CS 359C: Classics of Cryptography **Spring 2017**

Problem Set 1

Due: April 26, 2017, by 2:30pm (submit hard copy at the *beginning* of lecture)

Instructions: You must typeset your solution in LaTeX using the provided template:

`https://web.stanford.edu/class/cs359c/homework.tex`

Problem 1: Even-Mansour (5 points). The Even-Mansour cipher $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ we saw in class uses a public random permutation $\Pi : \{0,1\}^n \to \{0,1\}^n$ and is defined as:

$$E(k, m) \stackrel{\text{def}}{=} \Pi(k \oplus m) \oplus k.$$

The original Even-Mansour paper used a slightly different construction, with two independent keys $k_1, k_2 \in \{0,1\}^n$:

$$\hat{E}((k_1, k_2), m) \stackrel{\text{def}}{=} \Pi(k_1 \oplus m) \oplus k_2.$$

Prove that if E is a secure PRP, then $\hat{E}$ is also.

Problem 2: RSA with a Common Modulus (5 points). Suppose you have two RSA keys (N, e_1) and (N, e_2) that share a common *modulus* N such that e_1 and e_2 are relatively prime. Consider the following candidate PRG construction $G : \mathbb{Z}_N \to \mathbb{Z}_N \times \mathbb{Z}_N$ where $G(x) = (x^{e_1}, x^{e_2}) \in \mathbb{Z}_N \times \mathbb{Z}_N$. Show that this is not a secure PRG.

Problem 3: RSA Watermarking (10 points). Suppose you wanted to embed a short string σ in your RSA modulus N. Given your modulus N, anyone should be able to recover the string σ without knowledge of the factors of N. More formally, for this question, you must produce a pair of algorithms:

- $\mathsf{Hide}(1^\lambda, \sigma) \to N$. This algorithm takes as input a security parameter λ and an $O(\log \lambda)$-bit string σ, and outputs an RSA modulus N composed of two λ-bit primes.
- $\mathsf{Extract}(N) \to \sigma$. This algorithm takes as input an RSA modulus produced by the Hide algorithm and outputs the string σ embedded in the modulus.

(a) Produce efficient algorithms Hide and Extract, *prove* that for all σ, $\mathsf{Extract}(\mathsf{Hide}(1^\lambda, \sigma)) = \sigma$, and explain why your algorithms run in polynomial time.

(b) Show that if there exists an algorithm for factoring a watermarked RSA modulus that runs in time t and succeeds with probability ϵ, then there exists an algorithm for factoring a standard RSA modulus that runs in time t' and succeeds with probability ϵ'. You should have that $t' = \mathsf{poly}(\lambda) \cdot t$ and $\epsilon' = \epsilon / \mathsf{poly}(\lambda)$. In other words, a watermarked RSA modulus is roughly as hard to factor as a standard RSA modulus.

(c) What goes wrong with Part (b) if σ is of length $\Omega(\lambda)$ bits?

Problem 4: Fancy Meet-in-the-Middle (10 points). Let $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Consider the block cipher $E^4 : \{0,1\}^{4n} \times \{0,1\}^n \to \{0,1\}^n$, which invokes E four times in serial using independent keys:

$$E^4((k_1, k_2, k_3, k_4), m) \stackrel{\text{def}}{=} E(k_4, E(k_3, E(k_2, E(k_1, m)))).$$

(a) Show that there is a key-recovery attack on E^4 that takes time $O(2^{2n})$ and space $\tilde{O}(2^{2n})$.

(b) [More difficult.] Show that there is a key-recovery attack on E^4 that takes time $O(2^{2n})$ and space $\tilde{O}(2^n)$.