ECS 227 — Modern Cryptography — Winter 2012

Phillip Rogaway

Out: Tuesday, 21 February 2012.
Due: Thursday, 1 March 2012.

1. A *nonce-based symmetric encryption scheme* is a three-tuple of algorithms $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ that is like the encryption schemes we have defined before except that $\mathcal{E}$ is now deterministic and stateless (as is $\mathcal{D}$), and $\mathcal{E}$ and $\mathcal{D}$ now take in an additional argument $N \in \mathcal{N} \subseteq \{0,1\}^*$, the *nonce*. When encrypting, a party is required to select a new nonce N to go with each message that is encrypted. As long as he does this, privacy should be assured. The nonce could be a counter, for example, or a long enough random string.

(a) Carefully formalize a notion of ind\$-security for a nonce-based symmetric encryption scheme.

(b) Describe a blockcipher-based scheme Π that achieves your notion of security from (a), assuming that the blockcipher $E\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ from which Π is defined is secure as a PRP.

(c) Do you see any advantages of the nonce-based notion? Any disadvantages? Briefly discuss.

2. Suppose there exists a public-key encryption scheme that is IND-CPA secure. Show that there is a public-key encryption scheme that is IND-CPA secure but that is not IND-CCA secure.

3. Suppose you have a fast deterministic algorithm I that inverts $f(x) = x^e \bmod N$ on 1% of all inputs—the inputs in $\mathbb{Z}_N^*$ that your algorithm likes. Construct a usually-fast probabilistic algorithm J that inverts $f(x) = x^e \bmod N$ on every point in $\mathbb{Z}_N^*$. Analyze the efficiency of your algorithm: what is the expected running time of J? Your algorithm should be of the "Las Vegas" variety: it is always correct, and on every input it is usually fast. Analyze the efficiency of your algorithm.