Cryptography and Security — Midterm Exam
Solution

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

The exam grade follows a linear scale in which each question has the same weight.

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 $\mathbf{Z}_p^*$ with a prime p, he decides to run it in a subgroup of $\mathbf{Z}_n^*$ 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 $g \in \mathbf{Z}_n^*$ and we denote by m its order in the group. We denote p' resp. q' the multiplicative order of g in $\mathbf{Z}_p^*$ resp. $\mathbf{Z}_q^*$. We assume that n and g are known by everyone.

Q.1 Prove that both p' and q' divide m.

> *p is a factor of n. We have $g^m \bmod n = 1$ so $g^m \bmod p = 1$ as well. Hence, m is a multiple of the order of g in $\mathbf{Z}_p^*$, which is p'. Therefore, p' divides m.*
> *The same argument holds with q.*

Q.2 In this question, we assume that $q' = 1$ and $m > 1$. Prove that anyone can factor n easily.

> *Since $q' = 1$, we have $g \bmod q = 1$. Hence, q is a factor of $\mathsf{gcd}(g-1, n)$ which is a factor of n. If $\mathsf{gcd}(g-1, n) = n$, this implies that $g \bmod n = 1$, which is not possible because $m > 1$. Hence, $\mathsf{gcd}(g-1, n) = q$. We can compute the factor q of n by using the Euclid algorithm. We deduce $p = n/q$ which gives the full factorization of n.*

Q.3 We now assume that p' and q' are two different prime numbers. Prove that $m = p'q'$.

> *We first observe that $g^{p'q'} \bmod p = g^{p'q'} \bmod q = 1$ so $g^{p'q'} \bmod n = 1$ due to the Chinese Remainder Theorem. Thus, m divides $p'q'$.*
> *We have $g^m \bmod n = 1$ so $g^m \bmod p = 1$ so p' divides m. Similarly, q' divides m. Hence, $\mathsf{lcm}(p', q')$ divides m. (Recall that for any triplet of integers a, b, c such that $a|c$ and $b|c$, we have $\mathsf{lcm}(a, b)|c$.) Since p' and q' are different primes, $\mathsf{lcm}(p', q') = p'q'$ which divides m.*
> *Therefore, $m = p'q'$.*

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.

Since $m = p'q'$ (due to the previous question), m is known, and m is easy to factor, p' and q' are also known.
Alice picks $a \in \mathbf{Z}_m^*$ and sends $A = g^a \bmod n$ to Bob. Bob picks $b \in \mathbf{Z}_m^*$ and sends $B = g^b \bmod n$ to Alice. By picking a and b in $\mathbf{Z}_m^*$, this makes sure that A and B both have multiplicative order m, so they do not belong to a subgroup.
Alice verifies $1 < B < n$, $B^{p'} \bmod n \neq 1$, $B^{q'} \bmod n \neq 1$, and $B^m \bmod n = 1$. This ensures that B has multiplicative order m.
Similarly, Bob verifies $1 < A < n$, $A^{p'} \bmod n \neq 1$, $A^{q'} \bmod n \neq 1$, and $A^m \bmod n = 1$.
They both compute $C = B^a \bmod n = A^b \bmod n = g^{ab} \bmod n$. Finally, they apply a KDF on C to obtain the final output K.

Q.5 What is the problem if m is not known by Alice or Bob?

They have a problem to select their ephemeral secret at random. Ideally, they should pick it in $\mathbf{Z}_m^*$.

Q.6 If m is prime, prove that either $p' = m$ and $q' = 1$, or $p' = 1$ and $q' = m$, or $p' = q' = m$.

We have seen that both p' and q' divide m. Since m is prime, $p' = 1$ or $p' = m$. Similarly, $q' = 1$ or $q' = m$. If $p' = q' = 1$, we have $g^1 \bmod p = 1$ and $g^1 \bmod q = 1$ so $g \bmod n = 1$ thus $m = 1$ which contradicts that m is prime. Hence, we can conclude.

Q.7 Is it a good idea to select m prime?

We have seen it is not a good idea to have $p' = 1$ or $q' = 1$ (otherwise, we can factor n and there is no point in using an RSA group). What is left is the $p' = q' = m$ case.
With $p' = q' = m$, we can write $p = \alpha m + 1$, $q = \beta m + 1$, so $n = \alpha\beta m^2 + (\alpha+\beta)m + 1$. This special form with m known may ease factorization.
For instance, when $\alpha + \beta < m$, we can recover

$$\alpha + \beta = \frac{n-1}{m} \bmod m$$

We can also recover

$$\alpha\beta = \left\lfloor \frac{n-1}{m^2} \right\rfloor$$

Then, α and β are the roots of the equation

$$x^2 - (\alpha + \beta)x + \alpha\beta = 0$$

from which we deduce p and q.
When $\alpha + \beta \geq m$, it is more complicated.

2 ElGamal over Exponentials

We consider the following public-key cryptosystem:

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

We assume that the encryption domain is the set of small integers: $\mathsf{pt} \in \{0, 1, \ldots, P(\lambda) - 1\}$, 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.

> *If we encrypt correctly with $u = g^r$ and $v = g^{\mathsf{pt}} y^r$, then $v/u^x = g^{\mathsf{pt}} y^r / g^{rx} = g^{\mathsf{pt}}$. So, pt is a solution to the equation to solve. The value of the solution is unique modulo q. Since $q > 2^{\lambda-1} \geq P(\lambda)$, the solution in the encryption domain is unique. Hence, we have correctness.*

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.

> *The generic baby-step giant-step algorithm computes x from y within a complexity of $\mathcal{O}(\sqrt{q})$ group operations. So, the complexity is $\mathcal{O}(2^{\frac{\lambda}{2}})$ group operations.*

Q.3 Propose a polynomial-time algorithm to implement Dec.

> *We can use the baby-step giant-step algorithm which works with complexity $\mathcal{O}(\sqrt{P(lambda)})$ group operations.*

Q.4 Propose an appropriate way to select P and λ.

> *We need $\sqrt{P(\lambda)}$ to be small. For instance, $\sqrt{P(\lambda)} < 2^{32}$. We need $2^{\frac{\lambda}{2}}$ to be huge. For instance, $2^{\frac{\lambda}{2}} = 2^{128}$. So, $\lambda = 256$ and $P(\lambda) = 2^{64}$ could be good.*
> *As a rule of thumb, we could suggest $P(\lambda) = \lambda^8$.*

3 Generator of 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 QR_n be the group of quadratic residues modulo n, i.e. all elements which can be written $x^2 \bmod n$ for $x \in \mathbf{Z}_n^*$.

Q.1 Prove that QR_n has order $\varphi(n)/4$.

Thanks to the Chinese Remainder Theorem, $\mathbf{Z}_n^*$ is isomorphic to $\mathbf{Z}_p^* \times \mathbf{Z}_q^*$, which is isomorphic to $\mathbf{Z}_{p-1} \times \mathbf{Z}_{q-1}$.
$\mathbf{Z}_{p-1}$ is isomorphic to $\mathbf{Z}_2 \times \mathbf{Z}_{p'}$ because p' is odd. Similarly, $\mathbf{Z}_{q-1}$ is isomorphic to $\mathbf{Z}_2 \times \mathbf{Z}_{q'}$. Hence, $\mathbf{Z}_n^*$ is isomorphic to $\mathbf{Z}_2 \times \mathbf{Z}_2 \times \mathbf{Z}_{p'} \times \mathbf{Z}_{q'}$.
Using this isomorphism, the squares of $\mathbf{Z}_n^*$ is isomorphic to the doubles of $\mathbf{Z}_2 \times \mathbf{Z}_2 \times \mathbf{Z}_{p'} \times \mathbf{Z}_{q'}$. Since 2 is invertible modulo p' and q', we have

$$2.(\mathbf{Z}_2 \times \mathbf{Z}_2 \times \mathbf{Z}_{p'} \times \mathbf{Z}_{q'}) = \{(0, 0, a, b); (a, b) \in \mathbf{Z}_{p'} \times \mathbf{Z}_{q'}\}$$

which has order $p'q' = \varphi(n)/4$.

Q.2 Prove that QR_n is cyclic. How many generators exist in QR_n?

By the previous isomorphism, QR_n and $\mathbf{Z}_{p'} \times \mathbf{Z}_{q'}$ are isomorphic. It is isomorphic to $\mathbf{Z}_{p'q'}$ which is cyclic. So, QR_n is cyclic.
The number of generators is the same as in $\mathbf{Z}_{p'q'}$ which is $\varphi(p'q')$.

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

We show that if we pick a random $r \in \mathbf{Z}_n^*$ and we set $g = x^2 \bmod n$, then g is a generator almost surely.
Indeed, each element of QR_n has exactly 4 square roots in $\mathbf{Z}_n^*$ so the squaring operation is a balanced function onto QR_n. Hence, g is uniform in QR_n.
The probability it is not a generator is

$$1 - \frac{\varphi(p'q')}{p'q'} = \frac{1}{p'} + \frac{1}{q'} - \frac{1}{p'q'}$$

We have $p' > 2^{\lambda-2}$ and $q' > 2^{\lambda-2}$, so this is upper bounded by $2^{3-\lambda}$, which is negligible in terms of λ.