Preuves sans divulgation de connaissance

Julien Bureaux

20 décembre 2008

L'objet de ce travail de rédaction est de présenter la notion de preuve sans divulgation de connaissance telle qu'elle fut introduite par Goldwasser, Micali, et Rackoff en 1985 [GMR]. Il s'agit d'un échange entre deux agents P et V sur le principe suivant : V veut *vérifier* que P est capable de *prouver* l'appartenance d'un mot à un langage donné, et ce sans que celui-ci ne révèle la moindre information sur la preuve elle-même. De tels protocoles sont particulièrement intéressants en cryptographie [POUP]. Goldreich, Micali,et Wigderson ont prouvé que tous les langages **NP** possèdent un tel système de preuve interactive [GMW].

Table des matières

1 Introduction **1**
1.1 Un premier exemple 1
1.2 Trois conditions nécessaires 2

2 Preuves interactives **3**
2.1 Protocoles interactifs 3
2.2 Systèmes de preuve interactifs 4

3 Preuves sans divulgation de connaissance **5**
3.1 Indistinguabilité de variables aléatoires 5
3.2 Approximabilité et non divulgation de connaissance 6

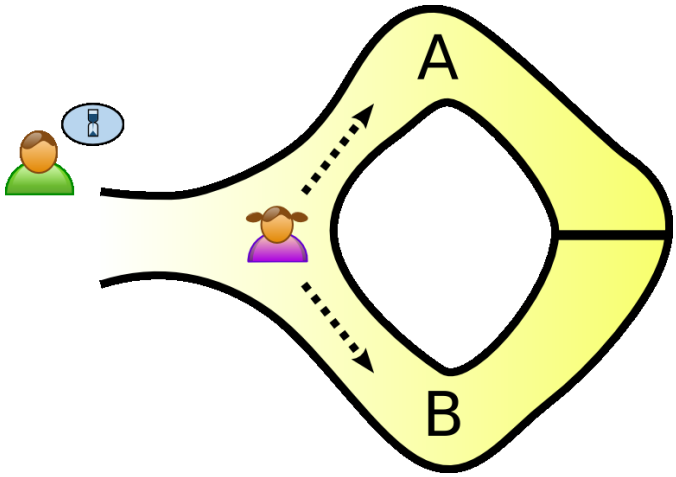
4 Preuve de la résiduosité quadratique **7**

1 Introduction

1.1 Un premier exemple

Dans un article intitulé *How to explain Zero-Knowledge protocols to your children* Quisquater et Guillou proposent l'exemple suivante : Alice veut prouver à Bob qu'elle

connait le mot magique pour ouvrir la porte au fond de la grotte représentée ci-dessous, mais sans lui dire le mot magique.



(réalisé par Dake sous license Creative Commons)

Ils se mettent d'accord sur l'expérience suivante : Tandis que Bob reste à l'extérieur, Alice rentre dans la grotte et lance une pièce pour savoir dans quel couloir aller. Une fois qu'elle s'est arrêtée, Bob se place à la disjonction des couloirs, lance une seconde pièce pour en choisir un et crie à Alice le résultat, lui laisse une minute, puis rentre dans le couloir choisi et vérifie que Alice l'y attend.

Il est clair que si Alice sait comment ouvrir la porte elle pourra toujours aller dans le couloir choisi par Bob. Réciproquement si elle a menti et ne connait pas le mot magique, alors elle a une chance sur deux d'être dans le bon couloir. Si l'expérience est répétée N fois elle n'aura qu'une probabilité 2^{-N} d'avoir été à chaque fois dans le bon couloir. On peut alors supposer qu'à partir de $N = 10$ Bob est convaincu qu'elle a dit la vérité. Notons cependant qu'il est impossible pour Bob de convaincre qui que ce soit d'autre qu'Alice connait le mot magique puisqu'il n'a eu accès à aucune information autre que la quasi-certitude qu'elle le connait effectivement.

1.2 Trois conditions nécessaires

L'exemple précédent permet de mettre en avant trois propriétés que toute preuve sans divulgation de connaissance doit nécessairement posséder.

Les deux premières correspondent à ce qu'on attend intuitivement de toute *preuve interactive* (une définition formelle sera donnée dans le 2), à savoir :

Consistance : Le vérificateur V doit être convaincu par la preuve de P lorsque le protocole est correctement suivi par les deux interlocuteurs.

Sûreté : Si la preuve est erronnée, le fournisseur de preuve P malhonnête ne doit pas être en mesure de convaincre V de sa véracité.

La troisième condition est quant à elle inhérente à la notion de *non divulgation de connaissance* (qui développée dans la partie 3).

Secret : V ne doit obtenir de P aucune information sur la preuve autre que sa véracité, même si V est malhonnête et ne respecte pas le protocole.

Notons de plus, que l'introduction d'expériences aléatoires dans le protocole semble clairement nécessaire dans le cadre d'une preuve sans divulgation de connaissance. On peut faire un parallèle avec la méthode suivante utilisée par les sondeurs pour obtenir des statistiques sur des questions qui peuvent être gênantes : on pose au sondé une question commune et une question gênante et on lui demande de tirer à pile ou face la question à laquelle répondre et de n'indiquer qu'un oui ou un non au sondeur ; un second sondage portant sur la question commune permet d'obtenir le résultat souhaité.

2 Preuves interactives

Il s'agit ici de donner une définition formelle à la notion de *preuve interactive* en respectant les conditions de *consistance* et de *sûreté* précédemment évoquée. On commence par introduire le modèle de calcul utilisé, à savoir modéliser les des deux interlocuteurs par deux machines de Turing mises en communication.

2.1 Protocoles interactifs

Afin de pouvoir faire intervenir l'aléa, on considère ici une version randomisée des machines de Turing en leur ajoutant une bande aléatoire de bits (ce qui correspond à leur donner la possibilité de faire des lancés de pièce). On note une fois pour toute $(\Omega, \mathcal{A}, \mathbb{P})$ l'espace probabiliste.

Définition Si $(X_i)_{i\in\mathbf{N}}$ est une suite de variables aléatoires à valeurs dans $\{0,1\}$, indépendantes et de loi uniforme, on appelle *machine de Turing probabiliste de bande aléatoire* $(X_i)_{i\in\mathbf{N}}$ une machine de Turing munie d'une bande supplémentaire dont la case de numéro i contient la variable aléatoire X_i. De plus cette bande est en lecture seule, et ne peut être lue que de la gauche vers la droite.

Par la suite on parlera plus simplement de *machine de Turing probabiliste*, sans faire référence à la suite de variables aléatoires sous-jacente. Il existe toute une théorie de la complexité adaptée à cette définition des machine sur laquelle on ne s'étendra pas ici.

On introduit des machines munie de bande de communications (on peut se le permettre car l'ajout d'un nombre fini de bandes ne change pas la classe de complexité) pour simplifier la description des protocoles.

Définition On appelle *machine de Turing interactive* (MTI) une machine de Turing déterministe et probabiliste munie
- d'une bande d'entrée en lecture seule
- d'une bande de travail
- d'une bande aléatoire
- d'une bande de communication entrante, en lecture seule
- d'une bande de communication sortante, en écriture seule

On va maintenant coupler ces machines.

Définition On appelle *protocole interactif* un couple noté (A, B) de machines de Turing interactives A et B de bandes aléatoires indépendantes et telles que

- A et B partagent la même bande d'entrée et la bande de communication sortante de l'une est la bande de communication entrante de l'autre.
- Les deux machines sont actives tour à tour, en commençant par B.
- Lorsqu'elle s'active, la machine $A(B)$ s'exécute en utilisant la bande d'entrée, sa bande de travail, sa bande aléatoire et sa bande de communication entrante, puis elle écrit un mot sur la bande de communication sortante et se désactive. $B(A)$ elle alors activée à moins que le protocole ait été interrompu. $B(A)$ procède de même.
- Le i-ème message de $A(B)$, noté m_i, est le mot contenu (c'est à dire sans les blancs terminaux) sur la bande de communication sortante de $A(B)$ après sa i-ème activation. On pose $m_0 = \epsilon$.
- Chacune des machines peut interrompre le protocole en n'écrivant pas de message sur sa bande de communication sortante. La machine B décide alors d'*accepter* ou de *rejeter* le mot m d'entrée.
- La complexité en temps de la machine B pour une entrée initiale m est définie comme la somme des complexités en temps de B lors des activations de $B(A)$. Si le protocole s'arrête après $n+1$ activations de $B(A)$, elle vaut

$$t_B(m) = \sum_{i=0}^{n} t_B(m, m_i)$$

- Cette complexité est supposée polynomialement bornée : il existe un entier $k > 0$ tel que pour tout $m \in \{0,1\}^*$ on ait $t_B(m) = O(|m|^k)$ presque sûrement.

Notons que dans ce modèle la complexité en temps de la machine de Turing A n'est par contre pas limitée a priori.

On définit finalement une variable aléatoire qui va rendre compte du comportement du protocole sur une entrée donnée.

Définition Soit (A, B) un protocole interactif. À tout mot $m \in \{0,1\}^*$ en entrée de (A, B) on associe la variable $B_A(m)$ définie par

$$B_A(m) = \begin{cases} 1 & \text{si } B \text{ accepte } m \\ 0 & \text{sinon} \end{cases}$$

On peut remarquer une certaine dissymétrie dans cette définition. En effet $B_A(m) = 0$ correspond à la fois aux cas où la machine rejette l'entrée et aux cas où le protocole ne s'arrête pas.

2.2 Systèmes de preuve interactifs

On se contente de définir un système de preuve pour un langage donné comme un protocole interactif vérifiant les conditions de consistante et de sûreté dont on a déjà

parlé.

Définition Soit $\mathcal{L}$ un langage sur $\{0,1\}$. On dit qu'un protocole interactif (P,V) est un *système de preuve interactif pour* $\mathcal{L}$ s'il vérifie les deux conditions suivantes :

Consistance : Pour tout entier $k > 0$, tout mot $m \in \mathcal{L}$ suffisamment grand donné en entrée de (P,V) est accepté par V avec une probabilité au moins égale à $1 - |m|^{-k}$.

$$\forall k \in \mathbf{N}^*, \exists n_0 \in \mathbf{N}, \forall m \in \mathcal{L}, \quad |m| > n_0 \Longrightarrow \mathbb{P}(V_P(m) = 1) \geq 1 - |m|^{-k}$$

Sûreté : Pour toute machine de Turing interactive $\tilde{P}$: pour tout entier $k > 0$ et pour tout mot $m \notin \mathcal{L}$ suffisamment grand en entrée de $(\tilde{P},V)$ est accepté par V avec une probabilité au plus égale à $|m|^{-k}$.

$$\forall k \in \mathbf{N}^*, \exists n_0 \in \mathbf{N}, \forall m \notin \mathcal{L}, \quad |m| > n_0 \Longrightarrow \mathbb{P}(V_{\tilde{P}}(m) = 1) \leq |m|^{-k}$$

Cette définition a le mérité de coller d'assez près à notre intuition et en particulier s'adapte parfaitement à l'exemle de la caverne donné en introduction. Soulignons de nouveau le fait que si la puissance de calcul de V est polynomialement limitée en temps, ce n'est pas le cas de celle de P.

Définition On désigne par **IP** (*Interactive Polynomial time*) la classe des langages pour lesquels il existe un système de preuve interactive.

Étant donnée l'équivalence entre le caractère **NP** et l'existence de vérificateur en temps polynomial on a immédiatement l'inclusion $\mathbf{NP} \subset \mathbf{IP}$. L'inclusion réciproque est plus problèmatique : le problème *d'isomorphisme de graphes* abordé dans [GMW] par exemple est dans **IP** mais n'est pas connu comme étant dans **NP**.

3 Preuves sans divulgation de connaissance

Dans cette section, on s'intéresse à la condition de secret. On commence par définir l'indistinguabilité de deux familles de variables aléatoires.

3.1 Indistinguabilité de variables aléatoires

Soit $\mathcal{L}$ un langage donné sur $\{0,1\}$. Il s'agit ici de formaliser l'idée que deux familles de variables aléatoires indexées sur $\mathcal{L}$ ne puissent pas être différentiées par un juge. Dans le cas de l'indistinguabilité parfaite, le juge est supposé tout puissant.

Définition Deux familles (X_m) et (Y_m) de variables aléatoires à valeurs dans $\{0,1\}^*$ indexées par $m \in \mathcal{L}$ sont dites *parfaitement indistinguables* sur $\mathcal{L}$ si elles sont égales.

$$\forall m \in \mathcal{L}, \quad X_m = Y_m$$

Pour l'indistinguabilité polynomiale on suppose que le juge ne peut traiter que des exemples polynomialement limités par rapport à taille de l'entrée. C'est cette définition qui nous sera utile pour définir l'absence de divulgation de connaissance.

Définition Deux familles (X_m) et (Y_m) de variables aléatoires à valeurs dans $\{0,1\}^*$ indexées par $m \in \mathcal{L}$ sont dites *polynomialement indistinguables* sur $\mathcal{L}$ si pour tout entier $k > 0$ et pour tout mot m suffisamment grand,

$$\sum_{\alpha \in \{0,1\}^*} \left|\mathbb{P}(X_m = \alpha) - \mathbb{P}(Y_m = \alpha)\right| \leq |m|^{-k}$$

Exemple. Considérons pour tout mot m de taille $|m| = k$ les variables aléatoires X_m et Y_m définies par

$$\mathbb{P}(X_m = m') = \begin{cases} 2^{-k} & \text{si } |m'| = k \\ 0 & \text{sinon} \end{cases} \quad \mathbb{P}(Y_m = m') = \begin{cases} 0 & \text{si } |m'| \neq k \text{ ou } m' = 0^k \\ 2^{-k+1} & \text{si } m' = 1^k \\ 2^{-k} & \text{si } |m'| = k \end{cases}$$

On a alors

$$\sum_{\alpha \in \{0,1\}^*} \left|\mathbb{P}(X_m = \alpha) - \mathbb{P}(Y_m = \alpha)\right| = |2^{-|m|} - 0| + |2^{-|m|} - 2^{-|m|+1}| = 2^{-|m|+1}$$

Les familles (X_m) et (Y_m) sont donc polynomialement indistinguables, mais pas parfaitement.

Il est immédiat que si deux familles de variables aléatoires sont parfaitement indistinguables alors elles sont a fortiori polynomialement indistinguables.

3.2 Approximabilité et non divulgation de connaissance

On traduit ici l'idée qu'une famille de variable aléatoires puisse être simulée par une machine de Turing probabiliste.

Définition On dit qu'une famille $(X_m)_{m \in \mathcal{L}}$ de variables aléatoires à valeurs dans $\{0,1\}^*$ est parfaitement (polynomialement) *approximable* sur $\mathcal{L}$ s'il existe une machine de Turing probabiliste $\mathcal{M}$ en temps moyen polynomial telle que les familles de variables aléatoires (X_m) et $(\mathcal{M}(m))$ soient parfaitement (polynomialement) indistinguables.

Intuitivement on peut voir une preuve sans divulgation de connaissance comme un système de preuve dans lequel dans lequel tout ce qui peut être calculé (de manière efficace) après avoir interagi avec le prouveur correspond exactement à ce qui peut être calculé en possédant pour seule information la validité de l'assertion prouvée. La non-divulgation de connaissance caractérise la robustesse du prouveur face à un interlocuteur cherchant à obtenir des informations supplémentaires.

Sans perdre de généralité on peut supposer que la machine V possède une bande supplémentaire « d'historique » sur laquelle elle retranscrit toute l'information qu'elle obtient au cours du protocole, c'est à dire les mots échangés par le biais des bandes de communication. On note $\mathcal{H}(V, P, m)$ la variable aléatoire à valeurs dans $\{0,1\}^*$ qui associe le contenu de cette bande à la fin du protocole.

Définition Un système de preuve interactive (P, V) pour un langage $\mathcal{L}$ est dit *sans divulgation de connaissance* si pour tout machine de Turing interactive à temps moyen polynomial $\tilde{V}$ la famille de variables aléatoires $\Big(\mathcal{H}(\tilde{V}, P, m)\Big)_{m\in\mathcal{L}}$ est polynomialement approximable.

En d'autres termes, on est capable de construire une machine à temps moyen polynomiale qui simule de manière polynomialement indistinguable le protocole réel sur les entrées $m \in \mathcal{L}$. Ainsi, un attaquant est incapable de *distinguer polynomialement* un protocole simulé d'un vrai protocole pour des mots dans le langage, ce qui signifie intuitivement qu'il ne peut apprendre en interagissant avec P aucune information autre que l'appartenance au langage. C'est bien ce qu'on voulait.

Définition On note **ZK** (pour *Zero-Knowledge*) la classe des langages pour lesquels il existe une preuve interactive sans divulgation de connaissance.

4 Preuve de la résiduosité quadratique

On va fournir ici un exemple de langage appartenant à **ZK** lié à la cryptographie. Pour n entier naturel on note $\mathbf{Z}_n$ l'anneau quotient $\mathbf{Z}/n\mathbf{Z}$ et $\mathbf{Z}_n^\times$ le groupe des inversibles de l'anneau $\mathbf{Z}_n$. Quitte à faire les calculs modulo n on peut supposer que $\mathbf{Z}_n^\times = \{1 \leq k \leq n \in \mathbf{Z}_n \ ; \ k \wedge n = 1\}$. De même, le sous-groupe $\mathbf{Z}_n^{\times 2}$ des résidus quadratiques modulo n sera représenté par $\left\{u^2 \bmod n \ ; \ u \in \mathbf{Z}_n^\times\right\}$. Notons que $\mathbf{Z}_n^\times$ et $\mathbf{Z}_n^{\times 2}$ peuvent être générés en temps polynomial. Nous considérions le langage QR défini par

$$QR = \left\{(n, u) \ ; \ n \in \mathbf{N}^*, u \in \mathbf{Z}_n^{\times 2}\right\}$$

Commençons par décrire le protocole de manière informelle. On se donne un mot $(n, x) \in QR$ dont P veut convaincre V qu'il sait prouver son appartenance à QR, c'est à dire que x est un résidu quadratique modulo n. Pour cela on répète autant de fois que nécessaire les étapes suivantes :

- P communique à V un résidu quadratique $u \in \mathbf{Z}_n^{\times 2}$ choisi aléatoirement.
- V lit alors sur sa bande aléatoire un bit $b \in \{0, 1\}$ qu'il transmet à P.
- P renvoie à V une racine carrée v de $u \times x^b$ modulo n.
- V vérifie qu'on a bien $v^2 = u \times x^b$ modulo n.

Remarquons que si S est un ensemble fini on est capable grâce à la bande aléatoire de choisir un élément s de S de manière uniforme et polynomiale. On notera $\mathrm{Random}(S)$ la variable aléatoire associée.

Les machines de Turing P et V sont décrites respectivement par les algorithmes 1 et 2 données ci-après.

Pour commencer, le fait que la description des machines P et V ci-dessus donne un protocole interactif est immédiat, il suffit de se persuader qu'on est capable de déterminer si un entier k est dans $\mathbf{Z}_n^\times$ en temps polynomial en n et k, ce qui peut se faire à l'aide de l'algorithme d'Euclide.

Algorithm 1 Protocole pour P sur une entrée $(n, x) \in QR$ de taille l

pour $i = 1$ to l **faire**
  $u \leftarrow \text{Random}(\mathbf{Z}_n^{\times 2})$
  Envoyer u
  Recevoir un bit b
  $v \leftarrow \text{Random}(\{\text{racines carrées de } u \times x^b \text{ modulo } n\})$
  Envoyer v
fin pour

Algorithm 2 Protocle pour V sur une entrée (n, x) de taille l

si $n \geq 1$ et $x \in \mathbf{Z}_n^{\times}$ **alors**
  pour $i = 1$ to l **faire**
    Recevoir u
    si $u \notin \mathbf{Z}_n^{\times}$ **alors**
      Rejeter l'entrée
    fin si
    $b \leftarrow \text{Random}(\{0, 1\})$
    Envoyer b
    Recevoir v
    si $v^2 \bmod n \neq u \times x^b \bmod n$ **alors**
      Rejeter l'entrée
    fin si
  fin pour
  Accepter l'entrée
fin si

Lemme Le protocole (P, V) est un système de preuve interactive pour le langage QR.

Preuve On commence par traiter la consistance. Soit $(n, x) \in QR$; alors si u est un résidu quadratique modulo n c'est aussi le cas de $u \times x$ et donc P n'aura aucun mal à trouver un entier v tel que $v^2 = u \times x \mod n$ ou $v^2 = u \mod n$ selon les cas, ce qui permet de conclure la consistance.

Passons à la sûreté de la preuve. On suppose que V interagit avec une machine de Turing interactive $\tilde{P}$ arbitraire. Soient alors $n \in \mathbf{N}^*$ et $x \in \mathbf{Z}_n^\times$ qui ne soit pas un résidu quadratique modulo n. Si $u \in \mathbf{Z}_n^\times$, soit u est un résidu quadratique, soit $x \times u$ l'est. Quelque soit le choix de u, le succès du test ne dépend donc que du bit aléatoire. Avec l itérations, la probabilité d'acceptation est donc au mieux de 2^{-l}. De plus pour tout $k \geq 1$ on a $l^k = o(2^l)$ donc il existe $l_0 \in \mathbf{N}^*$ tel si $l \geq l_0$ on ait

$$\mathbb{P}(V_{\tilde{P}}(n, x) = 1) \leq 2^{-l} \leq l^{-k}$$

Ce qui achève la preuve de la sûreté.

On passe maintenant à la partie la plus intéressante : le protocole décrit pour la preuve de résiduosité quadratique est sans divulgation de connaissance.

Théorème Le protocole (P, V) est un système de preuve interactive parfaitement sans divulgation de connaissance pour le langage QR.

Preuve Soit $\tilde{V}$ une machine de Turing interactive à temps moyen polynomial. L'idée est de construire une machine de Turing probabiliste à temps polynomial capable de simuler les triplets (u, b, v) échangés au cours du protocole. Pour cela on peut supposer que $\tilde{V}$ possède une bande suplémentaire *d'historique* sur laquelle elle stocke toute l'information qu'elle obtient au cours du protocole, c'est à dire la suite des triplets (u, b, v) échangés.

On note $\text{Question}_{\tilde{V}}(n, x, hist, u)$ la variable aléatoire associant le bit issu du calcul de $\tilde{V}$ lorsqu'elle est initialisée avec (n, x) en entrée, que sa bande d'historique est $hist$, et que le prouveur lui envoie un résidu quadratique u sur sa bande de communication entrante.

On prend comme silmulateur la machine de Turing probabiliste décrite par l'algorithme 3 ci-dessous.

Algorithm 3 Simulateur $\mathcal{M}$ sur une entrée $(n, x) \in QR$ de taille l

$hist \leftarrow \epsilon$
pour $i = 1$ à l **faire**
  répéter
    $v \leftarrow \text{Random}(\mathbf{Z}_n^{\times 2})$
    $b \leftarrow \text{Random}(\{0, 1\})$
    $u \leftarrow v^2 / x^b \bmod n$
  jusqu'à ce que $b = Question_{\tilde{V}}(n, x, hist, u)$
  $hist \leftarrow hist \cdot (u, b, v)$
fin pour
Accepter l'entrée

Étant donné que v et b sont des variables aléatoires indépendantes, la probabilité que $b = Question_{\tilde{V}}(n, x, hist, u)$ est de $1/2$ et donc la boucle des lignes 3 à 7 s'éxécute en moyenne 2 fois. De plus les triplets (u, b, v) simulés sont corrects, au sens où on a par construction que u est un résidu quadratique modulo n avec v racine carré de $u \times x^b$ modulo n. D'après ce qui précède et l'indépendance des variables aléatoires à chaque itération de la boucle 2-9, une récurrence sur i montre que l'historique *hist* généré par $\mathcal{M}$ est une approximation parfaite de celui de $\tilde{V}$ lors d'un protocole réel avec P, ce qui permet de conclure.

Références

[GMW] Oded Goldreich, Silvio Micali et Avi Wigderson, *Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems*, Journal of the Association for Computing Machinery, Vol. 38, No. 1, juillet 1991.

[GMR] Shafi Goldwasser, Silvio Micali et Charles Rackoff, *The Knowledge Complexity of Interactive Proof Systems*, SIAM J. Comput. Vol. 18, février 1989.

[POUP] Guillaume Poupard, *Authentification d'entités, de messages et de clés cryptographiques : théorie et pratique*, thèse soutenue le 19 mai 2000.