TD - Générateur pseudo-aléatoire cryptographiquement sûr

Le générateur BBS (Blum Blum Shub) est un générateur crytographiquement sûr qui fonctionne comme suit:

- On choisit deux nombres premiers $p = 4.k_1 + 3$ et $q = 4.k_2 + 3$. On calcule l'entier de Blum $n = pq$.
- On choisit un entier $x < n$ aléatoire et premier avec n.
- On pose $x_0 = x^2 \bmod n$ et, pour $i \geq 1$, $x_i = x_{i-1}^2 \bmod n$.

Le $i^{\text{ème}}$ bit b_i pseudo-aléatoire est alors le bit de poids faible de x_i, i.e. $b_i = x_i \bmod 2$.
Ce générateur est cryptographiquement sûr: sa sécurité (admise) repose sur la difficulté de factoriser n. Ce générateur est conjecturé sûr à gauche et à droite.

Question 1. On suppose que Bob, qui connait p et q, a choisi x_0 et veut calculer efficacement x_i.

1. Soit $u_i = 2^i \mod (p-1)(q-1)$; montrer que $x_i = x_0^{u_i} \mod n$.
2. En déduire un algorithme qui prend en entrée i, x_0, p, q et n et qui génère en sortie le bit b_i. Donner le coût de cet algorithme.
3. Quel est l'intérêt de cette propriété ?

Correction :

1. *Soient* $2^j \mod \phi(n)$ *et* $\rho = 2^j \div \phi(n)$ *respectivement le reste et quotient dans la division euclidienne de* 2^j *par* $\phi(n)$*:* $2^j = (2^j \mod \phi(n)) + \rho.\phi(n)$*. On a:*

$$x_j = x_{j-1}^2 \mod n = x_0^{2^j} \mod n = x_0^{2^j \bmod \phi(n)}.x_0^{\rho.\phi(n)} \mod n.$$

En outre, comme x *et* n *sont premiers entre eux et* $x_0 = x^2 \mod n$*, alors* x_0 *et* n *sont premiers entre eux. Le théorème d'Euler s'applique et on a:* $x_0^{\phi(n)} \mod n = 1$*. Donc,* $x_0^{\rho.\phi(n)} = 1 \mod n$ *ce qui aboutit à:*

$$x_j = x_0^{2^j} \mod n = x_0^{2^j \bmod \phi(n)} \mod n.$$

2. *L'algorithme est:*
   `ui := FastExponentiation(`$2, i, (p-1)(q-1)$`) ;`
   `xi := FastExponentiation(`x_0`, ui, `x`) ;`
   `return LSB(xi) ;`
   Le coût est celui de deux exponentiations modulaires modulo deux entiers de $\log_2 n$ *bits, soit* $\tilde{O}(\log^2 n)$.
   Plus précisément: The cost is the one of two modular exponentiations with two moduli of $\log_2 n$ *bits: thus* $\Theta(\log_2 i + \log^2 n)$ *multiplications, each multiplications costing* $\tilde{0}(\log n)$*. Considering large primes* p *and* q *(more than one thousand bits), we may assume* $\log_2 i \leq \log_2 n$*, which leads to a cost* $\tilde{0}(\log^2 n)$.
3. *La propriété permet de générer directement et rapidement* b_i *à partir de* x_0 *sans avoir besoin de générer les bits intermédiaires.*

Question 2. On suppose que Bob, qui connait p et q, connait x_i mais pas x_0.

1. Montrer que $u_i = 2^i \mod (p-1)$ est premier avec l'entier $\frac{p-1}{2}$.
2. $\star$ Soit v_i l'inverse de u_i modulo $\frac{p-1}{2}$. Donner un algorithme permettant de calculer $x_0 \mod p$ à partir de $x_i \mod p$ et en utilisant v_i.
3. En déduire un algorithme qui prend en entrée i, x_i, p, q et n et qui génère en sortie x_0.

Correction :

1. *On a $p = 4k+3$ donc $\frac{p-1}{2} = 2k+1$ est un entier impair. Il est donc premier avec 2^i. Donc, d'après Bezout, $\exists a, b$ integers such that: $a.2^i + b.\frac{p-1}{2} = 1$ (E). On a aussi $2^i = \rho.(p-1) + (2^i \mod p-1)$ où $\rho = (2^i)$ div $(p-1)$ est un entier. En remplaçant dans (E), on obtient: $a.(2^i \mod p-1) + (2a\rho + b).\frac{p-1}{2} = 1$. On en déduit (d'après Bezout car a et $2a\rho+b$ sont entiers) que $(2^i \mod p-1)$ et $\frac{p-1}{2}$ sont premiers.*
2. *Let $v_i = u_i^{-1} \mod \frac{p-1}{2}$, which exists from previous question. Then $u_i.v_i = 1 + a.\frac{p-1}{2}$ with a integer. Thus, let $w = x_i^{v_i} \mod p = x_0^{1+a.\frac{p-1}{2}} \mod p$. Since $x_0 = x^2 \mod n$ and $n = pq$, we have $x_0 = x^2 \mod p$; so $w = x_i^{v_i} \mod p = x^{2+a.(p-1)} \mod p = x^2 \mod p = x_0$.*
3. *Donner un algorithme permettant de calculer $x_0 \mod p$ à partir de $x_i \mod p$ et en utilisant v_i.*

Question 3. Le protocole de chiffrement de Blum-Goldwasser fonctionne comme suit. Pour envoyer un message $M = [M_1, \ldots, M_t]$ de t bits à Bob, Alice procède comme suit. Alice choisit une valeur x_0 aléatoire secrète; à partir de x_0, elle génère avec le générateur BBS et la clef publique n de Bob une suite de t bits $B = [b_1, \ldots, b_t]$ pseudo-aléatoires et elle envoie à Bob le message chiffré $[M', x_t]$ où :

- $M' = [M_1 \oplus b_1, \ldots, M_t \oplus b_t]$ est le ou exclusif de M et B;
- x_t est le t^{ime} itéré de la suite (x_i) générée

Justifier que cet algorithme est sûr. En utilisant les questions précédentes, donner l'algorithme qui permet à Bob de déchiffrer efficacement le message reçu.