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

*A. Kumar*
*P. Lafourcade*
*M. Gagné*

# SET 3 (Solutions)

Date: 27.11.2014

**Disclaimer:** Solutions are not provided for the last exercise on e-voting.

**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 $m$ first ?

3. Suppose that an adversary tries to find $d$ and $e$ with $m = c^d \bmod N$ and $(c^d)^e = c \bmod N$. Is this attack feasible ?

4. Can you think of any other attacks ? Conclude.

***Solution :***

1. *If the adversary decides to first choose the public exponent $e$ and then find the required $m$ such that $c = m^e \bmod N$, then the same adversary can also solve the RSA problem. But, according to the RSA Assumption, the RSA problem is hard to solve. Hence, by contradiction, this attack on the Weak RSA problem is not feasible.*

2. *If the adversary chooses a random $m$ first and then tries to find the required $e$ such that $c = m^e \bmod N$, this adversary should be able to solve the Discrete Logarithm problem over the group $G = \langle m \rangle$.*

3. *If the adversary chooses $d$ and computes $m = c^d \mod N$ and then looks for $e$ such that $(c^d)^e \mod N = c$ then ord(c) divides $f := de - 1$. He can therefore also solve the RSA problem for the same $N$ and $c$: Let a random exponent $e'$ be given. It is sufficient to consider the case where $e'$ is prime and no factor of $f$. Then we set $d' := e'^{-1} \mod f$ and obtain $(c^{d'})^{e'} = c \mod N$, because ord(c) divides $d'e' - 1$.*

4. *One may imagine that the adversary chooses a weak public exponent, and uses attacks on the RSA Problem to solve the Weak RSA problem. However, these attacks require that the adversary has in position the encryption of certain number of messages using the same exponent but different moduli. Other attacks require that the messages have a polynomial relationship and are encrypted using the same modulus. Hence, such attacks do not work here.*

   *This however does not prove that the Strong RSA Assumption holds, since it is quite likely that an adversary may come up with some other technique to solve the problem.*

**Exercise 2 (Security of OAEP)**
We have seen in a previous exercise sheet that the textbook RSA is not even IND-CPA secure. The question that remains is how may one turn such an encryption scheme into an IND-CPA secure scheme and furthermore into an IND-CCA secure scheme. To this end, 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 and since RSA function is a one-way trapdoor permutation, the construction transforms RSA into an IND-CCA secure scheme.

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 $\mathsf{OAEP}$.

2. Find a function $f$ to disprove that the scheme is IND-CCA secure. This would imply that the construction is not as "*generic*" as it sounds.

***Solution :***

1. *$\mathcal{D}_{\mathtt{sk}}(c)$ : using the private key $\mathtt{sk} = f^{-1}$, the decryption algorithm $\mathcal{D}_{\mathtt{sk}}$ extracts:*

   $$(s||t) = f^{-1}(c) \text{ and } r = t \oplus \mathcal{H}(s) \text{ and } m = s \oplus \mathcal{G}(r).$$

   *If the $k_1$ least significant bits of $m$ are 0s, then the algorithm returns the $n$ most significant bits of $m$, otherwise it returns $\perp$ ("reject").*

2. *The counter example was provided by Shoup and makes use of the ad-hoc notion of an* XOR-malleable *trapdoor one-way permutation. For such permutation $f_{xm}$, one can compute $f_{xm}(x \oplus a)$ from $f_{xm}(x)$ and $a$, with non-negligible probability. Let us define a permutation $f(s||t)$ as $s||f_{xm}(t)$. Start with a challenge ciphertext $y = f(s||t) = s||u$ where,*

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

*Since, $f$ is identity on its leftmost part, we know $s$ and can define $\Delta = \delta||0^{k_1}$, for any random string $\delta$, and $s' = s \oplus \Delta$. We then set $t' = r \oplus \mathcal{H}(s') = t \oplus (\mathcal{H}(s) \oplus \mathcal{H}(s'))$. The XOR-malleability of $f_{xm}$ allows one to obtain $u' = f_{xm}(t')$ from $u = f_{xm}(t)$ and $\mathcal{H}(s) \oplus \mathcal{H}(s')$, with significant probability. Finally, $y' = s'||u'$ is a valid ciphertext of $m' = m \oplus \delta$, built from $r' = r$, since:*

$$t' = f_{xm}^{-1}(u') = t \oplus (\mathcal{H}(s) \oplus \mathcal{H}(s')) = \mathcal{H}(s') \oplus r \text{ and } r' = \mathcal{H}(s') \oplus t' = r,$$

*and*

$$s' \oplus \mathcal{G}(r') = \Delta \oplus s \oplus \mathcal{G}(r) = \Delta \oplus (m||0^{k_1}) = (m \oplus \delta)||0^{k_1}.$$

*We note that the above definitely contradicts adaptive chosen-ciphertext security: asking the decryption of $y'$ after having received the ciphertext $y$, an adversary obtains $m'$ and easily recovers the actual cleartext $m$ from $m'$ and $\delta$.*