Security models: Part Security proofs [JL Roch]

Preamble

- This assignment is part A/Security Proof (duration: 1.30 hours) of the 2008 examination.
- This assignment is to be performed individually by each student in less than 3 hours (limited time but not controlled); it is to be returned before the examination of Security Models.
  For the answers you will submit in your assignment, you have to **limit your time to 3 hours**. Of course, if you have not finished after this delay, you are encouraged to spend additional time to train yourself and acquire the related content: but you must not include the answers found after the delay of 3 hours in the assignment you submit.
- The mark for this assignment will account for 50% of the mark of the continuous control for the part Security proofs.
- All exercises are independent.
- Your answers should be **short** but **clearly argued or commented**.

PART A - Security proofs [J-L. Roch]

Exercise 1
Perfect secrecy (points: 25%) Alice and Bob communicate through a public channel that enables to send and to receive only sequences of symbols from a set V of $L \geq 2$ symbols: $V = \{s_0, \ldots, s_{|L|-1}\}$.

The plaintext messages of Alice are written using characters in V.

It is assumed that Alice and Bob have previously agreed on a secret key $K = \{k_1, \ldots, k_m\} \in V^m$ with m very long.
Alice wants to send a secret message M of $n < m$ symbols to Bob using a Vernam cipher.

1. Only in this question, we assume that there exists a binary operator $\star$ in V such that $(V, \star)$ is a group. Describe briefly the coding and the decoding of a message. How to choose K to guarantee perfect secrecy (i.e. that the crypto-system is unconditionally secure) ?
2. In all the sequel, V is the roman alphabet with cardinal $L = 26$: $V = \{'A', 'B', \ldots, 'Z'\}$.
   Explain how to define a group operator $\star$ on V in order to implement Vernam cipher.
3. Now, in order to generate a cryptographically secure shared secret key $K \in \{'A', 'B', \ldots, 'Z'\}^n$, Alice and Bob want to use a Blum-Blum-Shub pseudo-random generator. Explain how they proceed: precise the size of public modulo of the BBS generator (the Blum integer in BBS) and how Alice and Bob generate the alphabetic key K. On what condition the resulting cryptosystem is unconditionnaly secure?

Exercise 2
Hash function based on RSA (points: 25%)
Let $p = 2p_1 + 1$ and $q = 2q_1 + 1$ be two secret large primes such that p_1 and q_1 are primes. Let $n = pq$. Let α be an element of maximal order in $\mathbb{Z}_n^*$, i.e.: $(\alpha \mod n, \alpha^2 \mod n, \ldots, \alpha^{2p_1q_1} = 1 \mod n)$ is a subgroup of $\mathbb{Z}_n^*$ of cardinal $2p_1q_1$.
We consider the following compression function: $h : \{1, \ldots, n^2\} \to \{1, \ldots, n-1\}$ defined by: $h(x) = \alpha^x \mod n$.

1. Let x_1 and x_2 be a collision for h: $h(x_1) = h(x_2)$. Prove that $(x_1 - x_2)$ is a multiple of $2p_1q_1$. **Hint:** *note that* $h(x_1).h(x_2)^{-1} = 1 \mod n$.

2. In all the sequel, it is assumed that x_1, x_2, x_3 are known collisions for h: $h(x_1) = h(x_2) = h(x_3)$. Moreover, it is assumed that $gcd(x_1 - x_2, x_1 - x_3) = 2p_1.q_1$.
Provide a polynomial time algorithm that takes in input n, x_1, x_2, x_3 and returns the factor p and q of n.

3. Justify that the assumption $gcd(x_1 - x_2, x_1 - x_3) = 2p_1.q_1$ is reasonnable.

4. What can be deduced about the hash function h ?

Exercise 3
Quadratic residues and CSPRNG. (points: 50%)

Question 1. Quadratic residues. (points: 25%)
Let p be an odd prime. A number $a \in \mathbb{Z}_p^*$ is a *quadratic residue modulo* p if there exists $x \in \mathbb{Z}_p^*$ such that $x^2 = a \mod p$ (note that this definition excludes 0 as a quadratic residue).

1. Prove that there are exactly $(p-1)/2$ quadratic residues modulo p.

2. For $x \in \mathbb{Z}_p^*$, the *Legendre symbol* of x mod p, denoted $\left(\frac{x}{p}\right)$, is defined by:

   - $\left(\frac{x}{p}\right) = 1$ if x is a quadratic residue modulo p;
   - $\left(\frac{x}{p}\right) = -1$ otherwise.

   Prove that $\left(\frac{x}{p}\right) = x^{(p-1)/2} \mod p$.

3. Deduce a polynomial time algorithm for determining whether or not a given number x is a quadratic residue modulo p; analyze the number of operations performed by your algorithm.

4. We now consider that p is a prime of the form $p = 4k+3$. Let $a \in \mathbb{Z}_p^*$ be a quadratic residue mod p. Prove that $a^{k+1} \mod p$ is a square root of a modulo p. Deduce a polynomial time algorithm to compute a square root modulo p.

Question 2. Blum integer, polynomial reduction and CSPRNG. (points: 25%)

1. Let $n = p.q$ be a Blum integer: $p = 4k_1 + 3$ and $q = 4k_2 + 3$ are prime integers. It is assumed that p and q are known. Give a polynomial time algorithm for determining whether or not a given number x is a quadratic residue modulo n; analyze the number of operations performed by your algorithm.

2. Consider the five following problems:

   - PrimeQudraticResidue
     - input: p a prime integer and $a \in \mathbb{Z}_p^*$;
     - output: YES iff a is a quadratic residue modulo p.
   - PrimeSquareRoot
     - input: p a prime integer and $a \in \mathbb{Z}_p^*$ a quadratic residue modulo p.
     - output: x such that $x^2 = a \bmod p$.
   - BlumQudraticResidue
     - input: n a Blum integer and $a \in \mathbb{Z}_n^*$;
     - output: YES iff a is a quadratic residue modulo n.
   - BlumSquareRoot
     - input: n a Blum integer and $a \in \mathbb{Z}_n^*$ a quadratic residue modulo n.
     - output: x such that $x^2 = a \bmod n$.
   - BlumFactorization
     - input: n a Blum integer;
     - output: p a prime factor of n.

   From previous questions only, what can be said about the relative complexities of those problems? Argue precisely by exhibiting polynomial time reductions.

3. Complete your answer using some other properties studied during the lectures and/or TD-tutorials.

4. We assume that Alice has a public pseudo-random bit generator `rand()`, initialized with a seed, that passes all polynomial-time statistical tests.
   Alice chooses a secret prime number p and uses it to build a private bit generator R_p.
   To generate a random bit r with R_p, she proceeds as follows. With `rand()`, she generates $L = \log_2 p$ bits a random bit sequence $b_0, b_1, \ldots, b_L$ and computes $x = \sum_{i=0}^{L} b_i.2^i$; then if x is a quadratic residue modulo n, she returns the bit $r = 1$; else the bit $r = 0$.
   Under which conditions the random bit generator R_p can be considered as cryptographically secure?

5. Now, instead of the secret prime p, Alice chooses a public Blum integer n and generates random bits using R_n. Under which conditions the random bit generator R_n can be considered as cryptographically secure?

End Part A