Cryptography and Security — Midterm Exam

Serge Vaudenay

18.11.2021

– duration: 1h45
– no documents allowed, except one 2-sided sheet of handwritten notes
– a pocket calculator is allowed
– communication devices are not allowed
– the exam invigilators will not answer any technical question during the exam
– readability and style of writing will be part of the grade
– answers should not be written with a pencil

1 Diffie-Hellman in an RSA subgroup

The crypto apprentice wants to run the Diffie-Hellman protocol, but instead of running it in a subgroup of \( \mathbb{Z}_p^* \) with a prime \( p \), he decides to run it in a subgroup of \( \mathbb{Z}_n^* \) with an RSA modulus \( n \). He wants \( n \) to remain hard to factor, “for more security”. One goal of the exercise is to see if \( n \) indeed remains hard to factor.

We let \( n = pq \). We let \( g \in \mathbb{Z}_n^* \) and we denote by \( m \) its order in the group. We denote \( p' \) resp. \( q' \) the multiplicative order of \( g \) in \( \mathbb{Z}_p^* \) resp. \( \mathbb{Z}_q^* \). We assume that \( n \) and \( g \) are known by everyone.

Q.1 Prove that both \( p' \) and \( q' \) divide \( m \).
Q.2 In this question, we assume that \( q' = 1 \) and \( m > 1 \). Prove that anyone can factor \( n \) easily.
Q.3 We now assume that \( p' \) and \( q' \) are two different prime numbers. Prove that \( m = p'q' \).
Q.4 We still assume that \( p' \) and \( q' \) are different primes. We also assume that \( m \) is known and easy to factor. Fully specify a Diffie-Hellman protocol.
   Pay special attention to protection against subgroup issues.
Q.5 What is the problem if \( m \) is not known by Alice or Bob?
Q.6 If \( m \) is prime, prove that either \( p' = m \) and \( q' = 1 \), or \( p' = 1 \) and \( q' = m \), or \( p' = q' = m \).
Q.7 Is it a good idea to select \( m \) prime?

2 ElGamal over Exponentials

We consider the following public-key cryptosystem:

– Setup(1^λ): generate a prime \( q \) of size \( \lambda \) and parameters for a cyclic group of order \( q \). Select a generator \( g \) of this group. Set \( pp = (\text{parameters}, q, g) \). Given \( pp \), we assume that group operations are done in polynomial time complexity in \( \lambda \).
– Gen(pp): pick \( x \in \mathbb{Z}_q \) uniformly and \( y = g^x \) in the group. The secret key is \( x \) and the public key is \( y \).
– Enc(pp, y, pt): pick \( r \in \mathbb{Z}_q \) uniformly and output the ciphertext \( (u, v) = (g^r, y^r) \).
– Dec(pp, x, u, v): solve \( y^r = u/v^x \) in \( pt \).

We assume that the encryption domain is the set of small integers: \( pt \in \{0, 1, \ldots, P(\lambda) - 1\} \), where \( P \) denotes a polynomial which will be discussed.
Q.1 Assuming that $2^{\lambda - 1} \geq P(\lambda)$, prove that the cryptosystem is correct.

Q.2 Propose a (non-polynomial) algorithm to do a key recovery attack and give its complexity.
    Note: correct answers with the lowest complexity will get more points.

Q.3 Propose a polynomial-time algorithm to implement $\text{Dec}$. 

Q.4 Propose an appropriate way to select $P$ and $\lambda$.

### 3 Generator of $\mathbb{QR}_n$

We take $n = pq$ with two different primes $p$ and $q$ which are such that $p' = \frac{p - 1}{2}$ and $q' = \frac{q - 1}{2}$ are two odd prime numbers. We let $\mathbb{QR}_n$ be the group of quadratic residues modulo $n$, i.e. all elements which can be written $x^2 \mod n$ for $x \in \mathbb{Z}_n^*$. 

Q.1 Prove that $\mathbb{QR}_n$ has order $\varphi(n)/4$.

Q.2 Prove that $\mathbb{QR}_n$ is cyclic. How many generators exist in $\mathbb{QR}_n$?

Q.3 Propose an efficient algorithm to find a generator of $\mathbb{QR}_n$ which does not need the factor-ization of $n$ but may fail with negligible probability (in terms of $\lambda$, the bitlength of $p$ and $q$, i.e. $2^{\lambda - 1} < p < 2^\lambda$ and $2^{\lambda - 1} < q < 2^\lambda$).