# Introduction to Cryptography and Software Security
# Problem Set 1: Private-Key Encryption

Due date: 14/4/2019

**Instructions.**

- You are allowed to rely on any statement that we proved in class, unless you are explicitly asked to prove it, and as long as you state it clearly and accurately.
- Justify your answers with formal proofs.
- **On-line submissions only! Hard copies will not be accepted.** Please make sure that scanned submissions are readable (unreadable submissions will not be accepted).

**Problem 1 (10%).** Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric-key encryption scheme. Prove that $\Pi$ is *perfectly secret* if and only if for any message distribution $M$ over $\mathcal{M}$, for any message $m \in \mathcal{M}$ such that $\Pr[M = m] > 0$, and for any ciphertext $c \in \mathcal{C}$ it holds that

$$\Pr[C = c | M = m] = \Pr[C = c] .$$

**Problem 2 (10%).** Show that the definition of a pseudorandom generator cannot be satisfied with respect to distinguishers that are *not restricted to polynomial running time*. Specifically, prove that for any pseudorandom generator $G$ with expansion $\ell(n)$ there exists a (non-polynomial-time) distinguisher $D$ such that

$$\left| \Pr_{s \leftarrow \{0,1\}^n} [D(G(s)) = 1] - \Pr_{r \leftarrow \{0,1\}^{\ell(n)}} [D(r) = 1] \right| \geq \frac{1}{2}$$

for every $n \in \mathbb{N}$.

**Problem 3 (10%).** Let $F$ be a pseudorandom function, and consider the function $G : \{0,1\}^* \to \{0,1\}^*$ defined as

$$G(s) = F_s(1) || F_s(2) || \cdots || F_s(n+1)$$

where $n = |s|$ (and we naturally view the integers $1, \ldots, n+1$ as elements of $\{0,1\}^*$ via their binary representation). Prove that $G$ is a pseudorandom generator.

**Problem 4 (20%).** Let $F$ be a pseudorandom function such that for any $n \in \mathbb{N}$ and key $k \in \{0,1\}^n$ it holds that $F_k : \{0,1\}^n \to \{0,1\}^{2n}$.

1. Let $H_k(x) = F_k(x) \oplus F_{1^n}(x)$ for any $n \in \mathbb{N}$, $k \in \{0,1\}^n$ and $x \in \{0,1\}^n$. Is $H$ necessarily a pseudorandom function? Prove your answer.
2. Let $G(s) = F_{0^n}(s)$ for any $n \in \mathbb{N}$ and $s \in \{0,1\}^n$. Is $G$ necessarily a pseudorandom generator? Prove your answer.

**Problem 5 (20%).** Let $F$ be a pseudorandom function and let $G$ be a pseudorandom generator with expansion $\ell(n) = n + 1$. For each of the following candidate encryption schemes state whether it is an IND-secure scheme or a CPA-secure scheme (and formally justify your answers). In all cases the encryption key is a uniformly sampled $k \in \{0,1\}^n$.

1. To encrypt a message $m \in \{0,1\}^{n+1}$, sample $r \leftarrow \{0,1\}^n$, and output the pair $(r, G(r) \oplus m)$.
2. To encrypt a message $m \in \{0,1\}^n$ output $F_k(0^n) \oplus m$.
3. To encrypt a message $m \in \{0,1\}^n$, sample $r \leftarrow \{0,1\}^n$, and output the pair $(r, r \oplus F_k(r) \oplus m)$.

**Problem 6 (10%).** Recall that in class we defined the notion of a pseudorandom function with respect to keys that are *uniformly distributed*. More generally, it is possible to consider keys that are not necessarily uniformly distributed (as we did, for example, when defining the notion of a private-key encryption scheme).

1. Generalize the formal definition of a pseudorandom function presented in class to consider keys that are produced using a key-generation algorithm $\mathsf{KeyGen}$ that on input $1^n$ outputs keys of length $\ell(n)$ bits for every $n \in \mathbb{N}$.

   [Note: Your generalized definition should capture a pseudorandom function as a pair $(\mathsf{KeyGen}, F)$, where $\mathsf{KeyGen}$ is the key-generation algorithm and $F$ is the evaluation algorithm.]

2. Assuming the existence of a pseudorandom function $(\mathsf{KeyGen}, F)$, construct a pseudorandom function $(\mathsf{KeyGen}', F')$ where the algorithm $\mathsf{KeyGen}'$ produces uniformly distributed keys.

   [Note: The length $\ell'(n)$ of the keys produced by $\mathsf{KeyGen}'(1^n)$ does not have to be equal to the length $\ell(n)$ of the keys produced by $\mathsf{KeyGen}(1^n)$.]

**Problem 7 (20%).** Let $\mathsf{KeyGen}$ and $F$ be computable in polynomial time such that for any $n \in \mathbb{N}$ and for any key $k$ that is produced by $\mathsf{KeyGen}(1^n)$ it holds that $F_k : \{0,1\}^n \to \{0,1\}^n$. We say that $(\mathsf{KeyGen}, F)$ is a *leftmost-bit leakage-resilient pseudorandom function* if for any probabilistic polynomial-time algorithm $D$ and for any polynomial $p(\cdot)$ it holds that

$$\left| \Pr_{k \leftarrow \mathsf{KeyGen}(1^n)} \left[ D^{F_k(\cdot)}(1^n, k_1) = 1 \right] - \Pr_{\substack{k \leftarrow \mathsf{KeyGen}(1^n) \\ f \leftarrow \mathsf{Func}_{n,n}}} \left[ D^{f(\cdot)}(1^n, k_1) = 1 \right] \right| \leq \frac{1}{p(n)}$$

for all sufficiently large $n \in \mathbb{N}$, where $k_1$ is the leftmost bit of the key $k$.

1. Assuming the existence of any pseudorandom function $(\mathsf{KeyGen}, F)$, construct a pseudorandom function $(\mathsf{KeyGen}', F')$ that is leftmost-bit leakage resilient.

2. Assuming the existence of any pseudorandom function $(\mathsf{KeyGen}, F)$, construct a pseudorandom function $(\mathsf{KeyGen}', F')$ that is **not** leftmost-bit leakage resilient.