Advanced Cryptography — Final Exam

Serge Vaudenay
26.6.2018

- duration: 3h
- any document allowed
- 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 Ciphertext Collision in Semantically Secure Cryptosystems

We consider a public-key cryptosystem \((\text{Gen}, \mathcal{M}, \text{Enc}, \text{Dec})\). We assume perfect correctness, i.e. for all \(s\) and all \(x \in \mathcal{M}\), if \((K_p, K_s) \leftarrow \text{Gen}(1^s)\) then

\[
\Pr[\text{Dec}_{K_s}(\text{Enc}_{K_p}(x)) = x] = 1
\]

Given a probabilistic polynomial-time adversary \(A\), we consider the following game:

\[
\text{Game } \Gamma_A(s):
\]

1. \((K_p, K_s) \leftarrow \text{Gen}(1^s)\)
2. \(X \leftarrow A(K_p)\)
3. \(Y_0 \leftarrow \text{Enc}_{K_p}(X)\)
4. \(Y_1 \leftarrow \text{Enc}_{K_p}(X)\)
5. return \(1_{Y_0 = Y_1}\)

Q.1 Prove that if the cryptosystem is IND-CPA secure, then \(\Pr[\Gamma_A(s) \rightarrow 1]\) is negligible. Hint: construct an IND-CPA adversary with advantage related to \(\Pr[\Gamma_A(s) \rightarrow 1]\).
2 Non-Malleability in Adaptive Security

We consider a public-key cryptosystem \((\text{Gen}, \mathcal{M}, \text{Enc}, \text{Dec})\). We assume perfect correctness, i.e. for all \(s\) and all \(x \in \mathcal{M}\), if \((K_p, K_s) \leftarrow \text{Gen}(1^s)\) then
\[
\Pr[\text{Dec}_{K_s}(\text{Enc}_{K_p}(x)) = x] = 1
\]

Given an adversary in two parts \(A = (A_1, A_2)\), a bit \(b \in \{0, 1\}\), and the security parameter \(s\), we define the IND-CCA game as follows:

**Game IND-CCA\(_b\)(s)**
1: \((K_p, K_s) \leftarrow \text{Gen}(1^s)\)
2: \((X_0, X_1, \sigma) \leftarrow A_1^{O_1}(K_p)\) \(\triangleright \sigma\) is a “state” for \(A_1\) to transmit data to \(A_2\)
3: \(Y \leftarrow \text{Enc}_{K_p}(X_b)\)
4: \(b' \leftarrow A_2^{O_2}(\sigma, Y)\)
5: return \(b'\)

where the oracles \(O_1\) and \(O_2\) are defined as follows:

**Oracle \(O_1(y)\):**
1: return \(\text{Dec}_{K_s}(y)\)

**Oracle \(O_2(y)\):**
2: if \(y = Y\) then
3: abort the game
4: end if
5: return \(\text{Dec}_{K_s}(y)\)

We define the advantage
\[
\text{Adv}_{\text{A}}^{\text{IND-CCA}}(s) = \Pr[\text{IND-CCA}_1^b(s) \rightarrow 1] - \Pr[\text{IND-CCA}_0^b(s) \rightarrow 1]
\]

We say that the cryptosystem is IND-CCA secure if for all probabilistic polynomial time (PPT) adversary \(A\), \(\text{Adv}_{\text{A}}^{\text{IND-CCA}}(s)\) is negligible.

**Q.1** The definition of IND-CCA security which was given in the course (Def.5.5 on p.55–56 in the lecture notes, or slide p.404) was based on an interactive game between an adversary and a challenger. Prove that the two styles of definition for IND-CCA security are equivalent. (Carefully construct \((A_1, A_2)\) from an interactive adversary and an interactive adversary from \((A_1, A_2)\).)

**Q.2** Let \(A = (A_1, A_2)\) be an IND-CCA adversary. We define another IND-CCA adversary as follows:

**Algorithm \(B_1^{O_1}(K_p)\)**
1: simulate \(A_1^{O_1}(K_p) \rightarrow (X_0, X_1, \sigma)\)
2: if \(X_0 = X_1\) then
3: set \(\sigma' \leftarrow (\sigma, 1)\)
4: pick an arbitrary \(X\) such that \(X \neq X_1\)
Algorithm $\mathcal{B}_2^{O_2(\cdot)}(\sigma', Y)$
10: parse $\sigma' = (\sigma, c)$
11: if $c = 1$ then
12: return $0$
13: else
14: simulate $\mathcal{A}_2^{O_2(\cdot)}(\sigma, Y) \to b'$
15: return $b'$
16: end if

Prove that

$$\text{Adv}_{A}^{\text{IND-CCA}}(s) = \text{Adv}_{B}^{\text{IND-CCA}}(s)$$

Deduce that we can always assume $X_0 \neq X_1$ in an IND-CCA adversary.

We now define the NM-CCA game (for non-malleability) as follows:

**Game NM-CCA$_A^b(s)$**
1. $(K_p, K_s) \leftarrow \text{Gen}(1^s)$
2. $(M, \sigma) \leftarrow \mathcal{A}_1^{O_1(\cdot)}(K_p)$ \hspace{1em} $\triangleright$ $\sigma$ is a “state” which allows $\mathcal{A}_1$ to transmit data to $\mathcal{A}_2$
3. $X_0 \leftarrow M$ \hspace{1em} $\triangleright$ $M$ is a sampling algorithm defined by $\mathcal{A}_1$
4. $X_1 \leftarrow M$ \hspace{1em} $\triangleright$ we sample two independent plaintexts using $M$
5. $Y \leftarrow \text{Enc}_{K_p}(X_1)$
6. $(R, Y'_1, \ldots, Y'_n) \leftarrow \mathcal{A}_2^{O_2(\cdot)}(\sigma, Y)$ \hspace{1em} $\triangleright$ $R$ is a poly. algo. returning a boolean
7. $X'_i \leftarrow \text{Dec}_{K_s}(Y'_i)$, $i = 1, \ldots, n$
8. if $Y \notin \{Y'_1, \ldots, Y'_n\}$ and $\perp \notin \{X_1, \ldots, X'_n\}$ and $R(X_b, X'_1, \ldots, X'_n)$ then
9: return $1$
10: else
11: return $0$
12: end if

We use the same oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ as for IND-CCA. We define

$$\text{Adv}_{A}^{\text{NM-CCA}}(s) = \text{Pr}[\text{NM-CCA}_A^1(s) \to 1] - \text{Pr}[\text{NM-CCA}_A^0(s) \to 1]$$

We say that the cryptosystem is NM-CCA secure if for all probabilistic polynomial time (PPT) adversary $\mathcal{A}$, $\text{Adv}_{A}^{\text{NM-CCA}}(s)$ is negligible.

The goal of this exercise is to show the equivalence between NM-CCA security and IND-CCA security.
Q.3 We assume that $\mathcal{M}$ has a group structure (additively denoted), with at least two different elements 0 and 1, 0 being neutral. Assume that there is a polynomial algorithm $\text{Inc}$ such that for all $s$,

$$\Pr[\text{Dec}_{Ks}(\text{Inc}_{Kp}(\text{Enc}_{Kp}(X))) = X + 1] = 1$$

for $(K_p, K_s) \leftarrow \text{Gen}(1^*)$. By constructing an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, prove that the cryptosystem is not NM-CCA secure.

(The precision of the proof is important.)

HINT: use $M$ sampling in a set of two different plaintexts and $R$ defined by $R(X, X') = 1_{X' = X + 1}$.

Q.4 Given an NM-CCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we construct an IND-CCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ as follows:

**Algorithm $\mathcal{B}_1^{O_1}(K_p)$**
1: simulate $\mathcal{A}_1^{O_1}(K_p) \rightarrow (M, \sigma)$
2: sample $z_0 \leftarrow M$
3: sample $z_1 \leftarrow M$
4: set $\sigma' \leftarrow (z_0, z_1, \sigma)$
5: return $(z_0, z_1, \sigma')$

**Algorithm $\mathcal{B}_2^{O_2}(\sigma', Y)$**
6: parse $\sigma' = (z_0, z_1, \sigma)$
7: simulate $\mathcal{A}_2^{O_2}(\sigma, Y) \rightarrow (R, Y'_1, \ldots, Y'_n)$
8: for $i = 1, \ldots, n$ do
9: if $Y = Y'_i$ then return 0
10: $X'_i \leftarrow \mathcal{O}_2(Y'_i)$
11: if $X'_i = \bot$ then return 0
end for
12: compute $b' \leftarrow R(z_1, X'_1, \ldots, X'_n)$
13: return $b'$

Prove that

$$\text{Adv}_\mathcal{B}^{\text{IND-CCA}}(s) = \text{Adv}_\mathcal{A}^{\text{NM-CCA}}(s)$$

Deduce that IND-CCA security implies NM-CCA security.

Q.5 We assume that $\mathcal{M}$ has at least four elements.

Given an IND-CCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we construct an NM-CCA adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ as follows:

**Algorithm $\mathcal{B}_1^{O_1}(K_p)$**
1: simulate $\mathcal{A}_1^{O_1}(K_p) \rightarrow (z_0, z_1, \sigma)$
2: define $M$ sampling in $\{z_0, z_1\}$ with uniform distribution
3: set $\sigma' \leftarrow (\sigma, K_p, z_0, z_1)$
4: return $(M, \sigma')$

**Algorithm $\mathcal{B}_2^{O_2}(\sigma', Y)$**
5: parse $\sigma' = (\sigma, K_p, z_0, z_1)$
6: take an injective function $T$ on $\mathcal{M}$ such that $T(z_0) \not\in \{z_0, z_1\}$ and $T(z_1) \not\in \{z_0, z_1\}$
7: simulate $\mathcal{A}^{O_2(\cdot)}_2(\sigma, Y) \rightarrow b'$
8: $Y'' \leftarrow \text{Enc}_{K_{b'}}(T(z_{b''}))$
9: define $R(X, X') = 1_{T(X)=X'}$
10: return $(R, Y'')$

Prove that
\[
\text{Adv}^\text{NM-CCA}_B(s) = \frac{1}{2} \text{Adv}^\text{IND-CCA}_A(s)
\]

Deduce that NM-CCA security implies IND-CCA security.

HINT$_1$: assume without loss of generality that $z_0 \neq z_1$
HINT$_2$: compute $\Pr[X_0 = z_{b''} \mid X_1 = z_{b''}]$, $\Pr[X_1 = z_{b''} \mid X_1 = z_1]$, and $\Pr[X_1 = z_{b''} \mid X_1 = z_0]$. 

3 Unruh Transform from $\Sigma$ to NIZK

We consider a $\Sigma$ protocol $(P, V)$ for a relation $R$. We let $E$ be the set of challenges. Given some parameters $t$ and $m \geq 2$, we define the following non-interactive zero-knowledge proof (NIZK), with input $(x, w)$ such that $R(x, w)$ holds:

**Algorithm** Proof$(x, w)$:
1: for $i = 1$ to $t$ do
2: pick a sequence of fresh coins $\rho_i$
3: set $a_i \leftarrow P(x, w; \rho_i)$
4: for $j = 1$ to $m$ do
5: pick $e_{i,j} \in E - \{e_{i,1}, \ldots, e_{i,j-1}\}$ at random
6: set $z_{i,j} \leftarrow P(x, w, e_{i,j}; \rho_i)$
7: set $h_{i,j} \leftarrow G(z_{i,j})$
8: end for
9: end for
10: set $h \leftarrow H(x, (a_i, (e_{i,j}, h_{i,j})_{j=1,\ldots,m})_{i=1,\ldots,t})$
11: set $(J_1, \ldots, J_t) \leftarrow h$
12: set $z_i = z_{i,J_i}$ for $i = 1, \ldots, t$
13: set $\pi = (a_i, (e_{i,j}, h_{i,j})_{j=1,\ldots,m}, z_i)_{i=1,\ldots,t}$
14: return $\pi$

This algorithm uses two random oracles $G$ and $H$. Oracle $H$ is assumed to return a $t$-tuple of integers between 1 and $m$. We use the following verification algorithm (with some missing step):

**Algorithm** Verify$(x, \pi)$:
1: parse $\pi = (a_i, (e_{i,j}, h_{i,j})_{j=1,\ldots,m}, z_i)_{i=1,\ldots,t}$
2: set $h \leftarrow H(x, (a_i, (e_{i,j}, h_{i,j})_{j=1,\ldots,m})_{i=1,\ldots,t})$
3: set $(J_1, \ldots, J_t) \leftarrow h$
4: verify $\cdots$
5: verify $V(x, a_i, e_{i,j}, z_i)$ for $i = 1, \ldots, t$
6: verify $h_{i,J_i} = G(z_i)$ for $i = 1, \ldots, t$
7: return 1 if all verifications passed

Q.1 By taking the verification with the missing step, give an algorithm to forge a proof given $x$ but without the knowledge of $w$.
Which step should be added to have a sound proof?

Q.2 With the new verification step from the last question, given an algorithm with complexity $O(m^t)$ to forge a valid $\pi$ from $x$ but without $w$.

Q.3 Construct a simulator in the random oracle model to show that the protocol is non-interactive zero-knowledge.

Q.4 Let $P^*(x)$ be an algorithm taking $x$ as input, interacting with $G$ and $H$, and forging a valid $\pi$ with probability $p$. Use the next questions to prove that there is an extractor who can run $P^*$ once to extract a witness $w$ for $x$ with probability at least $p - \text{negl}$. 

---

6
Q.4a Transform $P^*$ into an algorithm $P'$ who either aborts or makes a valid $\pi$. It returns $\pi$ with probability $p$, and a complexity similar to $P^*$.

Q.4b Construct an extractor $E$ on the previous $P'$ such that by observing only one execution of $P'$ with all queries to $G$ and $H$, either $P'$ aborts, or $E$ finds a witness for $x$, or $E$ aborts. But the probability that $E$ aborts is bounded by $n_G n_H m t N^{-1} + n_H m^{-t}$, where $n_G$ is the number of queries to $G$, $n_H$ is the number of queries to $H$, and $N$ is the size of the range of $G$.

Hint: say that a query $q$ to $H$ is good if it can be parsed in the form

$$q = x, (a_i, (c_{i,j}, h_{i,j})_{j=1,...,m})_{i=1,...,t}$$

Consider an extractor which aborts if any fresh query to $G$ returns a value $h_{i,j}$ which is included in a previous good query $q$ to $H$. Define another abort condition and extract a witness in remaining cases.