Université de lorraine

Faculté Des Sciences

M1 Informatique

Cryptographie

Signature de schnorr

Compte rendu

Ipseiz Angela

Li-Cho Dylan

Nicolas Maxime

Question 1:

Le langage choisi est le C, la librairie utilisée permettant de gérer des nombres entiers de grande taille est GMP. (libgmp)

La bibliothèque permet de générer de grands nombres aléatoires, de déterminer si un nombre est premier ou non, faire des opérations tel que les multiplications, additions, soustractions, divisions, congruences, tests de divisibilité (savoir si d divise n), calculs exponentielles, racines carrés, puissances, ainsi que des opérations logiques (and, or, xor, ...).

Question 2:

Un nombre aléatoire est cryptographiquement sûr si :

1. Son générateur de nombre aléatoire est cryptographiquement sûr, c'est-à-dire s'il utilise une entropie externe, la plus élevée possible et que tous les nombres aléatoires ont la même probabilité d'être générés.
2. S'il est imprédictible : connaissant les n premiers bits, est-il possible de prédire le bit n+1 suivant ?
3. La période de son générateur est suffisamment longue.

Question 3:

On utilise le test de primalité de Miller-Rabin.
Un test probabilistes est un test qui s'autorise une certaine marge d'erreur et par conséquent qui ne donne pas forcément une réponse juste.
Sous GMP, la fonction "int mpz_probab_prime_p (const mpz_t n, int reps)" permet de faire ce test : Elle renvoie 2 si le nombre est premier à coup sûr, 1 si le nombre est probablement premier et 0 sinon.

Question 4 :

Sous GMP la fonction "mpz_powm" permet de faire ceci.

Question 7:

Une fonction de hachage cryptographiquement sûr est une fonction qui doit:

- Être rapide à utiliser.
- Résister à la collision: c'est dur (en 2^(nombres de bit/2)) de trouver x' différent de x tel que H(x)=H(x').
- Résister à la préimage: c'est dur (en 2^nombres de bit) de trouver un x tel que H(x)=y.
- Résister à la 2nd préimage: soit un x donné, c'est dur (en 2^nombres de bit) de trouver x' différent de x tel que H(x) = H(x').

Etant donné que la sortie fait au moins 256 bits on peut utiliser au minimum SHA-256, on peut aussi utiliser SHA-512 ou SHA-3.
Il existe quelques librairies en C permettant d'employer ces fonctions, ici nous utiliserons SHA256 avec la librairie Tinycrypt de Intel. (https://github.com/intel/tinycrypt).