Cryptography and Security — Final Exam
Solution

Ioana Boureanu and Serge Vaudenay

15.1.2013

- duration: 3h
- no documents are 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 write your name on every sheet!

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

1 Modular Arithmetic

Let p and q be two different odd prime numbers and $n = pq$.

Q.1 Show that p is invertible modulo q and that q is invertible modulo p.

p is invertible in $\mathbf{Z}_q$ if and only if p and q are coprime. Since p is prime, p is invertible in $\mathbf{Z}_q$ if and only if q is not divisible by p. Since $p \geq 2$ and since p is prime and different from q, q cannot be divisible by p. So, p is invertible modulo q. Similarly, q is invertible modulo p.

In what follows, $\alpha = q \times q'$ where $q' \in \mathbf{Z}$ is the inverse of q modulo p, and $\beta = p \times p'$ where $p' \in \mathbf{Z}$ is the inverse of p modulo q. We define $f(x, y) = \alpha x + \beta y$, where $x, y \in \mathbf{Z}$.

Q.2 For $x \in \{0, \ldots, p-1\}$ and $y \in \mathbf{Z}$, what is $f(x, y) \bmod p$?

Since β is a multiple of p, we have $f(x, y) \equiv \alpha x \pmod{p}$. Now, $\alpha \bmod p = 1$. Since $0 \leq x < p$, we obtain that $f(x, y) \bmod p = x$.

Q.3 Which concept of the course corresponds to the function f?

The function f is used in the Chinese Remainder Theorem. Indeed, $f(x, y) \equiv x \pmod{p}$ and $f(x, y) \equiv y \pmod{q}$.

Q.4 Show that $f(1, 1) = 1 + n$.

We have $f(1, 1) \bmod p = 1$ and $f(1, 1) \bmod q = 1$. Since p and q are coprime, we deduce $f(1, 1) = 1 + kn$ for some integer k.
We have $f(1, 1) = \alpha + \beta$ so $0 < f(1, 1) \leq q(p-1) + p(q-1) = 2n - p - q$. Clearly, $f(1, 1) = \alpha + \beta > 1$. So, $1 \leq k < 2$. We deduce $k = 1$ so $f(1, 1) = 1 + n$.

Q.5 Give the largest common factor of all numbers of the form $f(x,x) - x$ for $x \in \mathbf{Z}$.

Since $f(x,x) \bmod p = x$ due to the previous question, we obtain that $f(x,x) - x$ is divisible by p. Similarly, it must be divisible by q. Since p and q are coprime, $f(x,x) - x$ is divisible by n, for all x.
Let m be the largest common factor of all $f(x,x) - x$. Since n divides all $f(x,x) - x$, n divides m as well, so $m \geq n$.
Since $f(1,1) - 1 = n$, we have $m \leq n$. So, $m = n$.

Q.6 Let $x \in \mathbf{Z}_n$. Using f, list all the square roots of $x^2 \bmod n$ in $\mathbf{Z}_n$.

We only have two square roots modulo p which are x and $-x$. The same holds modulo q. Then, the square roots modulo n must be of form $f(\pm x, \pm x) \bmod n$. We can see that we obtain the four square roots which are x, $-x$, $f(x,-x)$, and $-f(x,-x)$ modulo n.

Q.7 Assuming that $p < q$, that $x \in \{0, \ldots, p-1\}$, $y \in \{0, \ldots, q-1\}$, that $x \neq y$, let $z = f(x,y)$. Give an algorithm to compute p and q when given z, x, and n.

We know that $z - x$ is divisible by p. We further have $z - x \equiv y - x \pmod{q}$. Due to the assumptions, this is not divisible by q. So, $\gcd(z - x, n) = p$. We can compute p by this formula then compute $q = n/p$.
This algorithm was seen in the course when we have shown that we can factor n based on a square root algorithm. In this case, the square roots of $x^2 \bmod n$ can be written $f(\pm x, \pm x) \bmod n$, and we hoped to get $f(x,-x) \bmod n$. Given $f(x,-x)$ and x, we could factor n.

2 A MAC Based on DES

We construct a (bad) MAC as follows: given a message m and a key K, we first compute $h = \mathsf{trunc}(\mathsf{SHA1}(m))$ where trunc maps onto the keyspace of DES (assume that the preimages by trunc have the same size). Then, we compute $c = \mathsf{DES}_h(K)$ which is the authentication code.

Q.1 How many bits of entropy are used from m to compute c?

Since a DES key has 56 bits, only 56 useful bits from m are used. (Although h has 64 bits, 8 of these bits are not used by DES.)

Q.2 How many random messages do we need in order to see the same authentication code twice with a good probability? (Explain.)

Due to the birthday paradox, we roughly need 2^{28} random messages to have a collision on the DES key which is computed from m.

Q.3 Describe a chosen-message forgery attack against the MAC which uses only one chosen message.

We first look for the collision in an offline way, using about 2^{28} random messages. When this is done, we obtain m_1 and m_2 producing the same DES key. Therefore, we must have $c = \mathsf{MAC}_K(m_1) = \mathsf{MAC}_K(m_2)$. So, we just use m_1 as a chosen message to get c, and we produce the forgery (m_2, c).

The above was the expected answer to the exercise. However, there was a typo in the definition of the MAC which made possible to propose a better solution: With the given definition, there is a much better attack as follows: given m and $c = \mathsf{MAC}_K(m)$, compute h from m then $K = \mathsf{DES}_h^{-1}(c)$. This recovers the key, based on which we can make forgeries.
The correct definition of the MAC would have been $c = K \oplus \mathsf{DES}_h(K)$.

3 Secure Communication

We want to construct a secure communication channel using cryptography.

Q.1 List the three *main* security properties that we need *at the packet level* to achieve secure communication. For each property, explain what it means and say which cryptographic technique can be used to obtain it.

Confidentiality. *This ensures that only the legitimate receiver can receive the packet. This is protected by symmetric encryption.*
Integrity. *This ensures that the received packet must be equal to the sent one. This is protected by a MAC, together with the next property.*
Authentication. *This ensures that only the legitimate sender can send valid messages. This is protected by a MAC.*

Q.2 Assuming that packet communication is secure, list two extra properties (other than key establishment) that we need in order to secure *an entire session*, and how to ensure these properties.

Sequentiality. *This ensures that the sequence of packets which is seen at both ends of the channel are prefix of each other. This is protected by numbering the packets and authenticating the packet number together with the packet itself.*
Fairness of termination. *This ensures that both ends see the same final message. This is hard to protect. It can be achieved by using the KiT protocol, but it is expensive.*

Q.3 How to secure a key establishment to initialize the secure channel? Give two solutions.

We must securely set up a symmetric key. This can be done using an extra secure communication channel.
For instance, with a channel protecting authentication and integrity, we can use the Diffie-Hellman protocol.
Otherwise, we can use a third party. E.g., a secure channel to a human user to help authenticating key exchange with SAS-based cryptography. Or, secure channels to a certificate authority to set up a PKI.

4 On Entropies

We define $\mathsf{nextprime}(x)$ as the smallest prime number p such that $p \geq x$. We want to sample a prime number greater than 40 as follows: given a random number R with uniform distribution between 1 and 16, we compute $X = \mathsf{nextprime}(40 + R)$. For X secret, we consider the problem of finding X.

Q.1 Give the distribution of all possible values for X.

We compute the table of X in terms of R:

R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
X	41	43	43	47	47	47	47	53	53	53	53	53	53	59	59	59

So, we obtain the following distribution:

x	41	43	47	53	59
$\Pr[X]$	$\frac{1}{16}$	$\frac{2}{16}$	$\frac{4}{16}$	$\frac{6}{16}$	$\frac{3}{16}$

Q.2 Compute $H(X)$, the Shannon entropy of X and the value $c = \frac{1}{2}\left(2^{H(X)} + 1\right)$.
Reminder: $H(X) = -\sum_x \Pr[X = x] \log_2 \Pr[X = x]$

By applying the formula, we obtain $H(X) = 2.108459$ and $c = 2.656152$. So, X has about 2.1 bits of entropy. An exhaustive search on a uniformly distributed string of 2.1 bits would require and average complexity of 2.7.

Q.3 Compute $G(X)$, the guesswork entropy of X, and compare it with c. What do we deduce?
Reminder: $G(X)$ is the lowest expected complexity in the following game. A challenger samples X, keeps it secret, and answers questions as follows. The adversary, trying to guess X, can ask as many questions as he wants of the form "is the secret X equal to x?" for any value x. The complexity is the number of questions until one answer is "yes".

Clearly, the best strategy is to ask the questions in the following order until one answer is "yes":

is the secret X equal to 53?
is the secret X equal to 47?
is the secret X equal to 59?
is the secret X equal to 43?
is the secret X equal to 41?

The average complexity is

$$G(X) = \frac{6}{16} \times 1 + \frac{4}{16} \times 2 + \frac{3}{16} \times 3 + \frac{2}{16} \times 4 + \frac{1}{16} \times 5 = 2.250000$$

We notice that $G(X) < c$. Actually, the best way to do an exhaustive search on X is more efficient than doing an exhaustive search on $H(X)$ bits, although we might think it is the same.

Q.4 By sampling two independent prime numbers X and Y following the same distribution, what is the probability that $X = Y$?

This is

$$\Pr[X = Y] = \sum_x \Pr[X = Y = x] = \sum_x \Pr[X = x]^2 = 0.257813$$

5 Pedersen Commitment

The following exercise is inspired from Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing *by Pedersen, published in the proceedings of Crypto'91 pp. 129–140, LNCS vol. 576, Springer 1992.*

Let p and q be two prime numbers such that q divides $p-1$. Let g be an element of $\mathbf{Z}_p^*$ of order q. Let h be in the subgroup of $\mathbf{Z}_p^*$ generated by g but different from the neutral element. Given two numbers x and r, we define a commitment scheme by $\mathsf{commit}(x;r) = g^x h^r \bmod p$.

The protocol works as follows. We assume that the sender wants to commit to a message x to a receiver. In the commitment phase, the sender selects r at random, compute $y = \mathsf{commit}(x;r) = g^x h^r \bmod p$ and sends y to the receiver. In the opening phase, the sender sends some values and the receiver does some computation. (Formalizing further this phase is subject to a question.)

Q.1 Fully formalize what the sender sends to the receiver in *the opening phase* and which computation *the receiver is doing.*

Opening the commitment means that the sender provides the value x and his coins r. The algorithm consists of checking that $y = \mathsf{commit}(x;r)$ and producing x as the protocol outcome.

Q.2 Let X and R be two independent random variables with values in $\mathbf{Z}_q$ such that R is uniformly distributed in $\mathbf{Z}_q$. Let $Y = \mathsf{commit}(X;R)$. Show that Y is uniformly distributed in the subgroup of $\mathbf{Z}_p^*$ generated by g.

Hint: use h in the subgroup of $\mathbf{Z}_p^*$ generated by g.

We write $h = g^a \bmod p$ for some $a \in \mathbf{Z}_q$. Since h is not the neutral element and q is prime, we deduce that $a \in \mathbf{Z}_q^*$. Since X and R are independent and since R is uniformly distributed, we deduce that $X + aR \bmod q$ is uniformly distributed in $\mathbf{Z}_q$. Now, $x \mapsto g^x \bmod p$ is a one-to-one mapping from $\mathbf{Z}_q$ to the subgroup generated by g, so Y is uniformly distributed.

Q.3 With the sames settings, show that X and Y are independent.

Since X and R are independent and that R is uniformly distributed, we obtain that $X + aR \bmod q$ and X are independent. Due to the one-to-one mapping, we obtain that X and Y are independent.
The complete proof of uniformity and independence (not necessary in this exercise) goes like in the course: given $x \in \mathbf{Z}_q$, $y \in \langle g \rangle$, $y = g^b$,

$$\begin{aligned}\Pr[X = x, Y = y] &= \Pr[X = x, X + aR \equiv b] \\ &= \Pr[X = x, R \equiv (b - x)/a] \\ &= \Pr[X = x] \Pr[R = (b - x)/a \bmod q] \\ &= \frac{\Pr[X = x]}{q}\end{aligned}$$

So,

$$\Pr[Y = y] = \sum_x \Pr[X = x, Y = y] = \sum_x \frac{\Pr[X = x]}{q} = \frac{1}{q}$$

and Y is uniform. Finally, $\Pr[X = x, Y = y] = \frac{\Pr[X=x]}{q} = \Pr[X = x] \Pr[Y = y]$ so X and Y are independent.

Q.4 Given p, q, g, h, show that computing $x, r, x', r' \in \mathbf{Z}_q$ such that $\mathsf{commit}(x; r) = \mathsf{commit}(x'; r')$ and $x \neq x'$ is equivalent to computing $a \in \mathbf{Z}_q$ such that $h = g^a \bmod p$.

If we know a such that $h = g^a \bmod p$, for any x, r we have

$$\mathsf{commit}(x + a; r - 1) \equiv g^{x+a} h^{r-1} \equiv g^x h^r \equiv \mathsf{commit}(x; r) \pmod p$$

So, we can compute $x, r, x', r' \in \mathbf{Z}_q$ such that $\mathsf{commit}(x; r) = \mathsf{commit}(x'; r')$ and $x \neq x'$.
Conversely, if we know $x, r, x', r' \in \mathbf{Z}_q$ such that $\mathsf{commit}(x; r) = \mathsf{commit}(x'; r')$ and $x \neq x'$, we have $x + ar \equiv x' + ar' \pmod q$ where a is such that $h = g^a \bmod p$. If we had $r \equiv r' \pmod q$, we would obtain $x \equiv x' \pmod q$ which is not possible due to the assumptions. So, $r - r'$ is invertible modulo q and we deduce

$$a = \frac{x' - x}{r - r'} \bmod q$$

So, we can compute a from x, r, x', r', q.

Q.5 Finding $a \in \mathbf{Z}_q$ such that $h = g^a \bmod p$ is called the discrete logarithm problem.
Assuming that solving the discrete logarithm problem is hard, show that commit defines a *hiding* and *binding* commitment scheme.

Due to Question Q.3, $\mathsf{commit}(X; R)$ is independent from X. So, $\mathsf{commit}(X; R)$ perfectly hides X.
Being able to open a commitment on two different values x and x' means providing x, r, x', r' such that $\mathsf{commit}(x; r) = \mathsf{commit}(x'; r')$. Due to the last question, this is equivalent to being able to compute a such that $h = g^a \bmod p$. Assuming this is hard, we deduce that the commitment is computationally binding.