Security models

1st Semester 2014/2015

F.Autreau
P.Lafourcade
M. Gagné
JL. Roch

A. Kumar
P. Lafourcade
M. Gagné

SET 3

Date: 27.11.2014

Disclaimer: This exercise sheet is to be treated as a mock examination paper. However, the answers will not be graded.

Exercise 1 (Strong RSA Assumption)
The famous RSA problem is defined in the following manner.

Definition 1 (RSA Problem) *Given an RSA public key (N, e) and a ciphertext $c = m^e \bmod N$, to compute m.*

To solve the RSA problem, an adversary, who does not know the private key must nonetheless invert the RSA function ($m^e \bmod N$). The **RSA Assumption** is that the RSA problem is hard to solve when the modulus N is sufficiently large and randomly generated and the plaintext m is a random integer between 0 and $N - 1$. The RSA problem is the basis for the security of RSA public-key encryption as well as RSA digital signature scheme.

Barić and Pfitzmann introduced a new assumption in EUROCRYPT'97 called **Strong RSA Assumption**. To this end, the authors relax the RSA problem and define:

Definition 2 (Weak RSA Problem) *Given an RSA modulus N and a ciphertext c, to compute m and an (odd) public exponent $e \geq 3$ such that $c = m^e \bmod N$.*

This may well be easier than solving the RSA problem, whence the assumption that it is hard is stronger than the RSA Assumption. The Strong RSA Assumption is the basis for a variety of cryptographic constructions, in particular *cryptographic accumulators*.

The goal of this exercise is to analyze the veracity of this assumption.

1. Suppose that an adversary tries to solve the Weak RSA Problem by choosing a random e first. Is the ensuing attack feasible ?
2. What if he chooses a random y first ?
3. Suppose that an adversary tries to find d and e with $m = c^d \bmod N$ and $(c^d)^e = c \mod N$. Is this attack feasible ?
4. Can you think of any other attacks ? Conclude.

Exercise 2 (Security of OAEP)
Bellare and Rogaway in EUROCRYPT'94 propose the following generic construction to design an encryption scheme using any trapdoor one-way permutation f onto $\{0,1\}^k$. The construction works in the random oracle model.

We suppose that there are two hash functions:

$$\mathcal{G} : \{0,1\}^{k_0} \to \{0,1\}^{k-k_0} \quad \text{and} \quad \mathcal{H} : \{0,1\}^{k-k_0} \to \{0,1\}^{k_0},$$

for some k_0. We also need n and k_1 which satisfy $k = n + k_0 + k_1$. Then the encryption scheme $\mathsf{OAEP} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ can be described as follows:

- $\mathcal{K}(1^k)$: specifies an instance of the function f, and its inverse f^{-1}. The public key $\mathtt{pk}$ is therefore f, and the private key $\mathtt{sk}$ is f^{-1}.
- $\mathcal{E}_{\mathtt{pk}}(m, r)$: given a message $m \in \{0,1\}^n$, and a random value $r \xleftarrow{\$} \{0,1\}^{k_0}$, the encryption algorithm $\mathcal{E}_{\mathtt{pk}}$ computes:

$$s = (m||0^{k_1}) \oplus \mathcal{G}(r) \quad \text{and} \quad t = r \oplus \mathcal{H}(s),$$

and outputs the ciphertext $c = f(s, t)$.

1. Write the decryption algorithm $\mathcal{D}_{\mathtt{sk}}(\cdot)$ for OAEP.
2. Find a function f to disprove that the scheme is IND-CCA secure.

Exercise 3 (E-Voting)
We modify the El-Gamal encryption as follows to encrypt small integers:

- Compute $h = g^x$ where $g, h \in \mathbb{Z}_p^\star$ for a large prime p and g is a generator of $\mathbb{Z}_p^\star$.
- Public key is (p, g, h) and the private key is x.
- Encryption of $n \in \mathbb{Z}_p^\star$ is $c = (g^k, g^n \cdot h^k)$, where k is a random number between 0 and $p-1$.

1. Give the decryption algorithm associated to El-Gamal. *Hint:* since we assume that the encrypted integer n is small, it is ok to use up to n operations to recover n.
2. If $\{m\}_k = (a, b)$ and $\{n\}_k = (c, d)$ are two ciphertexts, we define $\{m\}_k \cdot \{n\}_k = (a \cdot c, b \cdot d)$. Prove that, under this operation, El-Gamal is a homomorphic encryption ($\{m\}_k \cdot \{n\}_k$ will be a ciphertext for $m + n$).
3. Explain 5 security properties that an e-voting system should satsify.
4. Naïve voting system: We consider that 1 and 0 are the two possible ballots for an elections. A server publishes his public RSA key (N, e). Each voter encrypts his vote, 0 or 1, as $RSA_{(N,e)}(0)$ or $RSA_{(N,e)}(1)$ respectively. At the end of the election the server decrypts all received messages and counts the votes. Show how an attacker eavesdropping on the network can learn everybody's vote.
5. We improve this scheme by replacing RSA by the variant of El-Gamal above. At the end, instead of decrypting all the votes, the server uses the homomorphic property to sum all the ciphertexts, and decrypts only this final ciphertext to obtain the tally.
   - Explain why this IND-CPA encryption can prevent the previous attack.

   (∗) Find how a voter (who does not have a decryption oracle at any point of the attack) could cheat in this protocol to favor his candidate.
   - Suppose that an attacker A wants to force a voter V to vote for candidate 0. If A intercepts the ciphertext containing V's vote, how could he ask V to prove that he voted for candidate 0?
   - Propose a solution in order to avoid this attack found in question (∗).