Notes for Lecture 23 (draft)

Summary

Today we show how to construct an efficient CCA-secure public-key encryption scheme in the *random oracle model* using RSA.

As we discussed in the previous lecture, a cryptographic scheme defined in the *random oracle model* is allowed to use a random function $H : \{0,1\}^n \rightarrow \{0,1\}^m$ which is known to all the parties. In an implementation, usually a cryptographic hash function replaces the random oracle. In general, the fact that a scheme is proved secure in the random oracle model does not imply that it is secure when the random oracle is replaced by a hash function; the proof of security in the random oracle model gives, however, at least some heuristic confidence in the soundness of the design.

1 Hybrid Encryption with a Random Oracle

We describe a public-key encryption scheme $(\overline{G}, \overline{E}, \overline{D})$ which is based on: (1) a family of trapdoor permutations (for concreteness, we shall refer specifically to RSA below); (2) a CCA-secure *private-key* encryption scheme (E, D); (3) a random oracle H mapping elements in the domain and range of the trapdoor permutation into keys for the private-key encryption scheme (E, D).

1. Key generation: $\overline{G}$ picks an RSA public-key / private-key pair $(N, e), (N, d)$;
2. Encryption: given a public key N, e and a plaintext M, $\overline{E}$ picks at random $R \in \mathbb{Z}_N$, and outputs
$$R^d \bmod N, E(H(R), M)$$
3. Decryption: given a private key N, d and a cyphertext C_1, C_2, $\overline{D}$ decrypts the plaintext by computing $R := C_1^d \bmod N$ and $M := D(H(R), C_2)$.

This is a hybrid encryption scheme in which RSA is used to encrypt a "session key" which is then used to encrypt the plaintext via a private-key scheme. The important difference from hybrid schemes we discussed before is that the random string encrypted with RSA is "hashed" with the random oracle before being used as a session key.

2 Security Analysis

Theorem 1 *Suppose that, for the key size used in $(\overline{G}, \overline{E}, \overline{D})$, RSA is a (t, ϵ)-secure family of trapdoor permutations, and that exponentiation can be computed in time $\leq r$; assume also that (E, D) is a (t, ϵ) CCA-secure private-key encryption scheme and that E, D can be computed in time $\leq r$.*

Then $(\overline{G}, \overline{E}, \overline{D})$ is $(t/O(r), 2\epsilon)$ CCA-secure in the random oracle model.

We sketch an outline of the proof. We assume that there is an algorithm A showing that $(\overline{G}, \overline{E}, \overline{D})$ is not (t', ϵ') CCA-secure. From A we derive an algorithm A' running in time $O(t' \cdot r)$ which is a CCA attack on (E, D). If A' succeeds with probability at least ϵ as a distinguishing CCA attack against (E, D), then we have violated the assumption on the security of (E, D). But if A' succeeds with probability less than ϵ, then we can devise an algorithm A'', also of running time $O(t' \cdot r)$, which inverts RSA with probability at least ϵ.