
Homework 7

Due: Thursday, May 26 at 12PM (noon).

This complements the PlayCrypt version of this problem set. You need turn in only the latter, on Gradescope. This version is being given out so that you can see what the problems look like in mathematical notation.

We suggest that you start with this version. Work out a solution using pencil and paper. Move to implementation in PlayCrypt only after that.

As usual our convention is that the running time of an adversary does not include the time taken by game procedures to compute responses to adversary queries.

Problem 1 [20 points] Let $\mathcal{K}_{\text{rsa}}$ be an RSA generator with security parameter k, and let ℓ be an additional parameter. We require that k and ℓ are both multiples of 8, and that $\ell \le k-8$. Consider the key-generation algorithm $\mathcal{K}$ and encryption algorithm $\mathcal{E}$ defined below:

Alg $\mathcal{K}$ $(N,p,q,e,d) \xleftarrow{\$} \mathcal{K}_{\text{rsa}}$ Return $((N,e),(N,d,p,q))$	**Alg** $\mathcal{E}((N,e),M)$ // $M \in \{0,1\}^{k-\ell-8}$ $R \xleftarrow{\$} \{0,1\}^{\ell/2}$ $Z \leftarrow 0^{\ell/2}$ $X \leftarrow (M \,\|\, R \,\|\, Z)$ $m \leftarrow \mathbf{int}(X)$ $c \leftarrow m^e \bmod N$ Return c

The notation $\mathbf{int}(X)$ indicates that we are converting the binary string X to an integer. In particular, we would like to ensure that $m \in Z_N^*$, when N is a k-bit integer. Note that we could also compute m by $m \leftarrow \mathbf{int}(M)\cdot 2^{\ell} + \mathbf{int}(R)\cdot 2^{\ell/2} + 0$.

1. [**8 points**] Specify in pseudocode an $\mathcal{O}(k^3)$-time decryption algorithm $\mathcal{D}$ such that $\mathcal{AE} = (\mathcal{K},\mathcal{E},\mathcal{D})$ is an asymmetric encryption scheme satisfying the correct decryption requirement.

   Note that your algorithm $\mathcal{D}$ must return $\perp$ (`None` in Python) if the input ciphertext could not have been produced by the $\mathcal{E}$ algorithm.

2. [**12 points**] Show that $\mathcal{AE} = (\mathcal{K},\mathcal{E},\mathcal{D})$ is not IND-CCA-secure by presenting an $\mathcal{O}(k^3)$-time adversary A making one **LR** query, one **Dec** query, and achieving $\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{AE}}(A) \approx 1$.

Optional. Here are a few questions that may help you think about these problems. You do not need to include answers in your code.

- What modification to the above scheme would make it trivially not IND-CPA-secure?
- What is the maximum bit length of $(M \parallel R \parallel Z)$? Why does (or doesn't) this guarantee that $m \in Z_N^*$?
