Cryptography and Security — Midterm Exam
Solution

Ioana Boureanu and Serge Vaudenay

30.11.2012

- duration: 1h45
- no documents is allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will *not* answer any technical question during the exam
- (if extra space is needed:) the answers to each exercise must be provided on separate sheets
- readability and style of writing will be part of the grade
- do not forget to put your name on every sheet!

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

1 Message Encoding in a Subgroup of $\mathbf{Z}_p^*$ of Prime Order

In what follows, p is an odd prime number which can be written $p = 2q + 1$ with q being another odd prime number.

Q.1 What is the order of $\mathbf{Z}_p^*$?
List all factors of this number.
What are the orders of 1 and -1 in $\mathbf{Z}_p^*$?

Since p is a prime number, $\mathbf{Z}_p^*$ is of order $p-1=2q$. The factors of $p-1$ are thus 1, 2, q, and $2q$.
1 has order 1 since this is the smallest power i such that $x^i = 1$ in $\mathbf{Z}_p^*$.
-1 has order 2 for the same reason.

Q.2 If $x \in \mathsf{QR}_p$ is such that $x \neq 1$, show that x generates QR_p.
Hint: What is the order of QR_p?

The order of QR_p is known to be $\frac{p-1}{2} = q$. Here is a proof of this:
Quadratic residues are roots of $x^{\frac{p-1}{2}} = 1$. Since we are in a field, we have at most $\frac{p-1}{2}$ roots.
Let i be a non-quadratic residue. For any non-quadratic residue y, iy must be a quadratic residue. Since $y \mapsto iy$ is a 1-to-1 mapping, we have at most $\frac{p-1}{2}$ non-quadratic residues.
Now, Z_p^* has $p-1$ terms which are either quadratic residues or non-quadratic residues. So, we have exactly $\frac{p-1}{2}$ quadratic residues and $\frac{p-1}{2}$ non-quadratic residues. So, QR_p is a group of order q.

The order of x must divide the order of QR_p which is q. Since q is prime, the order must be 1 or q. If $x \neq 1$, the order cannot be 1. So, x has order q. This means that x generates QR_p.

Q.3 Let QR_p be the set of all quadratic residues of $\mathbf{Z}_p^*$. Show that for all $x \in \mathbf{Z}_p^*$, we have $x \in \mathsf{QR}_p$ if and only if $x^{\frac{p-1}{2}} = 1$ in $\mathbf{Z}_p$.

If $x \in \mathsf{QR}_p$, we can write $x = y^2$. So, $x^{\frac{p-1}{2}} = y^{p-1} = 1$ due to the Little Fermat Theorem.
Conversely, if $x^{\frac{p-1}{2}} = 1$, since we know that $\mathbf{Z}_p^*$ is cyclic, we can write $x = g^i$ for some generator g of $\mathbf{Z}_p^*$ and have $g^{i\frac{p-1}{2}} = 1$. Since g is a generator, this implies that $p-1$ divides $i\frac{p-1}{2}$, so that i is even. Hence, $i = 2j$ for some integer j and we have $x = g^{2j} = y^2$ with $y = g^j$. That is, x is a quadratic residue.

Q.4 Given $x \in \{1, \ldots, q\}$, show that the cardinality of $\{x, -x\} \cap \mathsf{QR}_p$ is 1.
Hint: is -1 in QR_p?

We have $(-1)^{\frac{p-1}{2}} = (-1)^q = -1$ since q is odd. So, -1 is not a quadratic residue in $\mathbf{Z}_p^*$.
Since -1 is not a quadratic residue, x and $-x$ cannot be quadratic residues at the same time. If x is not a quadratic residue, it means that $x^{\frac{p-1}{2}} = -1$. So, $(-x)^{\frac{p-1}{2}} = 1$. Therefore, $-x$ is a quadratic residue. So, either x or $-x$ is a quadratic residue, but not both.

Q.5 Given $x \in \{1, \ldots, q\}$, let $\mathsf{map}(x)$ be the only element between x and $-x$ which is a quadratic residue. Show that map is an one-to-one mapping between $\{1, \ldots, q\}$ and QR_p.

We have already shown that it is a mapping. Given $y \in \mathsf{QR}_p$, $\mathsf{map}(x) = y$ implies that $x = y$ or $x = -y$. Since we cannot have y and $-y$ belonging to $\{1, \ldots, q\}$ at the same time, map is 1-to-1.

2 Arithmetic Modulo 101 and 99 999

Let $m = 101$, $n = 99\,999$, $a = 4\,499\,955$ and $b = 5\,599\,945$.

Q.1 For $N = 10^k \pm 1$, $k \geq 1$, give a method to compute by hand the modulo N reduction of a big decimal number.

We group the digits by packets of k from the left to the right.
In the case of $N = 10^k - 1$, we have $10^k \equiv 1 \pmod{N}$ so we can just add the obtained numbers and iterate on the result until it is less than 10^k.
In the case of $N = 10^k + 1$, we have $10^k \equiv -1 \pmod{N}$ so we can alternate the numbers with $+$ and $-$ and iterate on the result. If the final number is negative, we can just add N. If the final number is N, we can replace it by 0.

Q.2 Compute $a \bmod m$, $a \bmod n$, $b \bmod m$, and $b \bmod n$.

When applying to the modulo m cases, we have

$$a \equiv 4\,49\,99\,55 \equiv 55 - 99 + 49 - 4 \equiv 1 \pmod{m}$$
$$b \equiv 5\,59\,99\,45 \equiv 45 - 99 + 59 - 5 \equiv 0 \pmod{m}$$

When applying to the modulo n cases, we have

$$a \equiv 44\,99955 \equiv 44 + 99955 \equiv 99999 \equiv 0 \pmod{n}$$
$$b \equiv 55\,99945 \equiv 55 + 99945 \equiv 1\,000001 + 0 \equiv 1 \pmod{n}$$

So, a is 1 modulo m and b is 0 modulo m. So, a is 0 modulo n and b is 1 modulo n.

Q.3 Deduce the *lowest* positive multiple of n which is equal to 2 modulo m.

By applying the Chinese Remainder Theorem, we obtain $(2a + 0b) \bmod (mn) = 2 \times 4\,499\,955 = 8\,999\,910$.

3 Every Day I'm Shuffling

The following exercise is inspired from An Enciphering Scheme Based on a Card Shuffle *by Tung Hoang, Morris, and Rogaway, published in the proceedings of Crypto'12 pp. 1–13, LNCS vol. 7417, Springer 2012; and by* The End of Encryption based on Card Shuffling *by Vaudenay, presented at the Rump Session of Crypto'12.*

Let n and r be integers. We consider the vector space $\mathsf{GF}(2)^n$ over $\mathsf{GF}(2)$. A vector $x = (x_1, \ldots, x_n)$ has n binary coordinates $x_1, \ldots, x_n$. We denote by $\oplus$ the addition of vectors. We denote by $x \cdot y$ the inner product between two vectors x and y. I.e., $x \cdot y = x_1y_1 + \cdots + x_ny_n \bmod 2$. Finally, given two vectors x and y, we define the function $\max(x, y)$ giving the one vector among x and y which represents the binary expansion of the largest integer. (Assume that bits written from left to right, i.e. x_n is the least significant bit.)

Given $2r$ vectors $K_1, \ldots, K_r, L_1, \ldots, L_r$, we denote $KL = (K_1, \ldots, K_r, L_1, \ldots, L_r)$ and we define the encryption $E_{KL}(X)$ of a vector X with key KL by the following algorithm:

proc $E_{KL}(X)$
1: **for** $i = 1$ to r **do**
2: $X' \leftarrow K_i \oplus X$
3: $\hat{X} \leftarrow \max(X, X')$
4: if $L_i \cdot \hat{X} = 1$ then $X \leftarrow X'$
5: **end for**
6: return X

Q.1 Let j be the smallest index such that the jth component of K_i is 1. In iteration i, we consider the values of X and $\hat{X}$ in step 3. Show that $\hat{X} = X \oplus (1 - X_j)K_i$.

To compute $\max(X, X \oplus K_i)$*, we have to compare the bits in* X *and* $X \oplus K_i$ *starting from the most significant ones. The first index where they can be compared is at position* j *since they are always equal before. The maximum is* X *if* $X_j = 1$ *and* $X \oplus K_i$ *otherwise. So, in all cases, it can be written* $X \oplus (1 - X_j)K_i$*.*

Q.2 In iteration i, we let X_{new} be the value of X after step 4 and still consider the same X and $\hat{X}$. Show that $X_{\mathsf{new}} = X \oplus (L_i \cdot \hat{X})K_i$.

X_{new} *is equal to* X *if* $L_i \cdot \hat{X} = 0$ *and to* $X \oplus K_i$ *otherwise. So, in all cases, it can be written* $X \oplus (L_i \cdot \hat{X})K_i$*.*

Q.3 Deduce that for whatever KL, x, and y, we have $E_{KL}(x \oplus y) \oplus E_{KL}(0) = E_{KL}(x) \oplus E_{KL}(y)$.

Due to the previous questions, we know that each iteration is just replacing X *by*

$$\begin{aligned} X_{\mathsf{new}} &= X \oplus (L_i \cdot (X \oplus (1 - X_j)K_i))K_i \\ &= X \oplus (L_i \cdot X)K_i \oplus X_j(L_i \cdot K_i)K_i \oplus (L_i \cdot K_i)K_i \end{aligned}$$

which is an affine function in X*. Since the composition of affine functions is affine, we obtain that* E_{KL} *is an affine function as well. So, it satisfies* $E_{KL}(x \oplus y) \oplus E_{KL}(0) = E_{KL}(x) \oplus E_{KL}(y)$*.*

Q.4 Propose a way to break this symmetric encryption scheme.

Since E_{KL} is an affine function, it can be written $E_{KL}(X) = M \times X \oplus c$ for some matrix M and some constant vector c depending on KL. So, with a few known plaintexts-ciphertext pairs (X_i, Y_i), we can recover M and c by solving a linear system of equations $Y_i = M \times X_i \oplus c$ in M and c. After solving, we can decrypt any message Y by $M^{-1} \times (Y \oplus c)$.