Homework 2 (Due April 10, 2018)

This assignment is to be returned on April 10, during tutorial. Answers may be typewritten or in legible writing in English or French, to your convenience. The quality, precision and concision of the arguments will play an important role in the overall grading process.

1 First results

In this homework, we will let λ be the security parameter. All sentences like "the algorithm is polynomial time" or "the advantage is negligible" assume that this is polynomial in λ or negligible in λ.
The Euler's phi function φ counts the number of elements between 1 and n that are co-prime with n. More formally, for any positive integer n, it is defined by $\varphi(n) = \text{Card}(\{m \in \{1, \cdots, n\} \text{ s.t. } \gcd(m,n) = 1\})$. Recall that if $n = \prod_i p_i^{\alpha_i}$ with the p_i's distinct and prime, and $\alpha_i \geq 1$ for all i, then we have $\varphi(n) = \prod_i (p_i - 1) p_i^{\alpha_i - 1}$.
In the following, we will consider two odd prime numbers p and q of λ bits, and we let $N = pq$.

1. Show that if $\varphi(N)$ and N are known, then it is possible to compute p and q in polynomial time.

2. Let e, d be integers such that $ed = 1 \bmod \varphi(N)$. Show that for all $x \in \mathbb{Z}_N$, we have $(x^e)^d = x \bmod N$.

   Hint: *Use the Chinese remainder theorem.*

3. Can we compute d efficiently from e and $\varphi(N)$?

Definition 1 (The RSA problem and RSA assumption)**.** *Let (p,q) be two prime numbers of λ bits, chosen according to some distribution D_1. Let $N = pq$ and let $e \in (\mathbb{Z}/\varphi(N)\mathbb{Z})^\times$ be sampled according to some distribution D_2.*[1] *The* RSA problem *asks, given (N, e, r^e) for some r chosen uniformly in $\mathbb{Z}/N\mathbb{Z}$, to recover r. The* RSA assumption *states that, for all probabilistic polynomial time algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$, defined by $\text{Adv}^{RSA}(\mathcal{A}) = \Pr(\mathcal{A}(N, e, r^e) = r)$, is negligible in λ.*

Remark. We have seen in the questions above that the RSA problem can be solved efficiently if we know (p,q) or $\varphi(N)$. The RSA assumption states that if we only know N, then the problem is difficult to solve. This implies, in particular, that it is hard to recover (p,q) from N, but this is a possibly stronger assumption than simply assuming that factoring is hard (it may be possible to solve the RSA problem without factoring N).

2 An IND-CPA encryption scheme from the RSA assumption

We define the *Textbook-RSA* encryption scheme as follows.

- KeyGen. Choose p, q and e as in the RSA problem. Let $d \in \mathbb{Z}_{\varphi(N)}$ such that $ed = 1 \bmod \varphi(N)$. The public key is $pk = (N, e)$, while the secret key is $sk = d$.
- Enc. For $m \in \mathbb{Z}/N\mathbb{Z}$, compute and return $c = m^e \bmod N$.
- Dec. For $c \in \mathbb{Z}/N\mathbb{Z}$, compute and return $m = c^d \bmod N$.

4. Show that the Textbook-RSA encryption scheme is not IND-CPA secure.

[1] The RSA assumption is known not to hold if d is small, or if p and q are too close. We will not detail here how p, q, e are sampled and we will just assume that they are drown from some distributions D_1 and D_2 for which the RSA assumption holds.

We define a better public key encryption scheme, which we will refer to as *IND-CPA RSA encryption scheme*.

- KeyGen. Choose p, q and e as in the RSA problem. Let $d \in \mathbb{Z}_{\varphi(N)}$ such that $ed = 1 \bmod \varphi(N)$. Let $H : \mathbb{Z}/N\mathbb{Z} \to \{0,1\}^\ell$ (with $\ell \geq \lambda$) be a hash function which will be modelled as a random oracle in the security proof. The public key is $pk = (N, e)$, while the secret key is $sk = d$.
- Enc. For $m \in \{0,1\}^\ell$, sample r uniformly in $\mathbb{Z}/N\mathbb{Z}$ and return $c = (r^e \bmod N, H(r) \oplus m)$.
- Dec. For $c = (c_1, c_2) \in \mathbb{Z}/N\mathbb{Z} \times \{0,1\}^\ell$, compute $r = c_1^d \bmod N$ and return $m = H(r) \oplus c_2$.

We want to show that this scheme is IND-CPA secure in the random oracle model, if the RSA assumption holds.

5. Recall the IND-CPA security game in the random oracle model.

Let b be the bit chosen by the challenger (i.e., b is sampled uniformly in $\{0,1\}$ and the challenger encrypts the message m_b). Let $\mathcal{A}$ be an adversary against the IND-CPA RSA encryption scheme (in the random oracle model). We denote by `Query` the event "$\mathcal{A}$ makes a random oracle query on the randomness r that was used by the challenger to encrypt the challenge message m_b".

6. Show that $|\Pr(\mathcal{A} \to 1 \text{ and not } \texttt{Query} | b = 0) - \Pr(\mathcal{A} \to 1 \text{ and not } \texttt{Query} | b = 1)| = 0$ (in the random oracle model).

7. Show that $\Pr(\texttt{Query})$ is negligible if the RSA assumption holds (in the random oracle model).

8. Conclude that the IND-CPA RSA encryption scheme is IND-CPA secure in the random oracle model, if the RSA assumption holds. In particular, give an upper-bound on $Adv(\mathcal{A})$ as a function of $\max_{\text{PPT } \mathcal{B}}(Adv^{RSA}(\mathcal{B}))$.

3 A digital signature from the RSA assumption

9. Recall the definition of EU-CMA (existential unforgeability against chosen message attack) security for a public key signature scheme.

We define the simple signature RSA scheme as follows

- KeyGen. Choose p, q and e as in the RSA problem. Let $d \in \mathbb{Z}_{\varphi(N)}$ such that $ed = 1 \bmod \varphi(N)$. The public key is $pk = (N, e)$, while the secret key is $sk = d$.
- Sign(m, d). For $m \in \mathbb{Z}/N\mathbb{Z}$, return $s = m^d \bmod N$.
- Verify(m, s, e). For $s \in \mathbb{Z}/N\mathbb{Z}$, return 1 if $s^e = m \bmod N$ and 0 otherwise.

10. Show that the simple signature scheme above is not EU-CMA secure.

Let us define a better signature scheme, called *Full-domain hash RSA signature scheme* (we will prove its security in the random oracle model).

- KeyGen. Choose p, q and e as in the RSA problem. Let $d \in \mathbb{Z}_{\varphi(N)}$ such that $ed = 1 \bmod \varphi(N)$. Let $H : \{0,1\}^* \to \mathbb{Z}/N\mathbb{Z}$ be a hash function which will be modelled as a random oracle in the security proof. The public key is $pk = (N, e)$, while the secret key is $sk = d$.
- Sign(m, d). For $m \in \{0,1\}^*$, output $s = H(m)^d \bmod N$.
- Verify(m, s, e). For $s \in \mathbb{Z}/N\mathbb{Z}$, outputs 1 if $s^e = H(m) \bmod N$ and 0 otherwise.

We want to show that this scheme is EU-CMA secure in the random oracle model, if the RSA assumption holds.

Let $\mathcal{A}$ be an adversary against the Full-domain hash RSA signature scheme (in the random oracle model). We denote by `Query` the event "$\mathcal{A}$ makes a random oracle query for the m^* for which it forges a signature (m^*, s^*)" (equivalently, `not Query` is the event "$\mathcal{A}$ forges a signature (m^*, s^*) without making a random oracle query to get $H(m^*)$").

11. Show that $\Pr(\mathcal{A}$ forges a valid signature and not $\texttt{Query})$ is negligible (in the random oracle model).

12. Show that $\Pr(\mathcal{A}$ forges a valid signature and $\texttt{Query})$ is negligible (in the random oracle model) if the RSA assumption holds.

Hint : you may want to introduce Q the number of (different) random oracle queries made by the attacker and guess on which one it queries the x^* for which it will forge a signature.

13. Conclude that the Full-domain hash signature scheme is EU-CMA secure in the random oracle model, if the RSA assumption holds. In particular, give an upper-bound on $\mathrm{A}dv(\mathcal{A})$ as a function of $\max_{\mathrm{PPT}\ \mathcal{B}}(\mathrm{A}dv^{RSA}(\mathcal{B}))$.

4 A CCA2 encryption scheme from the RSA assumption

14. Recall the definition of CCA2 security for a public key encryption scheme.

We define the following (public key) encryption scheme, where $(KeyGen', Enc', Dec')$ is a symmetric encryption scheme which is CCA2 secure (with the key space being $\{0,1\}^s$, the message space being $\{0,1\}^\ell$ and the ciphertext space being $\{0,1\}^m$).

- KeyGen. Choose p, q and e as in the RSA problem. Let $d \in \mathbb{Z}_{\varphi(N)}$ such that $ed = 1 \bmod \varphi(N)$. Let $H : \mathbb{Z}/N\mathbb{Z} \to \{0,1\}^s$ be a hash function which will be modelled as a random oracle in the security proof. The public key is $pk = (N, e)$, while the secret key is $sk = d$.
- Enc. For $m \in \{0,1\}^\ell$, sample r uniformly in $\mathbb{Z}/N\mathbb{Z}$ and return $c = (r^e \bmod N, Enc'_{H(r)}(m))$.
- Dec. For $c = (c_1, c_2) \in \mathbb{Z}/N\mathbb{Z} \times \{0,1\}^m$, compute $r = c_1^d \bmod N$ and return $m = Dec'_{H(r)}(c_2)$.

We want to show that this scheme is CCA2 secure in the random oracle model, if the RSA assumption holds and if $(KeyGen', Enc', Dec')$ is a secure CCA2 symmetric encryption scheme.
Let b be the bit chosen by the challenger (i.e., b is sampled uniformly in $\{0,1\}$ and the challenger encrypts the message m_b). Let $\mathcal{A}$ be an adversary against the CCA2 security of $(KeyGen, Enc, Dec)$, in the random oracle model. We denote by $\texttt{Query}$ the event "$\mathcal{A}$ makes a random oracle query on the randomness r that was used by the challenger to encrypt the challenge message m_b".

15. Show that $\Pr(\texttt{Query})$ is negligible if the RSA assumption holds (in the random oracle model).

Hint: you may want to maintain a list of triples $(r, r^e, H(r))$, which may contain partially known triples of the form $(\star, r^e, H(r))$.

16. Show that $|\Pr(\mathcal{A} \to 1 \text{ and not } \texttt{Query} | b = 0) - \Pr(\mathcal{A} \to 1 \text{ and not } \texttt{Query} | b = 1)|$ is negligible if $(KeyGen', Enc', Dec')$ is a secure CCA2 secret key encryption scheme (in the random oracle model).

17. Conclude that $(KeyGen, Enc, Dec)$ is a CCA2 secure encryption scheme in the random oracle model, if the RSA assumption holds and if $(KeyGen', Enc', Dec')$ is a secure symmetric CCA2 encryption scheme. In particular, give an upper-bound on $\mathrm{A}dv(\mathcal{A})$ as a function of $\max_{\mathrm{PPT}\ \mathcal{B}}(\mathrm{A}dv^{RSA}(\mathcal{B}))$ and $\max_{\mathrm{PPT}\ \mathcal{B}'}(\mathrm{A}dv^{Enc'}(\mathcal{B}'))$, where $\mathrm{A}dv^{Enc'}(\mathcal{B}')$ is the advantage of $\mathcal{B}'$ against the CCA2 security of $(KeyGen', Enc', Dec')$.