

Instructions.

- Study the notes.
- Study the solutions to the homework exercises.
- Try the following additional practice exercises. For each exercise the full solution or a sketch of the solution is provided. Don't read the solutions first; try to solve them yourselves.

1. We proved in class that the RSA signature scheme is UF-CMA provided that you have a “full-domain” hash function $\mathcal{H} : \{0, 1\}^* \rightarrow \mathbb{Z}_n^*$ that behaves like a random oracle. Obviously the range \mathbb{Z}_n^* is rather non-standard: typically hash functions produce a bitstring as output. Suppose that the number $n = pq$ with p, q two $(k + 1)$ -bit numbers. Now suppose that you have a hash function $\mathcal{H} : \{0, 1\}^* \rightarrow \{0, 1, \dots, 2^{2k-1}\}$. Show that it is possible to use this function to define the RSA signature and prove it secure in the UF-CMA sense.

Proof Sketch. The primary problem in the proof is that we cannot return values of the form $t^e \bmod n$ to the adversary as answers from the random oracle and expect the adversary to behave in the same way as when there is no cheating with the table of the random oracle. This is because the uniform distribution over $\{0, 1, \dots, 2^{2k-1}\}$ is statistically different from the uniform distribution over \mathbb{Z}_n^* (find an easy distinguisher). So we have to change the way that we answer such queries; luckily the modification is relatively simple: you can keep trying different t 's till you get one in the right range and return this as an answer. You can show that the statistical distance of this distribution is very close to the uniform over $\{0, 1, \dots, 2^{2k-1}\}$ (recall the first homework for this point). A second problem will happen when you will have to plug the challenge y into the table. Obviously we cannot ask for another y instead! the good news is that we can use the homomorphic properties of the RSA function: indeed we may try the following experiment: choose s at random from \mathbb{Z}_n^* and compute $y' = s^e \cdot y \bmod n$. If it happens that $y' \in \{0, 1, \dots, 2^{2k-1}\}$ then we use this value in the special location in the table. When the adversary returns a signature σ^* for the message m^* with $\mathcal{H}(m^*) = y'$ then we can find an e -th root by computing $\sigma^* \cdot s^{-1} \bmod n$.

2. The existential discrete logarithm problem is a decisional problem (i.e., it requires only a yes-no answer) that is defined as follows: given a prime p , an element $g \in \mathbb{Z}_p^*$ and another element $h \in \mathbb{Z}_p^*$ answer the following question:

$$\exists? x \in \mathbb{Z} : g^x \equiv h \pmod{p}$$

Assume that $p = 2q_1q_2 + 1$ and q_1, q_2 are both prime numbers that are known. Find an efficient algorithm for solving the above problem. Argue why your algorithm is correct.

Proof Hint. Observe that since you know the factorization of $p - 1$, you know the order of all subgroups of $\langle g \rangle$. The idea here is to test whether g, h belong to the same subgroup.

3. Joe Smart realizing that RSA is not IND-CPA secure (show why), proposes the following “salting” variant of the cryptosystem: let $\nu = \lfloor n \rfloor$; to encrypt a plaintext $m \in \{0, 1\}^\nu$ select a random integer $r \leftarrow [0, 2^\nu)$ and form the ciphertext $\langle r^e \bmod n, r \oplus m \rangle$ where \oplus denotes the x-or operation. Decryption of a ciphertext $\langle C, D \rangle$ is accomplished by computing $(C^d \bmod n) \oplus D$. Joe suggests that his cryptosystem is probabilistic and thus (sic) “very secure.” Show how an IND-CPA attacker can break Joe’s cryptosystem.

Proof: Consider the following IND-CPA attacker:

- (a) (play stage) Input: e, n .
- (b) (play stage) Fix $m_0 = 1 \dots 1$ and $m_1 = 0 \dots 0$ from $\{0, 1\}^\nu$.
- (c) (play stage) Output $aux = \langle e, n, m_0, m_1 \rangle, m_1, m_2$.
- (d) (guess stage) Input: $aux = \langle e, n, m_0, m_1 \rangle, c$.
- (e) (guess stage) Parse c as a tuple $\langle C, D \rangle$.
- (f) (guess stage) If $(D \oplus m_0)^e = C \pmod{n}$ then return 0 else return 1.

Observe that if $c = \langle C, D \rangle$ is a ciphertext encrypting m_0 it holds that $\langle C, D \rangle = \langle r^e \pmod{n}, r \oplus m_0 \rangle$, where r is a random ν -bit number.

It follows that it will hold $(D \oplus m_0)^e = (r \oplus m_0 \oplus m_0)^e = r^e \pmod{n}$ and thus the above attacker will be successful in returning 0.

On the other hand if c is a ciphertext encrypting m_1 it holds that $\langle C, D \rangle = \langle r^e \pmod{n}, r \oplus m_1 \rangle$, where r is a random ν -bit number.

It holds now that $(D \oplus m_0)^e = (r \oplus m_1 \oplus m_0)^e \pmod{n}$. Observe that $m_1 \oplus m_0 = m_0$ and in general, for any ν -bitstring it holds that $r \oplus m_0 = \bar{r}$ where \bar{r} is a bitstring as r with all bits flipped. The attacker will fail to return the correct answer (which is 1) only in the case $r^e = \bar{r}^e \pmod{n}$. But this event can only happen if $r = \bar{r}$ (why?) which in turn is an impossible event.

4. Consider the following attack game against a public-key cryptosystem $\langle \text{GEN}, \text{ENC}, \text{DEC} \rangle$ for which it holds that the plaintext space is $\{0, 1\}^k$, that we call the “known-or-random” chosen-plaintext-attack KOR-CPA game:

1. $\langle \text{pk}, \text{sk} \rangle \leftarrow \text{GEN}(1^\lambda)$.
2. $\langle aux, m \rangle \leftarrow \mathcal{A}(\text{fix}, 1^\lambda, \text{pk})$.
3. $b \stackrel{r}{\leftarrow} \{0, 1\}$.
4. if $b = 0$ then $c \leftarrow \text{ENC}(\text{pk}, m)$.
5. if $b = 1$ then $m' \stackrel{R}{\leftarrow} \{0, 1\}^k$; $c \leftarrow \text{ENC}(\text{pk}, m')$.
6. $b^* \leftarrow \mathcal{A}(\text{guess}, aux, c)$.
7. if $b = b^*$ return 1 else return 0.

Informally a KOR-CPA adversary selects a fixed message m in his fix stage of operation (given the public-key of course). Then he receives either (i) the encryption of his fixed message, or (ii) the encryption of a random plaintext (with probability 1/2 either of the two cases). Then, he tries to guess which of the two cases happened.

A KOR-CPA adversary \mathcal{A} is successful if it holds that its winning probability T when playing the above game $\text{Prob}[T]$ is significantly different from 1/2 (by a non-negligible fraction). Show that a cryptosystem that is IND-CPA secure is also KOR-CPA secure.

Proof. Let \mathcal{A} be an KOR-CPA attacker. We will show how to derive a IND-CPA attacker with essentially the same probability of success.

Below we describe the IND-CPA attacker called \mathcal{A}'

- (a) (stage 1) Input $1^\lambda, \text{pk}$.
- (b) (stage 1) Simulate $\mathcal{A}(\text{fix}, 1^\lambda, \text{pk})$ to obtain aux, m .
- (c) (stage 1) Choose $m' \leftarrow_R \{0, 1\}^k - \{m\}$.
- (d) (stage 1) Output $\text{aux}, m_0 = m, m_1 = m'$.
- (e) (stage 2) Input: $\langle \text{aux}, c \rangle$ so that c is with $1/2$ probability an encryption of m_0 , and with probability $1/2$ an encryption of m_1 .
- (f) (stage 2) Simulate $\mathcal{A}(\text{guess}, \text{aux}, c)$ to obtain b^* .
- (g) (stage 2) Return b^* .

First observe that \mathcal{A}' adheres to the syntax of the IND-CPA attack.

Moreover observe that the input c to the guess stage of the above attack is with probability $1/2$ the encryption of the fixed message m , and with probability $1/2$ it is an encryption of a plaintext m' drawn at random from $\{0, 1\}^k - \{m\}$.

Now observe that the input that the KOR-CPA attacker \mathcal{A} expects in its guess stage is exactly of this form with the only difference the fact that the ciphertext c is with probability $1/2$ the encryption of a plaintext $m' \leftarrow_R \{0, 1\}^k$.

It is clear that the probability distributions of c : (i) in an actual KOR-CPA attack and (ii) in the simulation performed by \mathcal{A} above, have statistical distance $1/2^k$ (compare the distributions $m' \leftarrow_R \{0, 1\}^k$ and $m' \leftarrow_R \{0, 1\}^k - \{m\}$). Using the above ideas you can prove that the distance between the advantage of a given KOR-CPA attacker and the constructed IND-CPA attacker above is at most $1/2^k$. It follows that IND-CPA security will imply KOR-CPA security (assuming that 2^{-k} is negligible in λ).