Notes for Lecture 18

Scribed by Steve Hanna, posted April 30, 2009

Summary

Today we discuss the three ways in which definitions of security given in class differ from the way they are given in the Katz-Lindell textbook,.

Then we study the security of *hybrid* encryption schemes, in which a public-key scheme is used to encode the key for a private-key scheme, and the private-key scheme is used to encode the plaintext.

We also define RSA and note that in order to turn RSA into an encryption scheme we need a mechanism to introduce randomness.

1 Definitions of Security

There are three ways in which the definitions of security given in class differ from the way they are given in the textbook. The first one applies to all definitions, the second to definitions of encryption, and the third to CPA and CCA notions of security for encryption:

1. Our definitions usually refer to schemes of fixed key length and involve parameters t, ϵ, while the textbook definitions are asymptotic and parameter-free.

   Generally, one obtains the textbook definition by considering a family of constructions with arbitrary key length (or, more abstractly "security parameter") k, and allowing t to grow like any polynomial in k and requiring ϵ to be negligible. (Recall that a non-negative function $\nu(k)$ is negligible if for every polynomial p we have $\lim_{k\to\infty} p(k) \cdot \nu(k) = 0$.)

   The advantage of the asymptotic definitions is that they are more compact and make it easier to state the result of a security analysis. (Compare "if one-way permutations exist, then length-increasing pseudorandom generators exist" with "if $f : \{0,1\}^n \to \{0,1\}^n$ is a (t, ϵ) one-way permutation computable in time $\leq r$, then there is a generator $G : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ computable in time $r + O(n)$ that is $(t\epsilon^4/(n^2 \log n) - r\epsilon^{-4}n^3 \log n, \epsilon/3)$ pseudorandom")

The advantage of the parametric definitions is that they make sense for fixed-key constructions, and that they make security proofs a bit shorter.

(Every asymptotic security proof starts from "There is a polynomial time adversary A, an infinite set N of input lengths, and a polynomial $q()$, such that for every $n \in N \ldots$.)

2. Definitions of security for an encryption algorithm $E()$, after the proper quantifications, involve an adversary A (who possibly has oracles, etc.) and messages m_0, m_1; we require

$$|\,\mathbb{P}[A(E(m_0)) = 1] - \mathbb{P}[A(E(m_1)) = 1]| \leq \epsilon \tag{1}$$

while the textbook usually has a condition of the form

$$\mathop{\mathbb{P}}_{b\in\{0,1\}}[A(E(m_b)) = b] \leq \frac{1}{2} + \frac{\epsilon}{2}$$

The two conditions are equivalent since

$$\mathbb{P}[A(E(m_b)) = b] = \frac{1}{2}\,\mathbb{P}[A(E(m_0)) = 0] + \frac{1}{2}\,\mathbb{P}[A(E(m_1)) = 1$$

and the absolute value in (1) may be removed without loss of generality (at the cost of increasing the complextiy parameter by one).

3. Definitions of CPA and CCA security, as well as all definitions in the public-key setting, have a different structure in the book. The difference is best explained by an example. Suppose we have a public-key scheme (G, E, D) such that, for every valid public key pk, $E(pk, pk)$ has some distinctive pattern that makes it easy to distinguish it from other ciphertexts. This could be considered a security weakness because an eavesdropper is able to see if a party is sending a message that concerns the public key.

   This would not be a concern if the encryption mechanism were separate from the transport mechanism. For instance, if the application of this scheme occurred in such a way they two parties are securely communicating over an instant messaging client which exists in the application layer and encryption were occurring layers below in the transport layer. This abstraction of layers and separation of the encryption mechanism from the application abstracts away the notion that the public key could be encrypted with the public key. The messaging client is aware of the interface, but it never exposes the actual public or private key to the user, which prevents incorrectly using the cryptographic primitives.

You can show as an exercise that if a secure public-key encryption scheme exists, then there is a public-key encryption scheme that is secure according to our definition from last lecture but that has a fault of the above kind.

The textbook adopts a two-phase definition of security, in which the adversary is allowed to choose the two messages m_0, m_1 that it is going to try and distinguish, and the choice is done *after* having seen the public key. A random bit b is chosen and then the ciphertext of m_b is computed and given to the adversary. The adversary continues to have access to the Encryption function with the given public key. When the adversary is done, it outputs a guess called b' The output of this procedure is 1 when $b = b'$. A cryptosystem in which $E(pk, pk)$ can be distinguished from other ciphertexts violates this definition of security.

2 Hybrid Encryption

Let (G_1, E_1, D_1) be a public-key encryption scheme and (E_2, D_2) a private-key encryption scheme.

Consider the following *hybrid* scheme (G, E, D):

- $G()$: same as $G_1()$
- $E(pk, m)$: pick a random key K for E_2, output $(E_1(pk, K), E_2(K, m))$
- $D(sk, (C_1, C_2))$: output $D_2(D_1(sk, C_1), C_2)$

A hybrid approach to public key cryptography is often desired due to the fact that public key operations are computationally expensive (i.e modular exponentiation), while symmetric key cryptosystem are usually more efficient. The basic idea behind the hybrid approach that if we encrypt the symmetric private key with the public key and encrypt the message with the symmetric private key, only the small symmetric private key needs to be encrypted with the public key and symmetric key encryption/decryption can take place on the actual message. This allows for efficient computation of the message encryption and decryption while only using asymmetric key cryptography for transmitting the symmetric shared secret. This construction makes encryption and decryption much more efficient while still ensuring the construction has message indistinguishability and CPA security.

Theorem 1 *Suppose (G_1, E_1, D_1) is (t, ϵ_1)-secure for one encryption and (E_2, D_2) is (t, ϵ_2)-secure for one encryption. Suppse also that E_1, E_2 have running time $\leq r$.*

Then (G, E, D) is $(t - 2r, 2\epsilon_1 + \epsilon_2)$-secure for one encryption.

We begin by assuming the conclusion of the theorem is false, that is (G, E, D) is not $(t - 2r, 2\epsilon_1 + \epsilon_2)$-secure.

Suppose there is an adversary A, that runs in time t' and there are two messages m_0 and m_1 such that:

$$|\,\mathbb{P}[A(pk, E(pk, m_0)) = 1] - \mathbb{P}[A(pk, E(pk, m_1)) = 1]| > \ 2\epsilon_1 + \epsilon_2$$

Then the definition of $E()$ is applied, so

$$|\,\mathbb{P}[A(pk, E_1(pk, K), E_2(K, m_0)) = 1] - \mathbb{P}[A(pk, E_1(pk, K), E_2(K, m_1) = 1]| > 2\epsilon_1 + \epsilon_2$$

We then apply a hybrid argument in which the hybrid distributions have $E_1(pk, \mathbf{0})$ instead of $E_1(pk, K)$. ($\mathbf{0}$ denotes a string of zeroes; any other fixed string could be used in the proof.)

Producing:

$$2\epsilon_1 + \epsilon_2 < \mathbb{P}[A(pk, E_1(pk, K), E_2(K, m_0)) = 1] - \mathbb{P}[A(pk, E_1(pk, K), E_2(K, m_1) = 1] \leq$$

$$\begin{aligned}
&\|\,\mathbb{P}[A(pk, E_1(pk, K), E_2(K, m_0)) = 1] - \mathbb{P}[A(pk, E_1(pk, \mathbf{0}), E_2(K, m_0)) = 1]\| + \\
&\|\,\mathbb{P}[A(pk, E_1(pk, \mathbf{0}), E_2(K, m_0)) = 1] - \mathbb{P}[A(pk, E_1(pk, \mathbf{0}), E2(K, m_1)) = 1]\| + \\
&\|\,\mathbb{P}[A(pk, E_1(pk, \mathbf{0}), E2(K, m_1)) = 1] - \mathbb{P}[A(pk, E_1(pk, K), E2(K, m_1)) = 1]\|
\end{aligned}$$

This means that at least one of the following cases must happen:

a) the first difference is at least ϵ_1

b) the second difference is at least ϵ_2

c) the third difference is at least ϵ_1

If (a) or (c) are true, then it means that there is a message m such that:

$$\|\,\mathbb{P}[A(pk, E_1(pk, K), E_2(K, m)) = 1] - \mathbb{P}[A(pk, E_1(pk, 0), E_2(K, m)) = 1]\| > \epsilon_1$$

Then there must exist one fixed K^* such that

$$\|\,\mathbb{P}[A(pk, E_1(pk, K^*), E_2(K, m)) = 1] - \mathbb{P}[A(pk, E_1(pk, \hat{0}), E2(K, m)) = 1]\| > \epsilon_1$$

and then we define an algorithm A' of complexity at most t such that:

$$\| \mathbb{P}[A'(pk, E_1(pk, K^*)) = 1] - \mathbb{P}[A'(pk, E_1(pk, \hat{0})) = 1] \| > \epsilon_1$$

which contradicts the security of E_1.
If (b) is true, then we define an algorithm A'' of complexity at most t such that:

$$\| \mathbb{P}[A''(E_2(K, m_0)) = 1] - \mathbb{P}[A''(E_2(K, m_1)) = 1] \| > \epsilon_2$$

which contradicts the security of E_2.

3 RSA

The RSA function has the same "syntax" of a public-key encryption scheme:

- Key generation: Pick two distinct prime numbers p, q, compute $N := pq$, and find integers e, d such that

  $$e \cdot d \equiv 1 \pmod{()p-1) \cdot (q-1}$$

  Set the public key to (N, e) and the private key to (N, d)
- "Encryption:" given $x \in \mathbb{Z}_N$ and public key (N, e), output

  $$E_{RSA}(x, (N, e)) := x^e \bmod N$$
- "Decryption:" given $y \in \mathbb{Z}_N$ and secret key (N, d), output

  $$D_{RSA}(y, (N, d)) := y^d \bmod N$$

It is a standard calculation using the Chinese remainder theorem and Fermat's little theorem that $E_{RSA}(\cdot, (N, e))$ and $D_{RSA}(\cdot, (N, d))$ are permutations over $\mathbb{Z}_N$, and they are one the inverse of the other.

This is, however, not a secure encryption scheme because it is *deterministic*, and it suffers from several weaknesses that can exploited in practice.

A conjectural way to turn RSA into a CPA-secure encryption scheme is to employ it to encrypt plaintexts whose length is only about $\frac{1}{2} \log N$, and then pad the plaintext with $\frac{1}{2} \log N$ random bits before applying E_{RSA}. (Other choices for the length of the plaintext and the amount of randomness are possible, half-half is just an example.)

The assumption that this padded-RSA is CPA secure is a very strong one. In the next lecture we will see how to turn RSA into a CPA secure encryption scheme under the minimal assumption that E_{RSA} is hard to invert on a random input for an adversary that knows the public key but not the secret key.