Cryptography and Security — Final Exam

Serge Vaudenay

28.1.2019

- duration: 3h
- 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

1 The Mersenne Cryptosystem

In what follows, p denotes a prime number of form $p = 2^n - 1$. It is called a *Mersenne prime*. Elements in $\mathbf{Z}_p$ are represented by numbers between 0 and $p-1$. Given $x \in \mathbf{Z}_p$, $W(x)$ denotes the number of 1's when writing the element x in binary.

Q.1 For all $x \in \mathbf{Z}_p^*$, prove that $W(-x \bmod p) = n - W(x)$.

Q.2 For all $x, y \in \mathbf{Z}_p$, prove that $W(x + y \bmod p) \leq W(x) + W(y)$.
HINT: first consider $y = 1$, then $W(y) = 1$, then proceed by induction.

Q.3 For all $x, y \in \mathbf{Z}_p$, prove that $W(x \times y \bmod p) \leq W(x) \times W(y)$.
HINT: use binary and show $W(x2^j \bmod p) = W(x)$.

Q.4 In what follows, h denotes a positive integer such that $4h^2 < n$.

After the parameters n, p, and h are set up, we define the following algorithms:

$\mathsf{Gen}(n, p, h)$:

1: pick $F, G \in \mathbf{Z}_p$ random such that $W(F) = W(G) = h$
2: set $\mathsf{pk} = \frac{F}{G} \bmod p$ and $\mathsf{sk} = G$
3: output pk and sk

$\mathsf{Enc}(\mathsf{pk}, b)$:

4: pick $A, B \in \mathbf{Z}_p$ random such that $W(A) = W(B) = h$
5: set $\mathsf{ct} = \big((-1)^b \times (A \times \mathsf{pk} + B)\big) \bmod p$
6: output ct

where b is a plaintext from the space $\{0, 1\}$ (i.e. we encrypt only one bit).

Design a decryption algorithm and prove it is correct.

Q.5 As a toy example, take $n = 17$, $p = 131\,071$, $h = 2$. Generate a key pair using $F = 2^{14} + 2^2$ and $G = 2^{10} + 2^6$. Then, encrypt $b = 1$ using $A = 2^{11} + 2^5$ and $B = 2^9 + 2^2$. Detail the computations and give, pk, sk, ct.
HINT1: for people who have a 4-operation calculator: $a \times 2^n + b \equiv a + b \pmod{2^n - 1}$.
HINT2: by thinking of how multiplication by 2 works modulo p, find a trick to perform the division by 2.
HINT3: $\frac{1}{17} \bmod p = 123\,361$.

2 Collision Attack on CBC Mode

We consider TLS using a block cipher with n-bit message blocks in CBC mode. The goal of this exercise is to develop message recovery attacks, or at least to recover a sensitive part of a partially-known plaintext.

Q.1 Given 2^d independent and uniformly distributed random variables $X_1, \ldots, X_{2^d}$ with values in $\{0,1\}^n$, what is the expected number of pairs (i,j) with $i < j$ such that $X_i = X_j$?

Q.2 Given 2^s independent and uniformly distributed random variables $X_1, \ldots, X_{2^s}$ and 2^t independent and uniformly distributed random variables $Y_1, \ldots, Y_{2^t}$, all with values in $\{0,1\}^n$, what is the expected number of pairs (i,j) such that $X_i = Y_j$?

Q.3 Consider a list of plaintexts of 2^d blocks in total. We assume that all blocks can be split into three categories: blocks which are already known by the adversary (we denote by α the fraction of blocks in this category), blocks which are privacy-sensitive thus an interested target for the adversary (we denote by β the fraction of blocks in this category), and other blocks which are unknown but uninteresting to recover (within a fraction $1 - \alpha - \beta$). All ciphertext blocks are known by the adversary.

Assuming that the inputs of the block cipher are independent and uniform, design an attack which recovers some privacy-sensitive blocks. How large must 2^d be in order for the expected number of recovered sensitive blocks to be 1? Compute the data complexity 2^d in terms of n, α, and β.

HINT: encryption uses the CBC mode.

Q.4 Assuming that the encryption key changes every 2^r blocks, adapt the previous attack and estimate its data complexity. Application: how much data do we need for $n = 64$, $\alpha = \beta = \frac{1}{2}$, $r = \frac{n}{2}$?

Q.5 We now assume that a plaintext of 2^u blocks is encrypted many times (with a random IV). We assume that all blocks but k sensitive ones are known by the adversary and that $k \ll 2^u$. However, the purpose is now to recover *all* sensitive blocks. Estimate the data complexity (in blocks) in terms of n, u, and k.

3 PKC vs KEM vs KA

In this exercise, we compare *Public-Key Cryptosystems* (PKC), *Key Encapsulation Mechanisms* (KEM), and non-interactive *Key Agreement* schemes (KA). We formalize the interface for each of the three primitives:

PKC

- $\mathsf{Setup} \xrightarrow{\$} \mathsf{pp}$
- $\mathsf{Gen}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{pt}) \xrightarrow{\$} \mathsf{ct}$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to \mathsf{pt}/\bot$

KEM

- $\mathsf{Setup} \xrightarrow{\$} \mathsf{pp}$
- $\mathsf{Gen}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$
- $\mathsf{Enc}(\mathsf{pk}) \xrightarrow{\$} (K, \mathsf{ct})$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to K/\bot$

KA

- $\mathsf{Setup} \xrightarrow{\$} \mathsf{pp}$
- $\mathsf{Gen}_A(\mathsf{pp}) \xrightarrow{\$} (\mathsf{pk}_A, \mathsf{sk}_A)$
- $\mathsf{Gen}_B(\mathsf{pp}) \xrightarrow{\$} (\mathsf{pk}_B, \mathsf{sk}_B)$
- $\mathsf{KA}_A(\mathsf{sk}_A, \mathsf{pk}_B) \to K/\bot$
- $\mathsf{KA}_B(\mathsf{sk}_B, \mathsf{pk}_A) \to K/\bot$

The notation $\xrightarrow{\$}$ means that the function is probabilistic while $\to$ is for deterministic ones. The notation $K/\bot$ means that either some K or an error message $\bot$ is returned.

Q.1 Define the correctness notion for *each* of the three primitives.

Q.2 The INDCPA security notion was defined for PKC in the course. We make a slight change and give a new definition: A PKC is (t, ε)-INDCPAror-secure if for all probabilistic adversary $\mathcal{A}$ limited to a time complexity of t, we have

$$\Pr[x = 1 | b = 0] - \Pr[x = 1 | b = 1] \leq \varepsilon$$

where b is an input bit and x is the output of the following procedure, and the probability is over all probabilistic operations:

input b
$\mathsf{Setup} \xrightarrow{\$} \mathsf{pp}$
$\mathsf{Gen}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$
pick coins at random
$\mathcal{A}(\mathsf{pp}, \mathsf{pk}; \mathsf{coins}) \rightarrow \mathsf{pt}_0$
pick pt_1 at random, of same length at pt_0
$\mathsf{Enc}(\mathsf{pk}, \mathsf{pt}_b) \xrightarrow{\$} \mathsf{ct}$
$\mathcal{A}(\mathsf{pp}, \mathsf{pk}, \mathsf{ct}; \mathsf{coins}) \rightarrow x$
return x

What was changed, compared to the INDCPA definition from the course?
Discuss on the importance of the change.

Q.3 We define the KEM security as follows. A KEM is (t, ε)-$\mathsf{INDCPAror}$-secure if for all probabilistic adversary $\mathcal{A}$ limited to a time complexity of t, we have

$$\Pr[x = 1 | b = 0] - \Pr[x = 1 | b = 1] \leq \varepsilon$$

where b is an input bit and x is the output of the following procedure, and the probability is over all random coins:

1: **input** b
2: $\mathsf{Setup} \xrightarrow{\$} \mathsf{pp}$
3: $\mathsf{Gen}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$
4: $\mathsf{Enc}(\mathsf{pk}) \xrightarrow{\$} (K_0, \mathsf{ct})$
5: pick K_1 at random of same length as K_0
6: $\mathcal{A}(\mathsf{pp}, \mathsf{pk}, \mathsf{ct}, K_b) \xrightarrow{\$} x$
7: **return** x

Given a PKC, construct a KEM.
Prove that if the PKC is correct, then the KEM is correct.
Prove that there exists a constant τ such that for all t and ε, if the PKC is (t, ε)-$\mathsf{INDCPAror}$-secure, then the KEM is $(t - \tau, \varepsilon)$-$\mathsf{INDCPAror}$-secure.

Q.4 Propose a definition for the INDCPAror-security of KA. Given a correct KA, construct a correct KEM.

Show that with the same method as in the previous question, we prove that there exists a constant τ such that for all t and ε, if the KA is (t, ε)-INDCPAror-secure, then the KEM is $(t - \tau, \varepsilon)$-INDCPAror-secure.