'une carte(i)

(2.2) chaque joueur J_i ajoute $\hat{\pi}_i(\pi_{i-1} \cdots \pi_1(k))$ à $\hat{M}_i$
pour chaque carte qu'il obtient à partir d'un identificateur k.

Tous les joueurs:

(2.3)† utilisent le PROTOCOLE 13 Détection des tricheurs

(2.4) chaque J_i génère $\hat{M}_i{}'$
une permutation de $\hat{M}_i$.
Il le prouve en utilisant le PROTOCOLE 5.

selon le déroulement de la partie, tous les joueurs:

(2.5) misent, écartent des cartes ou en ouvrent
ils utilisent pour ceci les PROTOCOLEs 15 et 16.

(2.6) SI ce n'est pas la fin de la partie

(2.6.1) ALORS chaque J_i utilise Reconstruction des traces(i)

(3) JUSQU'A la fin de la partie

† (2.3) et (2.4) ne sont pas nécessaires au premier tour de boucle du REPETER.

Conclusion

Les techniques de la cryptographie moderne nous permettent de simuler tous les aspects d'une véritable partie de poker dans le contexte où les joueurs ne peuvent pas physiquement se rencontrer. La solution proposée ici est la première qui garantisse la sécurité de la stratégie des joueurs. Ce nouveau développement rend beaucoup plus crédible l'utilité de la chose car précédemment, personne intéressé au jeu n'aurait accepté de jouer dans les conditions proposées. En plus de garantir une sécurité exceptionnelle, ce protocole permet à quiconque de participer au jeu sans risquer de compromettre sa stratégie par ses attitudes, sa nervosité, etc.

A l'aide des algorithmes connus actuellement dans le domaine de la théorie des nombres, il est clairement possible de construire des ordinateurs dédiés possédant un nombre de processeurs arithmétiques raisonnables (52×6) qui permettraient une implantation en temps réel du protocole. Sur une telle machine, le temps de calcul pour chaque étape du protocole est proportionnelle à s, le paramètre de sécurité. Par contre la probabilité de tricher sans être détecté est proportionnelle à 2^{-s}, si HRQ est vraie. Il est clair qu'aucune machine ne peut suffire actuellement pour tricher le protocole, à moins d'un progrès important dans les algorithmes pouvant décider la résiduosité quadratique. Et tant qu'aucun algorithme polynomial ne sera trouvé pour résoudre ce problème, il suffira de choisir un s suffisamment grand pour que la triche reste matériellement impossible.

En plus d'avoir réussi à résoudre le problème posé, ce mémoire ouvre la porte sur de nouvelles directions de recherches intéressantes. Premièrement DISTOR est un protocole d'utilité générale que ce mémoire introduit comme nouvel outil cryptographique. La question de savoir si DISTOR peut être réalisé dans d'autres contextes que sous HRQ est étudiée dans [BCR1] et [Ch]. On étudie également dans [BCR2] la possibilité de construire un tel protocole inattaquable au sens de Shannon. Egalement, le protocole 5 permettant de prouver la validité d'une permutation peut être utilisé de façon beaucoup plus générale. Il permet de prouver des énoncés beaucoup plus complexes sur des ensembles de chaînes attestées. On montre dans [BC] qu'un tel protocole permet de prouver l'appartenance à un quelconque langage L d'une chaîne attestée, à la seule condition que L soit dans **NP**.

Enfin, si on s'intéresse aux jeux de hasard en général, ce mémoire ouvre la voie sur une technique pouvant simuler pratiquement n'importe quel jeu de hasard existant (voir [Cr1]). Le résultat de [BC] cité plus haut permet entre autres d'étendre le protocole pour échanger des cartes, retourner des cartes dans le paquet, rebrasser le paquet de cartes et autres, ceci rendant possible la simulation de jeux tel le blackjack, le poker indien ou le scrabble.

Il serait souhaitable qu'un protocole similaire à celui présenté dans ce mémoire puisse être établi sous la seule hypothèse de l'existence d'une méthode d'attestation sécure et non pas sur un cas particulier comme HRQ. Un tel protocole est une voie de recherche intéressante. L'intérêt pour un tel protocole serait qu'il ne dépend pas d'une hypothèse spécifique; rappelons qu'une certaine hypothèse est nécessaire car [SRA] ont démontré l'impossibilité d'obtenir un protocole parfaitement inattaquable au sens de Shannon. Egalement, la simplification des protocoles rendrait encore plus réaliste leur implantation; pour une version améliorée des protocoles consulter [Cr2].

Remerciements

Je tiens à remercier premièrement Jean-Marc Robert pour son support et ses encouragements. Je remercie également Pierre McKenzie pour son travail, ses encouragements et son aide précieuse. Je remercie Roméo Guasteferi pour avoir écouté, sans rien comprendre, la première version de ce protocole. Je tiens à remercier spécialement David Chaum pour ma visite à Amsterdam qui a élargie ma vision de la cryptographie. Mes plus sincères remerciements vont à Gilles Brassard, mon directeur et mon ami, pour son support, ses encouragements, ses suggestions et pour tout le travail que nous avons accompli ensemble subséquemment à la rédaction de ce mémoire. Je remercie Michel Boyer, pour sa participation au comité d'évaluation de ce mémoire. J'accorde un remerciement spécial à Bernard Cantin et Michel Content pour m'avoir initié au poker. Finalement, je remercie de tout mon coeur Roxane Lacouture, pour son support, son dévouement et son sens pratique qui m'est fort précieux.

Bibliographie

[Ad] Adleman, L., ‘‘On Distinguishing Prime Numbers from Composite Numbers’’, dans *Proceedings of the 21st IEEE Symposium on the foundations of Computer Science (FOCS)*, Syracuse, N.Y., 1980, pp. 387-408.

[B] Brassard, G., ‘‘On Computationally Secure Authentication Tags Requiring Short Secret Shared Keys’’, dans *Advances in Cryptology:Proc. of CRYPTO 82*, D. Chaum, R. L. Rivest et A. T. Sherman, eds., Plenum, New York, 1983, pp. 79-86.

[BC] Brassard, G. et C. Crépeau, ‘‘Zero-Knowledge Simulation of Boolean Circuits’’, à paraître dans *Advances in Cryptology:Proc. of CRYPTO 86*.

[BCR1] Brassard G., C. Crépeau et J.-M. Robert, ‘‘All-or-Nothing Disclosure of Secrets’’, à paraître dans *Advances in Cryptology:Proc. of CRYPTO 86*.

[BCR2] Brassard G., C. Crépeau et J.-M. Robert, ‘‘Information Theoretic Reduction among Disclosure Problems’’, dans *Proceedings of the 27th Annual IEEE Symposium on the Foundations of Computer Science*, 1986.

[BF] Banary, I. et Z. Füredi, ‘‘Mental Poker with Three or More Players’’ dans *Information and Control*, 59 (1983), pp. 84-93.

[Ch] Chaum, D., *communiqué personel*, mai 1986.

[Co] Coppersmith, D., ‘‘Cheating at Mental Poker’’, dans *Advances in Cryptology: Proceedings of CRYPTO 85*, H. C. Williams ed., Lecture Notes in Computer Science 218, Springer-Verlag, Berlin, 1986, pp. 104-107.

[Cr1] Crépeau C., ‘‘A Secure Poker Protocol that Minimizes the Effect of the Player Coalitions’’, dans *Advances in Cryptology: Proceedings of CRYPTO 85*, H. C. Williams ed., Lecture Notes in Computer Science 218, Springer-Verlag, Berlin, 1986, pp73-86.

[Cr2] Crépeau C., ‘‘A Zero-Knowledge Poker Protocol that Achieves Confidentiality of the Players’ Strategy, *or* How to Achieve an Electronic Poker Face’’, à paraître dans *Advances in Cryptology:Proc. of CRYPTO 86.*

[FM] Fortune, S. et M. Merrit, ‘‘Poker Protocols’’ dans *Advances in Cryptology: Proc. of CRYPTO 84*, G. R. Blakley et D. Chaum, eds., Lecture Notes in Computer Science 196, Springer-Verlag, Berlin, 1985, pp. 454-464.

[Ga] Gauss, C.F., ‘‘Disquisitiones Arithmeticae’’, 1801, traduit par A. Arthur et S. J. Clark, Yale univ. Press, New Haven, 1966.

[GHY] Galil, Z., S. Haber et M. Yung, ‘‘A Private Interactive Test of a Boolean Predicate and Minimum-Knowledge Public-Key Cryptosystems’’ dans *Proceedings of the 26th Annual IEEE Symposium on the Foundations of Computer Science,* 1985, pp. 360-371.

[GM1] Goldwasser, S. et S. Micali, ‘‘Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information’’ dans *Proceedings of the 14th Annual ACM symp. on Theory of computing (STOC)*, 1982, pp. 365-377.

[GM2] Goldwasser, S. et S. Micali, ‘‘Probabilistic Encryption’’ dans *J. Comput. System Sci.*, 28 (1984), pp. 270-299.

[GMRa] Goldwasser, S., S. Micali et C. Rackoff, ‘‘The Knowledge Complexity of Interactive Proof-Systems’’, dans *Proceedings of the 17th Annual ACM Symposium on the Theory of Computing*, 1985, pp. 291-304.

[GMRi] Goldwasser, S., S. Micali et R. Rivest, ‘‘A "Paradoxical" Solution to the Signature Problem’’, dans *Proceedings of the 25th Annual IEEE Symposium on the Foundations of Computer Science*, 1984, pp. 441-448.

[GMY] Goldwasser, S., S. Micali et A. Yao, ‘‘Strong signature Schemes’’, dans *Proceedings of the 15th Annual ACM symp. on Theory of computing (STOC)*, 1983, pp. 431-439.

[Li] Lipton, R., ‘‘How to Cheat at Mental Poker’’, dans *Prooceedings of the AMS short course on Cryptology*, Janvier, 1981.

[NZ] Niven, I. et H.S. Zuckerman, ‘‘The Theory of Numbers’’, quatrième édition, Editions John Wiley & Sons, Toronto, 1980.

[Po] Poli, F., ‘‘le poker’’, éditions SOLAR, 1977.

[Ra1] Rabin, M. O., ‘‘Probabilistic algorithms’’, dans *Algorithms and Their Complexity: Recent Results and New Directions*, J. F. Traub (editor), Academic

Press, New York, NY, 1976, pp. 21-39.

[Ra2] Rabin, M. O., ‘‘Digitalized signatures and public-key functions as intractable as factorization’’, *MIT/LCS/TR-22*, 1979.

[RSA] Rivest, R., A. Shamir et L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", dans *Communications of the ACM* 21,2 (Février 1978), pp. 120-126.

[SRA] Shamir, A., R. Rivest et L. Adleman, ‘‘Mental Poker’’, *MIT/LCS/TM-125*, 1978.

[SS] Solovay, R. et V. Strassen, ‘‘A fast Monte Carlo test for primality’’, dans *SIAM Journal on Computing*, Vol. 6, 1977, pp. 84-85.

[WC] Wegman, M. N., et J. L. Carter, ‘‘New Hashing Functions and Their Use in Authentication and set Equality’’, dans *Journal of Computing and System Sciences*, pp. 265-279.

[Yu] Yung, M., ‘‘Cryptoprotocols: Subscription to a Public Key, The Secret Blocking and the Multi-Player Mental Poker Game’’ dans *Advances in Cryptology: Proc. of CRYPTO 84*, G. R. Blakley et D. Chaum, eds., Lecture Notes in Computer Science 196, Springer-Verlag, Berlin, 1985, pp. 439-453.

Table des matières

Sommaire

N personnes veulent jouer au poker. Malheureusement il leur est impossible de se rencontrer physiquement. Comment pourront-elles, grâce aux ordinateurs, arriver à leur fin? Evidemment le problème est très simple à résoudre dans le cas où les joueurs sont honnêtes. Mais bien sûr, chacun va tricher aussitôt que l'occasion va se présenter. Il faut donc que le protocole permette la détection des tricheurs. Idéalement nous voudrions obtenir un protocole respectant toutes les contraintes d'une partie réelle:

- Absence d'un parti honnête
- Unicité des cartes
- Distribution uniforme des jeux possibles
- Impossibilité d'influencer la distribution des cartes
- Détection des tricheurs avec haute probabilité
- Confidentialité des cartes
- Effet minimal des coalitions
- Confidentialité complète de la stratégie des joueurs.

Les techniques de la cryptographie moderne seront l'outil principal du protocole présenté dans ce mémoire. Néanmoins, de nouvelles techniques ont dû être inventées afin d'arriver au but fixé. Parmi celles-ci on trouve DISTOR, un protocole d'usage général qui permet de divulguer un secret parmi plusieurs en tout ou rien.

On trouve dans ce mémoire le premier protocole satisfaisant toutes les contraintes permettant de simuler une partie de poker tel qu'elle est jouée dans la réalité. Le protocole proposé est efficace, dans le sens qu'il peut être implanté en temps réel en utilisant la technologie actuelle. Aussi, ce protocole est sûr sous l'hypothèse qu'il n'existe aucun algorithme efficace pour décider de la résiduosité quadratique modulo un nombre composé (HRQ), c'est-à-dire qu'il est impossible de tricher le protocole, quel que soit l'ordinateur que l'on utilise à cette fin.

UNIVERSITE DE MONTREAL

PROTOCOLE CRYPTOGRAPHIQUE DE POKER ELECTRONIQUE PERMETTANT LA CONFIDENTIALITE DE LA STRATEGIE

PAR
CLAUDE CREPEAU
DEPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPERATIONNELLE
FACULTE DES ARTS ET DES SCIENCES

MEMOIRE PRESENTE A LA FACULTE DES ETUDES SUPERIEURES
EN VUE DE L'OBTENTION DU GRADE DE
MAITRE ES SCIENCES (M. Sc.)

(JUILLET, 1986)