Continuous Control

Date: 22.10.2009
TOTAL: 50 points

Notice: the number of point correspond approximatively to the number of minutes needed for solving an exercise.

Exercise 1
(10 points) Zheng & Seberry in 1993 proposed the following encryption scheme:

\[ f(r)||G(r) \oplus (x||H(x)) \]

where \( x \) is the plain text, \( f \) is a one way trap-door function (like RSA), \( G \) and \( H \) are two hash functions, \( || \) denotes the concatenation of bitstrings and \( \oplus \) is the exclusive-or operator. Give the associated decryption algorithm.

Solution:
First decrypt \( f(r) \) to get \( r \), then compute \( G(r) \oplus (x||H(x)) \) to get \( x||H(x) \) then check if the application of \( H \) on the first element of the concatenation is equal to the second to be sure that you get the right plaintext.

Exercise 2
(15 points)
1. Recall the IND-CPA definition.
2. Prove that

\[
\text{Adv}_{S,A}^{\text{IND}}(\eta) = Pr[b' \leftarrow \text{IND}^1(A) : b' = 1] - Pr[b' \leftarrow \text{IND}^0(A) : b' = 1] = 2Pr[b \leftarrow \{0,1\}, b' \leftarrow \text{IND}^b(A) : b' = b] - 1
\]

Solution:

\[
2Pr[b' \leftarrow \text{IND}^b(A) : b' = b] - 1 = 2\left(Pr[b' \leftarrow \text{IND}^1(A) : b' = 1]Pr[b = 1] + Pr[b' \leftarrow \text{IND}^0(A) : b' = 0]Pr[b = 0]\right) - 1
\]

\[
= 2\left(Pr[b' \leftarrow \text{IND}^1(A) : b' = 1]\frac{1}{2} + Pr[b' \leftarrow \text{IND}^0(A) : b' = 0]\frac{1}{2}\right) - 1
\]

\[
= Pr[b' \leftarrow \text{IND}^1(A) : b' = 1] + 1 - Pr[b' \leftarrow \text{IND}^0(A) : b' = 1] - 1
\]

\[
= Pr[b' \leftarrow \text{IND}^1(A) : b' = 1] - Pr[b' \leftarrow \text{IND}^0(A) : b' = 1]
\]

\[
= \text{Adv}_{S,A}^{\text{IND}_{XXX}}(\eta)
\]
other idea:
\[ Pr[b' \xleftarrow{R} \text{IND}^1(A) : b' = 1] - Pr[b' \xleftarrow{R} \text{IND}^0(A) : b' = 1] + 0 \]
where
\[ 0 = Pr[b' \xleftarrow{R} \text{IND}^0(A) : b' = 0] - Pr[b' \xleftarrow{R} \text{IND}^0(A) : b' = 0] \]
we get
\[ Pr[b' \xleftarrow{R} \text{IND}^1(A) : b' = 1] + Pr[b' \xleftarrow{R} \text{IND}^0(A) : b' = 0] - Pr[b' \xleftarrow{R} \text{IND}^0(A) : b' = 1] - Pr[b' \xleftarrow{R} \text{IND}^0(A) : b' = 0] \]
because
\[ Pr[b' \xleftarrow{R} \text{IND}^1(A) : b' = 1] + Pr[b' \xleftarrow{R} \text{IND}^0(A) : b' = 0] = 2Pr[b' \xleftarrow{R} \text{IND}^b(A) : b' = b] \]
due to
\[ Pr[b' \xleftarrow{R} \text{IND}^b(A) : b' = b] = Pr[b' \xleftarrow{R} \text{IND}^1(A) : b' = 1]Pr[b = 1] + Pr[b' \xleftarrow{R} \text{IND}^0(A) : b' = 0]Pr[b = 0] \]

\[ Pr[b' \xleftarrow{R} \text{IND}^1(A) : b' = 1]Pr[b = 1] + Pr[b' \xleftarrow{R} \text{IND}^0(A) : b' = 0]Pr[b = 0] = 1 \]

Exercise 3
(10 points) We recall the following definitions for hash functions:

- **2nd Preimage resistance:**
  Given an hash function \( h \) and an input \( x \), it is computationally infeasible to compute \( x' \) such that \( x \neq x' \) and
  \[ h(x') = h(x) \]

- **Collision resistance:**
  Given an hash function \( h \) it is computationally infeasible to compute \( x' \) and \( x \) such that \( x \neq x' \) and
  \[ h(x') = h(x) \]

Using a reduction proof, show that solving collision resistance is easier than the 2nd preimage resistance.

**Solution:**

- Consider an adversary \( B \) against 2nd preimage resistance, using \( B \) we will break collision resistance. It means that we have to build an adversary \( A \) which finds a collision given an hash function \( h \).
  - Pick \( x \) randomly
  - Call \( B \) with \( h \) and \( x \)
  - \( B \) outputs \( x' \) such that \( x' \neq x \) and \( h(x') = h(x) \)
Exercise 4
(15 points) We recall Elgamal algorithm

Key generation: Alice chooses a prime number \( p \) and a group generator \( g \) of \((\mathbb{Z}/p\mathbb{Z})^*\) and \( a \in (\mathbb{Z}/(p-1)\mathbb{Z})^* \).

Public key: \((p, g, h)\), where \( h = g^a \mod p \).

Private key: \( a \)

Encryption: Bob chooses \( r \in_R (\mathbb{Z}/(p-1)\mathbb{Z})^* \) and computes \((u, v) = (g^r, M h^r)\)

Decryption: Given \((u, v)\), Alice computes \( M \equiv_p v \div u^a \)

1. Prove that Elgamal is not IND-CCA2

Hint: using twice, even partialy, the same random is dangerous.

Solution : \( A_1 \) chooses two distinct messages \( m_1 = m_2 \)
\( A_2 \) gets \( y = (i, c) = Elgamal(m_b) \) and ask the decryption of \((i, c')\) to the decryption oracle
He obtains \( m' = \frac{c'}{i^x} \) hence he deduce that

\[
  i^x = \frac{c'}{m'}
\]

Hence he outputs 1 if \( \frac{c'}{i^x} = m_1 \), 0 otherwise.