Notes for Lecture 17

Summary

Today we begin to talk about public-key cryptography, starting from public-key encryption.

We define the public-key analog of the weakest form of security we studied in the private-key setting: message-indistinguishability for one encryption. Because of the public-key setting, in which everybody, including the adversary, has the ability to encrypt messages, this is already equivalent to CPA security.

We then describe the El Gamal cryptosystem, which is message-indistinguishable (and hence CPA-secure) under the plausible *Decision Diffie-Hellman* assumption.

1 Public-Key Cryptography

So far, we have studied the setting in which two parties, Alice and Bob, share a secret key K and use it to communicate securely over an unreliable channel. In many cases, it is not difficult for the two parties to create and share the secret key; for example, when we connect a laptop to a wireless router, we can enter the same password both into the router and into the laptop, and, before we begin to do online banking, our bank can send us a password in the physical mail, and so on.

In many other situations, however, the insecure channel is the only communication device available to the parties, so it is not possible to share a secret key in advance. A general problem of private-key cryptography is also that, in a large network, the number of required secret keys grows with the *square* of the size of the network.

In public-key cryptography, every party generates two keys: a secret key SK and a public key PK. The secret key is known only to the party who generated it, while the public key is known to everybody.

(For public-key cryptosystems to work, it is important that everybody is aware of, or has secure access to, everybody else's public key. A mechanism for the secure exchange of public keys is called a *Public Key Infrastructure* (PKI). In a network model in which adversaries are *passive*, meaning that they only eavesdrop on communication, the parties can just send each other's public keys over the network. In a network that has *active* adversaries, who can inject their own packets and drop other users'

packets, creating a public-key infrastructure is a very difficult problem, to which we may return when we talk about network protocols. For now we assume that either the adversary is passive or that a PKI is in place.)

As in the private-key setting, we will be concerned with two problems: *privacy*, that is the communication of data so that an eavesdropper can gain no information about it, and *authentication*, which guarantees to the recipient the identity of the sender. The first task is solved by public-key *encryption* and the second task is solved by *signature* schemes.

2 Public Key Encryption

A public-key encryption scheme is defined by three efficient algorithms (G, E, D) such that

- G takes no input and outputs a pair of keys (PK, SK)
- E, on input a public key PK and a plaintext message m outputs a ciphertext $E(PK, M)$.

  (Typically, E is a probabilistic procedure.)
- D, on input a secret key SK and ciphertext C, decodes C. We require that for every message m

$$\mathop{\mathbb{P}}_{\substack{(PK,SK)=G() \\ \text{randomness of } E}} [D(SK, E(PK, m)) = m] = 1$$

A basic definition of security is message-indistinguishability for one encryption.

Definition 1 *We say that a public-key encryption scheme (G, E, D) is (t, ϵ) message-indistinguishable if for every algorithm A of complexity $\leq t$ and for every two messages m_1, m_2,*

$$\left| \mathop{\mathbb{P}}_{\substack{(PK,SK)=G() \\ \textit{randomness of } E}} [A(PK, E(PK, m_1)) = 1] \right.$$

$$\left. - \mathop{\mathbb{P}}_{\substack{(PK,SK)=G() \\ \text{randomness of } E}} [A(PK, E(PK, m_2)) = 1] \right| \leq \epsilon$$

(From now on, we will not explicitly state the dependance of probabilities on the internal coin tosses of E, although it should always be assumed.)

Exercise 1 *Formalize the notion of CPA-security for public-key encryption. Show that if (G, E, D) is (t, ϵ) message indistinguishable, and $E(\cdot, \cdot)$ is computable in time $\leq r$, then (G, E, D) is also $(t/r, \epsilon)$ CPA-secure.*

3 The Decision Diffie-Hellman Assumption

Definition 2 (Decision Diffie-Hellman Assumption) *A distribution $\mathcal{D}$ over triples $(\mathbb{G}, g, q)$, where $\mathbb{G}$ is a cyclic group of q elements and g is a generator, satisfies the (t, ϵ) Decision Diffie-Hellman Assumption if for every algorithm A of complexity $\leq t$ we have*

$$\left| \mathop{\mathbb{P}}_{\substack{(\mathbb{G},g,q) \sim \mathcal{D} \\ x,y,z \sim \{0,\ldots,q-1\}}} [A(\mathbb{G}, g, q, g^x, g^y, g^z) = 1] \right.$$

$$\left. - \mathop{\mathbb{P}}_{\substack{(\mathbb{G},g,q) \sim \mathcal{D} \\ x,y \sim \{0,\ldots,q-1\}}} [A(\mathbb{G}, g, q, g^x, g^y, g^{xy}) = 1] \right| \leq \epsilon$$

Note that the El Gamal assumption may be plausibly satisfied even by a *fixed* group $\mathbb{G}$ and a fixed generator g.

4 El Gamal Encryption

The El Gamal encryption scheme works as follows. Let $\mathcal{D}$ be a distribution over $(\mathbb{G}, g, q)$ that satisfies the Decision Diffie-Hellman assumption:

- $G()$ samples $(\mathbb{G}, g, q)$, and picks a random number $x \in \{0, \ldots, q-1\}$.
  - $PK = (\mathbb{G}, g, q, g^x)$
  - $SK = (\mathbb{G}, g, q, x)$
- $E((\mathbb{G}, g, q, a), m)$:
  - pick at random $r \in \{0, \ldots, q-1\}$
  - output $(g^r, a^r \cdot m)$
- $D((\mathbb{G}, g, q, x), (c_1, c_2))$
  - Compute $b := c_1^x$
  - Find the multiplicative inverse b' of b
  - output $b' \cdot c_2$

Theorem 3 *Suppose $\mathcal{D}$ is a distribution that satisfies the (t, ϵ) Decision Diffie-Hellman assumption and that it is possible to perform multiplication in time $\leq r$ in the groups $\mathbb{G}$ occurring in $\mathcal{D}$.*

Then the El Gamal cryptosystem is $(t - r, \epsilon)$ message-indistinguishable.