calcul partagé à base d'encryption probabiliste
par Claude Crépeau

1 Chiffrement probabiliste à la Goldwasser-Micali

Soit $n = pq$ tel que p et q sont premiers et $p \equiv q \equiv 3 \pmod 4$. Donc $\left(\frac{-1}{n}\right) = +1$ mais $-1 \in NRQ_n$.

La clef publique du système est n et la clef secrète est p, q. L'algorithme de chiffrement d'un bit b est

$$E(b) = (-1)^b r^2 \bmod n$$

où r est choisit au hasard tel que $pgcd(r, n) = 1$ et le déchiffrement est

$$D(c) = (b, r) \text{ tels que } c = (-1)^b r^2 \bmod n.$$

Puisque $\left(\frac{-1}{n}\right) = +1$, on aura toujours $\left(\frac{E(b)}{n}\right) = +1$ pour tous b, r, mais également $E(b) \in RQ_n$ si $b = 0$ et $E(b) \in NRQ_n$ si $b = 1$. On ne connait aucun algorithme efficace pour décider si $E(b)$ est dans RQ_n ou NRQ_n quand on ne dispose pas des facteurs de n.

2 Transfert inconscient

Alice possède deux bits d'input b_0, b_1 qu'elle met en gage avec Bob par $E_A(b_0), E_A(b_1)$. Bob, d'autre part, met en gage son bit de choix c par $E_B(c)$ et la future réponse du calcul b_c par $E_B(z)$ pour un bit z aléatoire. La réponse finale du calcul sera $\pm E_B(z)$.

$E_A(b_0)$	$E_B(c)$
$E_A(b_1)$	$E_B(z)$

Afin d'obtenir b_c, Bob prépare une "question" $E_A(b_c \oplus z) = E_A(b_c)E_A(z)$ à partir de $E_A(b_c)$ et z.

$E_B(c)$
$E_B(z)$
$E_A(b_c \oplus z)$

Si cette question est construite correctement, il suffit à Alice de décrypter cette question par $D_A(E_A(b_c \oplus z))$ pour que chacun sache si $z = b_c$ ou non: si $E_A(b_c \oplus z)$ est un résidu quadratique alors $z = b_c$ et $E_B(z)$ est la réponse correcte du calcul, sinon $-E_B(z)$ est correcte.

Mais pourquoi Alice croirait-elle que la question est construite correctement? Bob peut la convaincre que la question est correcte sans lui faire savoir si elle porte sur b_0 ou sur b_1. À cette fin, Bob choisit des bits aléatoires $c_1, c_2, ..., c_s, z_0^1, z_0^2, ..., z_0^s, z_1^1, z_1^2, ..., z_1^s$. Il en obtient s paires de questions construites comme suit:

$E_B(c_1)$	$E_B(\bar{c}_1)$	$E_B(c_2)$	$E_B(\bar{c}_2)$		$E_B(c_s)$	$E_B(\bar{c}_s)$
$E_B(z_{c_1}^1)$	$E_B(z_{\bar{c}_1}^1)$	$E_B(z_{c_2}^2)$	$E_B(z_{\bar{c}_2}^2)$	...	$E_B(z_{c_s}^s)$	$E_B(z_{\bar{c}_s}^s)$
$E_A(b_{c_1} \oplus z_{c_1}^1)$	$E_A(b_{\bar{c}_1} \oplus z_{\bar{c}_1}^1)$	$E_A(b_{c_2} \oplus z_{c_2}^2)$	$E_A(b_{\bar{c}_2} \oplus z_{\bar{c}_2}^2)$		$E_A(b_{c_s} \oplus z_{c_s}^s)$	$E_A(b_{\bar{c}_s} \oplus z_{\bar{c}_s}^s)$

Dans chaque paire, une question porte sur b_0 et l'autre sur b_1. Alice ne sais pas a priori laquelle est laquelle car elle ne connais pas c_i.

$E_B(c_i)$	$E_B(\bar{c}_i)$
$E_B(z_{c_i}^i)$	$E_B(z_{\bar{c}_i}^i)$
$E_A(b_{c_i} \oplus z_{c_i}^i)$	$E_A(b_{\bar{c}_i} \oplus z_{\bar{c}_i}^i)$

Ensuite, pour chacune de ces paires, Alice va choisir soit de vérifier explicitement que ses question sont construites correctement, soit de vérifier qu'elle contient une question équivalente à la question originale. La première possibilité est démontrée en dévoilant les informations suivantes:

$D_B(E_B(c_i))$	$D_B(E_B(\bar{c}_i))$
$D_B(E_B(z_{c_i}^i))$	$D_B(E_B(z_{\bar{c}_i}^i))$
$D_A(E_A(z_{c_i}^i))$	$D_A(E_A(z_{\bar{c}_i}^i))$

et la seconde, en dévoilant les informations suivantes:

$D_B(E_B(c_i \oplus c))$	$D_B(E_B(\bar{c}_i \oplus c))$
$D_B(E_B(z^i_{c_i} \oplus z))$	$D_B(E_B(z^i_{\bar{c}_i} \oplus z))$
$D_A(E_A(z^i_{c_i} \oplus z))$	$D_A(E_A(z^i_{\bar{c}_i} \oplus z))$

Dans le cas où Bob a construit la question originale correctement et les s paires de questions correctement, il n'aura aucune difficulté à dévoiler les informations correctes à Alice. Par contre dans le cas contraire, quand Bob n'a pas construit la question originale correctement il devra choisir pour chaque paire de questions s'il la construira correctement ou non.

- s'il construit une paire de questions correctes il ne pourra montrer qu'elle contient une question équivalente à la question originale (qui n'est pas correcte)
- s'il construit une paire de questions incorrectes il ne pourra montrer explicitement que ses question sont construites correctement.

Dans tous les cas, Bob aura au moins une probabilité $1/2$ de ne pas savoir satisfaire aux exigeances d'Alice. Bob a donc au plus une probabilité $1/2^s$ de satisfaire Alice si sa question originale n'est pas correctement construite.

2.1 calcul partiel

On utilise le transfert inconscient pour obtenir à partir d'un bit a appartenant à Alice connu publiquement par $E_A(a)$ et d'un bit b appartenant à Bob connu publiquement par $E_B(b)$, deux nouveau bits a' appartenant à Alice connu publiquement par $E_A(a')$ et b' appartenant à Bob connu publiquement par $E_B(b')$, tels que

$$a \wedge b = a' \oplus b'.$$

Pour ce faire, Alice choisit un bit a' au hasard puis calcule $a'' = a' \oplus a$ et publie $E_A(a')$, $E_A(a'') = E_A(a)E_A(a')$. Alice et Bob font le transfert inconscient avec $b_0 = a'$, $b_1 = a''$ et $c = b$. Soit b' le bit de sortie obtenu par Bob lors du transfert. Remarquons que si $b = 0$ alors $a \wedge b = 0$, et que $b' = a'$ et donc $a \wedge b = 0 = a' \oplus b'$. D'autre part, si $b = 1$ alors $a \wedge b = a$, et que $b' = a''$ et donc $a \wedge b = a = a' \oplus b'$.

2.2 calcul partagé

Le calcul partagé s'effectue en phases. La phase initiale permet à chaque participant de mettre en gage ses bits d'inputs. À travers toutes les étapes du calcul, chaque bit sera partagé par Alice et Bob de la façon suivante: à chaque bit x sont associés des bits x_A, x_B tels que $x = x_A \oplus x_B$. Quand Alice veut mettre en gage des bits précis, Bob utilisera $x_B = 0$ et vice versa.

exemple: Si Alice veut mettre en gage un bit b elle annonce à Bob $E_A(b_A)$ pour $b_A = b$, et Bob annonce $E_B(b_B)$ pour $b_B = 0$ et montre que c'est bien le cas en révélant $D_B(E_B(0))$.

Tous les bits d'input d'Alice et de Bob sont mis en gage ainsi. Alice et Bob s'étant mis d'accord sur un circuit (fait de portes $\wedge$ et $\neg$) permettant de calculer la fonction voulue il feront ensuite l'évaluation du circuit porte par porte.

Supposons que (b^1_A, b^1_B) soit un bit partagé. S'ils veulent en obtenir la négation il suffit de remplacer $E_A(b^1_A)$ par $-E_A(b^1_A)$ ou $E_B(b^1_B)$ par $-E_B(b^1_B)$. Supposons maintenant que (b^1_A, b^1_B) et (b^2_A, b^2_B) soient deux bits partagés desquels Alice et Bob voudraient obtenir un troisième bit (b^3_A, b^3_B) tel que

$$(b^3_A \oplus b^3_B) = (b^1_A \oplus b^1_B) \wedge (b^2_A \oplus b^2_B).$$

Il suffit d'observer que $(b^1_A \oplus b^1_B) \wedge (b^2_A \oplus b^2_B) = (b^1_A \wedge b^2_A) \oplus (b^1_A \wedge b^2_B) \oplus (b^1_B \wedge b^2_A) \oplus (b^1_B \wedge b^2_B)$. S'ils évaluent ces valeurs par la méthode ci-haut on aura des $c^1_A, c^2_A, c^3_A, c^4_A$ et $c^1_B, c^2_B, c^3_B, c^4_B$ tels que

$$\begin{aligned}(b^1_A \oplus b^1_B) \wedge (b^2_A \oplus b^2_B) &= (c^1_A \oplus c^2_A) \oplus (c^3_A \oplus c^1_B) \oplus (c^2_B \oplus c^4_A) \oplus (c^3_B \oplus c^4_B) \\ &= (c^1_A \oplus c^2_A \oplus c^3_A \oplus c^4_A) \oplus (c^1_B \oplus c^2_B \oplus c^3_B \oplus c^4_B).\end{aligned}$$

Il s'agit donc de poser $b^3_A = c^1_A \oplus c^2_A \oplus c^3_A \oplus c^4_A$ et $b^3_B = c^1_B \oplus c^2_B \oplus c^3_B \oplus c^4_B$.