Principe et Algorithme Cryptographiques : examen

première session / documents et calculatrice autorisés / durée 2h / 4 pages

22 mai 2018

Ce sujet est composé de 4 pages. Vos réponses doivent **absolument** être justifiées. Le sujet est constitué d'un seul problème et certaines questions s'enchainent, mais pas toutes. Les différentes sections contiennent des questions de difficultés variés (certaines sont de simples questions de cours). La plupart des questions admettent des réponses assez courtes, mais n'oubliez pas de les justifier — vous avez été prévenu(e). Le sujet est probablement trop long, mais ce n'est pas grave de ne pas aller jusqu'au bout.

Votre progression dans le TP sera relevée une dernière le vendredi 1er juin à 23 :59. Un rendu sur PROF sera demandé. Plus de précisions dans un email à venir.

On rappelle que la notation $\{m\}_k$ désigne (comme dans le cours) le chiffrement du message m avec la clef k. Ceci peut faire référence à un chiffrement symétrique (auquel cas k est la clef symétrique) ou bien à un chiffrement asymétrique (auquel cas k est la clef publique du destinataire).

Échange équitable et transfert inconscient de secrets

1 Introduction

On s'intéresse à un scénario ou Alice et Bob, deux journalistes d'investigation, possèdent des informations confidentielles et souhaitent se les échanger. Par exemple, Alice possède un secret s_a (révélation sur des comportement sexuellements déviants d'un certain chef d'État), tandis que Bob possède s_b (état d'avancement du programme extra-légal de développement de drones tueurs autonomes par un certain autre état). Bien sûr, ils aimeraient bien avoir le beurre et l'argent du beurre : obtenir le scoop de l'autre sans rien révéler en échange.

Du coup, on souhaite construire un protocole *équitable* : si Alice apprend bien le secret de Bob, alors Bob apprend bien le secret d'Alice (et vice-versa). Même si Alice et Bob se lancent dans l'exécution d'un tel protocole, aucun mécanisme cryptographique ne semble à même de garantir que les chaines de bits s_a et s_b qui vont être échangées possèdent vraiment une nature « sensible » (et intéressante d'un point de vue journalistique). On va donc supposer que c'est bien le cas, et que ni Alice ni Bob n'essayent directement de tricher en remplaçant leur secret par une chaine de bits aléatoire (on admettra que les conséquences, par exemple la ruine de sa réputation professionnelle, seraient trop grave pour tenter le coup).

Alice et Bob ne se sont jamais rencontrés et communiquent uniquement par internet. Pour commencer, on suppose qu'ils possèdent tous les deux la clef publique de l'autre. On considère dans un premier temps le protocole le plus simple possible pour qu'Alice et Bob échangent leur secret respectif.

Protocole E (*Échange de secrets*). Alice et Bob envoient simplement leur secret, chiffré, l'un après l'autre.

E1. [A révèle son secret.] Alice envoie $\{s_a\}_{pk_b}$ à Bob.

E2. [B révèle son secret.] Bob envoie $\{s_b\}_{pk_a}$ à Alice.

▷ **Question 1 :** Expliquer en une phrase quel genre de mécanisme Alice et Bob peuvent utiliser, préalablement à l'exécution du protocole, pour être sûrs d'avoir la bonne clef publique de l'autre.

La solution la plus courante est probablement l'utilisation de certificats, c'est-à-dire de faire en sorte qu'une autorité de certification reconnue garantisse l'authenticité des clefs publiques avec une signature numérique. Alternativement, Alice et Bob peuvent utiliser le réseau de confiance qui est la norme dans l'écosystème PGP.

▷ **Question 2 :** Dans le protocole E, si Alice entreprenait de tricher de manière grossière en envoyant du charabia pseudo-aléatoire à la place de son « vrai » secret, que se passerait-il ?

Bob le détecterait à l'issue de l'étape E1 (car il pourra déchiffrer ce qu'Alice lui envoie et récupérer s_a, et il refuserait probablement de continuer le protocole. Cette manière de tricher semble donc inopérante.

▷ **Question 3 :** Si jamais le lien réseau entre Alice est Bob était rompu après l'étape E1 mais avant l'étape E2, que se passerait-t-il ? Bob aurait-il intérêt à provoquer cet évènement ?

Bob aurait récupéré s_a, mais Alice n'aura pas s_b. C'est précisément la situation que Bob recherche, et il a donc tout intérêt à la provoquer. Bonus : il peut essayer d'argumenter qu'il est de bonne fois, que ce n'est pas de sa faute, que c'est un problème technique indépendant de sa volonté, etc.

2 Échange équitable de secrets

Le problème du protocole E c'est qu'il n'est pas équitable : il favorise Bob. Un échange équitable de secret doit placer les deux participants à égalité : à n'importe quel moment du protocole, il faut que les deux participants aient appris autant d'information sur le secret de l'autre. Par conséquent, si l'un des deux est coupé d'internet, décide de se retirer ou de se mettre à répondre n'importe quoi, alors il ne pourra pas en obtenir un avantage. Une approche naïve consiste à utiliser un protocole où les secrets sont transmis « progressivement ».

Protocole P (*Échange de secrets progressif*). Alice et Bob chiffrent leurs messages avec des clefs de session jetables de n bits, puis s'échangent les clefs un bit après l'autre.

P1. [A met son secret en gage.] Alice génère une clef de session K_a puis envoie $\{\{s_a\}_{pk_b}\}_{K_a}$ à Bob.

P2. [B met son secret en gage.] Bob génère une clef de session K_b puis envoie $\{\{s_b\}_{pk_a}\}_{K_b}$ à Alice.

P3. [Échange des clefs.] Pour chaque $1 \leq j \leq n$:
Alice envoie à Bob le j-ème bit de K_a.
Bob envoie à Alice le j-ème bit de K_b.

▷ **Question 4 :** Le chiffrement « externe » (avec les clefs de session) est-il symétrique ou asymétrique ?

Il s'agit d'un chiffrement symétrique. C'est plus simple et surtout on n'a pas besoin pour ça d'un chiffrement asymétrique. La clef de chiffrement est révélée à la fin, et le chiffrement ne sert qu'à imposer une sorte de mise en gage (en tout cas, un séquencement dans le temps des opérations).

▷ **Question 5 :** En 2018, quelle taille de clefs de session (n) semble raisonnable ?

Comme il s'agit d'un chiffrement symétrique, 128 bits (choisis aléatoirement) semblent suffisants. Une recherche exhaustive de 2^{128} opérations est actuellement hors de portée de l'humanité.

▷ **Question 6 :** Pourquoi a-t-on tenu à conserver le chiffrement asymétrique ?

Si on l'avait retiré, les messages secrets s_a et s_b seraient révélés à un adversaire passif qui espionnerait une exécution du protocole. Avec le chiffrement asymétrique, leur confidentialité est garantie.

▷ **Question 7 :** Que se passe-t-il si l'un des deux (disons Bob) décide d'interrompre le protocole au milieu de l'étape P3 ? Ceci entraine-t-il une rupture d'équité ?

Si le protocole est interrompu au milieu de l'étape P3, volontairement ou involontairement, les deux participants ont tous les deux appris j bits de la clef de session de l'autre. A priori, il n'y en a aucun qui a un avantage sur l'autre. Les deux peuvent tenter de reconstituer la clef de session de l'autre en effectuant une recherche exhaustive sur les bits manquant, mais ça va leur prendre le même temps à tous les deux (2^{n-j} possibilités à tester). De ce point de vue-là, le protocole semble équitable.

▷ **Question 8 :** Que se passe-t-il si Alice décide de tricher en répondant n'importe quoi lors de l'étape P3 ? Bob, qui lui exécute le protocole honnêtement, peut-il le détecter avant la fin ?

La « bonne » valeur de K_a est une chaine de bits aléatoire, donc si Alice lui envoie une *autre* chaine de bits aléatoire, Bob n'a aucun moyen de s'en rendre compte au fur-et-à-mesure qu'il reçoit les bits envoyés par Alice. Bob peut se rendre compte qu'Alice ne lui a pas envoyé la bonne valeur de K_a... à la fin, en tentant le déchiffrement et en observant du charabia à la place de s_a, mais c'est trop tard car il a déjà envoyé s_b et il s'est fait rouler.

La suite du sujet propose une manière de réparer ce protocole, ce qui n'est pas trivial.

3 Transfert inconscient

Le « (1-2) transfert inconscient » (*oblivious transfer*) est un mécanisme cryptographique où Alice possède deux secrets s_0, s_1, et où l'un de ces secrets est communiqué à Bob. Alice souhaite que Bob apprenne *un seul* de ses petits secrets. Bob lui ne souhaite pas qu'Alice sache *lequel* il souhaite recevoir. Cette section est consacrée à un protocole dû à Vanessa Vitse (*univ. Grenoble-Alpes*), mais non-publié à ce jour.

Protocole OT (*Transfert Inconscient*). Bob possède un bit secret i. Le but du protocole est qu'Alice « transfère » s_i à Bob, sans lui révéler l'autre secret s_{1-i} et sans qu'Alice apprenne i (c'est ça le côté « inconscient »). Le protocole fonctionne dans un groupe avec un générateur g modulo p. L'ordre du générateur est un nombre premier q. La description du groupe (p, g, q) est publique.

OT1. [Mise en gage.] Alice choisit aléatoirement deux nombres a_0 et a_1, puis transmet g^{a_0} et g^{a_1} à Bob.

OT2. [Masquage.] Bob pose $x \leftarrow g^{a_i}$, choisit un nombre aléatoire b, calcule $y \leftarrow x^b$ et enfin envoie y à Alice.

OT3. [Envoi chiffré.] Alice calcule $k_0 \leftarrow y^{(a_0^{-1})}$ et $k_1 \leftarrow y^{(a_1^{-1})}$ puis envoie $\{s_0\}_{k_0}$ et $\{s_1\}_{k_1}$ à Bob.

Préliminaires On se penche d'abord sur la tailles des données manipulées pendant le protocole.

▷ **Question 9 :** La sécurité de ce protocole repose implicitement sur le fait qu'un problème calculatoire classique de la cryptographie à clef publique est difficile. Lequel ?

Il s'agit du problème du logarithme discret.

▷ **Question 10 :** Quelle taille (en bits) doit avoir p pour offrir une bonne sécurité, en 2018 ?

Vu la complexité des meilleurs algorithmes pour le calcul du logarithme discret modulo p (qui est grosso-modo $\exp\left(\sqrt[3]{\frac{64}{9} + o(1)} \log^{1/3} n \log^{2/3} \log n\right)$ pour des modulus premiers de n bits), il semble que des nombres p de 2048 bits sont suffisants en 2018.

▷ **Question 11 :** Quelle devrait, idéalement, être la taille de q ?

Un adversaire pourrait tenter une recherche exhaustive sur le log discret, et ceci pourrait aboutir si l'ordre est trop faible. Comme il existe aussi des algorithmes (la méthode ρ, etc.) en $\sqrt{q}$, il faut choisir q d'au moins 256 bits.

Annotations implicites Comme dans le cours, la description du protocole omet de signaler les opérations « modulo » qui ont pourtant lieu dans chacune des trois étapes. Il s'agit ici de préciser les choses.

▷ **Question 12 :** Dans l'étape OT1, Alice effectue deux exponentiations modulaires. Quel est le modulus [1] ?

Les calculs du type g^x ont lieu modulo p. En effet, ce qu'on veut dans le fond c'est $(g^x) \bmod p$.

▷ **Question 13 :** Dans l'étape OT3, indiquer le modulus utilisé : a) pour les deux inversions modulaires et b) pour les deux exponentiations.

Les deux exponentiations ont lieu modulo p, mais pas les inversions modulaires. En effet, ici a_i est un *exposant*, et le « théorème de passage à l'exponentielle » dit qu'il faut prendre les exposants modulo *l'ordre du générateur*, qui est ici q. Ca aurait aussi marché modulo $p-1$, mais c'est moins rapide car c'est un nombre plus grand.

▷ **Question 14 :** Dans quel intervalle est-ce qu'Alice et Bob doivent choisir les nombres a_0, a_1 et b ?

Tous ces nombres sont utilisés comme exposants, donc il faut les choisir aléatoirement modulo q (rappel : le même exposant modulo q donne la même exponentielle modulo p).

Un détail manquant Dans l'étape OT3, Alice calcule des clefs puis effectue un chiffrement. Le texte ne dit pas s'il s'agit-il d'un mécanisme de chiffrement symétrique ou asymétrique... mais on peut, par le raisonnement, s'y retrouver quand même.

1. Rappel : quand on fait des calculs « modulo n », le modulus est n.

▷ **Question 15 :** Expliquer pourquoi, dans une exécution honnête du protocole, les deux clefs k_0, k_1 sont des nombres aléatoires.

Notons i_0 la valeur de i choisie par Bob. On a $k_i \equiv g^{a_{i_0} b {a_i}^{-1}} \mod q$, et donc :

$$k_{i_0} \equiv g^b \mod p,$$
$$k_{1-i_0} \equiv g^{b a_{i_0} a_{1-i_0}^{-1}}.$$

Version courte : dans une exécution honnête du protocole, a_0, a_1 et b sont aléatoires, donc les deux exposants sont eux aussi aléatoires, et donc les clefs sont bien aléatoires.

Version longue. Montrons ce petit supplément au cours : si g est d'ordre q premier et que $x \not\equiv 0 \mod q$, alors g^x est un autre générateur du groupe engendré par g.
En effet, comme $x \neq 0$, alors $g^x \neq 1$, et l'ordre de g^b est strictement supérieur à 1. Ensuite, $(g^x)^q = (g^q)^x = 1$ (car g est d'ordre q), donc l'ordre de g^x est un diviseur de q (lemme 6 du cours). Mais comme q est premier, et que l'ordre de g^x divise q, c'est que g^x est d'ordre q. L'ensemble $\langle g^x \rangle$ est contenu dans $\langle g \rangle$ et il a la même cardinalité, donc ces deux ensembles sont égaux. CQFD.
Si Bob choisit b aléatoirement, alors $k_{i_0} = g^b$ est uniformément distribué dans le groupe engendré par g.
L'idée ensuite, c'est que comme $a_0 \neq a_1$, on a $a_{i_0} a_{1-i_0}^{-1} \neq 1$, et donc $g^{a_{i_0} a_{1-i_0}^{-1}}$ engendre le même groupe que g, vu le raisonnement ci-dessus. Et comme b est aléatoire, il s'ensuit que $g^{\left(a_{i_0} a_{1-i_0}^{-1}\right)^b}$ est lui aussi uniformément distribué dans le groupe.

Ceci dit, les deux distributions de k_0 et k_1 ne sont pas indépendantes.

▷ **Question 16 :** Dans les mécanismes de chiffrement à clef publique vus en cours, la clef publique est calculée à partir de la clef privée. Pourrait-il être possible de faire l'inverse, c'est-à-dire de calculer la clef privée à partir de la clef publique ?

Non. Si c'était le cas, comme la clef publique est... publique, alors rien n'interdirait à n'importe qui *d'autre* de calculer ma propre clef « privée » à partir de ma clef publique. Le système n'offrirait aucune sécurité.

▷ **Question 17 :** Expliquer pourquoi (vu la réponse à la question précédente) le mécanisme de chiffrement utilisé à l'étape OT3 est forcément un mécanisme de chiffrement *symétrique*.

Si le chiffrement de l'étape OT3 était un chiffrement asymétrique, alors k_0 et k_1 devraient être des clefs publiques des destinataires.
Or, on a vu que k_0 et k_1 sont les produits de calculs précédents, et qu'ils ont des valeurs « imposées ». Ceci exclut donc leur utilisation comme clef publique : vu la question précédente, il n'y a pas de mécanisme de chiffrement à clef publique dans lequel il est possible de choisir entièrement la clef publique, et de calculer la clef privée à partir de là.
On a donc affaire à un chiffrement symétrique, où les clefs (symétriques) peuvent être choisies arbitrairement.

Sécurité et correction du protocole On se penche maintenant sur les caractéristiques fonctionnelles du protocole.

▷ **Question 18 :** Que se passerait-il si Alice choisissait $a_i = 0$ dans l'étape OT1 ? Quel intérêt Alice aurait-elle à vouloir faire ceci ? Bob peut-il le détecter et l'empêcher efficacement ?

Supposons que $a_0 = 1$. Dans ce cas-là, on aura $(g^{a_0})^b = 1$ et $(g^{a_0})^b \neq 1$, or ceci est la valeur que Bob renvoie à Alice dans l'étape OT2. Ceci révèlerait donc à Alice lequel de ses deux secrets intéresse Bob, or le protocole est censé l'empêcher.
Heureusement, Bob peut le détecter facilement, car alors il reçoit $g^{a_0} = 1$.

▷ **Question 19 :** Même question avec $b = 0$.

Bob renvoie alors toujours la même chose dans l'étape OT2 (et donc Alice reçoit littéralement zéro information sur le choix de Bob). Pire, on a alors $k_0 = k_1 = 1$. Du coup, Bob peut obtenir les deux secrets d'Alice d'un seul coup !
Heureusement, Alice peut le détecter que $b = 0$, car elle reçoit alors $y = 1$, et interrompre le protocole à ce moment-là.

▷ **Question 20 :** Expliquer pourquoi est-ce qu'Alice n'apprend pas i (le choix de Bob) à l'issue de l'étape OT2, et ceci quelle que soit la puissance de calcul dont elle dispose.

L'argument principal est que si Bob choisit b uniformément au hasard modulo q, alors la valeur y envoyée à Alice dans l'étape OT2 est uniformément distribuée dans le groupe engendré par g, et ceci quelles que soient les valeurs de a_0, a_1 choisies par Alice et le choix i de Bob.

Argument complet : g^{a_0} et g^{a_1} engendrent le même groupe que g (cf. ci-dessus), donc comme b est uniformément distribué parmi les exposants possibles, alors $(g^{a_i})^b$ est uniformément distribué dans le groupe.

Alice reçoit donc un nombre aléatoire, qui par définition ne lui révèle pas d'information sur i (plus précisément, elle a exactement la même probabilité de recevoir n'importe quelle valeur, et ceci quel que soit le choix i de Bob).

▷ **Question 21 :** Expliquer ce que Bob doit faire pour récupérer s_i (et donc réaliser concrètement le « transfert »).

Bob doit calculer k_i puis effectuer le déchiffrement du (bon) message renvoyé par Alice avec la clef symétrique k_i. Le truc, c'est que k_i vaut g^b (cf. ci-dessus), or Bob connaît b.

▷ **Question 22 :** Expliquer informellement pourquoi Bob n'apprend pas s_{1-i}.

Pour déchiffrer l'autre message, Bob devrait réussir à calculer l'autre clef symétrique, c'est-à-dire $k1-i = g^{b\left(a_i a_{1-i}^{-1}\right)}$. Il connaît b, mais pas les deux valeurs a_0 et a_1, or c'est ça qui coince.
Il connaît g^{a_0} et g^{a_1}, or il lui faudrait réussir à calculer $g^{a_i a_{1-i}^{-1}}$. C'est une variante du problème Diffie-Hellman calculatoire (avec l'inversion en plus), et ça n'a pas l'air tellement plus simple que de calculer un logarithme discret...

▷ **Question 23 :** Indiquer de quelle(s) manière(s) le protocole échouerait si jamais un algorithme très efficace pour résoudre le problème calculatoire difficile sous-jascent venait à être découvert.

Si Bob en dispose, il pourrait obtenir a_0 et a_1 lors de l'étape OT1, donc calculer les deux clefs et obtenir les deux messages lors de l'étape OT3.
Par contre, ça ne peut pas aider Alice à apprendre le choix de Bob (cf. ci-dessus).

4 Échange équitable de paires de secrets

Armés d'un protocole de transfert inconscient, on av pouvoir résoudre le problème de l'échange équitabel de secrets, mais avant cela, il nous faut passer par une étape intermédiaire : l'échange de *paires secrètes*. On va noter $\texttt{OT}(A, B, x, y)$ pour dénoter un protocole de transfert inconscient (par exemple celui de la section précédente), où A transfère à B l'un de ses deux secrets x et y. Cette section est consacrée au protocole suivant :

Protocole PE (*Échange équitable de paires*). Alice et Bob possèdent tous les deux n *paires* de chaines de bits secrètes, toutes de la même taille ℓ. Alice possède $(u_i, v_i)_{1\leq i\leq n}$ tandis que Bob possède $(x_i, y_i)_{1\leq i\leq n}$. Le but du protocole est de s'échanger les paires équitablement.

PE1. [Transfert inconscient.] Alice effectue le transfert inconscient $\texttt{OT}(A, B, u_i, v_i)$ pour $1 \leq i \leq n$.
Bob effectue le transfert inconscient $\texttt{OT}(B, A, x_i, y_i)$ pour $1 \leq i \leq n$.
Dans les deux cas, le « receveur » choisit aléatoirement s'il veut recevoir le premier ou le deuxième élément de la paire.

PE2. [Transmission progressive.] Pour chaque $1 \leq j \leq \ell$:
Alice envoie à Bob le j-ème bit de u_i et v_i (pour tout $1 \leq i \leq n$).
Bob envoie à Alice le j-ème bit de x_i et y_i (pour tout $1 \leq i \leq n$).

▷ **Question 24 :** Justifier qu'à la fin d'une exécution (honnête) du protocole, les paires ont bien été échangées (Alice connaît les paires de Bob et réciproquement).

Les paires sont entièrement révélées, bit par bit, lors de l'étape PE2.

▷ **Question 25 :** Que va-t-il se passer si Alice ou Bob répond n'importe quoi à un moment dans l'étape PE2 ? L'autre peut-il le détecter ? Quel rôle joue le nombre n de paires dans le processus ?

Supposons qu'Alice se mette à tricher à un moment donné, en révélant incorrectement un des bits d'une de ses paires. Le point c'est que Bob a une bonne chance de s'en rendre compte.
En effet, lors de l'étape PE1, Bob a choisi aléatoirement un des « côtés » de chacune des n paires d'Alice. Alice ne peut pas prévoir ces choix aléatoires. De plus, le protocole de transfert inconscient ne révèle pas à Alice de quel côté il s'agit à chaque fois (c'est là toute la subtilité).
Si Alice révèle incorrectement un des bits, elle a donc une chance sur deux que Bob possède le côté correspondant de la paire en question et puisse détecter que le bit révélé par Alice lors de l'étape OT2 n'est pas celui qu'elle lui a transmis lors de l'étape OT1.
Chaque bit modifié par Alice multiplie donc par 2 la probabilité de détection de la fraude par Bob. S'il y a n paires, et qu'Alice souhaite que Bob n'en obtienne aucune correctement, il faut qu'elle les modifie toutes sur au moins un bit. Or, du coup, la probabilité que Bob détecte la fraude est de $1 - 1/2^n$.

5 Un protocole d'échange de secrets équitable

On est finalement en mesure de réparer le protocole PE de l'introduction.

Protocole P' (*Échange de secrets équitable*). Alice et Bob chiffrent leurs messages avec des clefs de session de n bits, puis s'échangent les clefs de manière équitable grace au protcole d'échange équitable de paires.

P'1.[A met son secret en gage.] Alice génère une clef de session aléatoire K_a puis envoie $\{\{s_a\}_{pk_b}\}_{K_a}$ à Bob.

P'2.[B met son secret en gage.] Bob génère une clef de session aléatoire K_b puis envoie $\{\{s_b\}_{pk_a}\}_{K_b}$ à Alice.

P'3.[Échange équitable des clefs.] Alice génère n paires (u_i, v_i) en choisissant u_i aléatoirement et en posant $v_i \leftarrow K_a \oplus u_i$.
Bob génère également n paires (x_i, y_i), où x_i est aléatoire et $y_i \leftarrow K_b \oplus x_i$.
Finalement, Alice et Bob exécutent le protocole PE de la section précédente pour s'échanger équitablement leurs paires respectives.

▷ **Question 26 :** En supposant que les participants exécutent le protocole honnêtement, comment font-ils pour obtenir leurs secrets respectifs ?

A la fin de l'étape P'3, si au moins une paire a été échangée correctement, alors le XOR des deux côtés de la paire révèle la clef de session utilisée lors des deux premières étape, et permet le dechiffrement du secret.

▷ **Question 27 :** Si le protocole est interrompu avant le début de l'étape P'3, un des deux participants est-il avantagé ?

Avant l'étape 3, les deux participants se sont échangés des messages chiffrés avec des clefs de sessions aléatoires. Tant que le mécanisme de chiffrement est sûr (c.a.d. possède une forme de securité sémantique), alors les échanges en P'1 et P'2 ne révèlent pas les secrets. Du coup, aucun des deux participants n'est avantagé et le protocole reste équitable.

▷ **Question 28 :** Supposons que l'un des deux participants décide de tout arrêter après l'étape PE1 du sous-protocole d'échange équitable des paires. Peut-il récupérer le secret de l'autre ?

Non. À ce stade, Alice (disons) possède un des deux côtés de chaque paire de Bob, mais il faut les deux côtés d'une même paire pour reconstituer la clef de session.
En effet, si Alice récupère x_i lors du transfert inconscient, elle a appris un nombre aléatoire, donc zéro information sur K_b. Si elle récupère y_i, alors elle a obtenu la clef K_b chiffrée avec un masque jetable aléatoire et inconnu, donc elle n'a rien non plus.
A ce stade-là, aucun des deux participant n'est avantagé, car personne n'a rien.

▷ **Question 29 :** Alice peut toujours décider d'interrompre le protocole au milieu de l'étape PE2 de l'échange de paires. Si elle dispose d'une grande puissance de calcul, comment peut-elle essayer d'apprendre le secret de Bob malgré tout ?

Si le protocole est interrompu (ou dévie de son cours normal) au milieu de l'étape PE2, alors chacun des deux participants est capable de reconstituer la même quantité, disons j, de bits de la clef de session utilisée par l'autre.
Si Alice possède une plus grande puissance de calcul, par contre, elle peut effectuer une recherche exhaustive sur les bits restants de K_b plus vite que Bob ne pourrait le faire pour K_a. Ceci peut lui permettre d'apprendre le secret de Bob en un temps bien plus faible qu'il ne faudrait à ce dernier pour apprendre celui d'Alice.

Ceci, par contre, est quasiment inévitable.