**CSE 599b: Cryptography** **(Winter 2006)**

# Lecture 1: Introduction and Definitions

3 January 2006

*Lecturer: Paul Beame* *Scribe: Paul Pham*

## 1 "Secret Writing"

The word cryptography means "secret writing". In the original two-party setting of cryptography:

- The sender, usually called Alice, transmits a message.
- The receiver, usually called Bob, receives a message.

The channel by which the message is sent is generally insecure. Also interacting with the channel is an adversary, often simply an eavesdropper and hence called Eve.

- An *passive* adversary can only observe messages in the channel.
- An *active* adversary can modify messages in the channel but cannot sever it entirely (severing it would be detectable by Alice and Bob).

## 2 Desiderata

Desired properties of *secure* communication over this channel.

**secrecy** : Eve learns nothing from interacting with the channel that she didn't already know before.

**authenticity** : the message Bob receives is the same one Alice sent.

## 3 Other scenarios

- access control / passwords / general authentication
- data privacy
- electronic payments
- electronic voting
- bit commitment

- pseudorandom generators

Cryptography overlaps with both computational complexity and security but is not entirely circumscribed by either of them.

# 4 Secrets

Note that, so far, in the transmission problem above the receiver and a passive adversary have the same information. Similarly for authentication the sender and an active adversary have the same information.

- To get secrecy: receiver must know something (or have some capability) that adversary doesn't. Call this a secret or a *key*.
- To get authenticity: likewise, the sender must have a key that the adversary doesn't know.

# 5 Trust Models

**symmetric** (also called shared key or private key model): Alice and Bob share the same key.

**asymmetric** (also called public key): Each party has its own private key which is kept secret and an associated public key which is openly published.

# 6 Symmetric Encryption

Three spaces and associated distributions:

- key space $\mathcal{K}$
- message space $\mathcal{M}$
- ciphertext space $\mathcal{C}$

Three (polynomial time) algorithms:

- decryption algorithm $D : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$
- encryption algorithm $E : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$
- key generation algorithm $G$ (leave vague for now) which produces a key $K \in \mathcal{K}$

Remarks on distributions:

- Note that the distribution of $\mathcal{K}$ is determined by the key generation algorithm. It must be random, otherwise the adversary can predict it.
- The sender is free to choose a distribution on $\mathcal{M}$.
- $\mathcal{C}$ is determined by the encryption algorithm, $\mathcal{M}$, and $\mathcal{K}$.

$\log_2(|\mathcal{K}|) = k$ is known as the security parameter and typically determines the running time and level of security of the algorithms $E$ and $D$.

**Symmetric Encryption Protocol:**

Given that the key $K$ has been chosen:

- Sender computes $E(K, M) = E_K(M) = C$
- Receiver computes $D(K, C) = D_K(C) = M'$

For correctness of decryption we require that $M = D_K(E_K(M))$

The adversary learns the ciphertext $C$ and in general the approximate length of $M$, since the algorithms $E$ and $D$ are publicly known.

## 6.1 One-time pad

Example of symmetric encryption.

- $\mathcal{K} = \{0,1\}^k$ with uniform distribution.
- $\mathcal{M} = \mathcal{C} = \{0,1\}^n, n \leq k$
- $E_K(M) = C$ where $C_i \leftarrow M_i \oplus K_i$ for $1 \leq i \leq n$.
- $D_K(C) = M'$ where $M'_i \leftarrow C_i \oplus K_i$ for $1 \leq i \leq n$.

Note that the encryption and decryption algorithms are identical (bitwise XOR with the key) and that the length of the message can be at most the length of the key (one-time pad).

Actually, in many situations, one will want to use a one-time pad with $k$ much larger than $n$ and send more than one message. In this case $E$ and $D$ above take a counter argument which is an offset into the one-time pad. The counter is then be sent with the ciphertext to keep the sender and receiver synchronized.

If the sender ever reuses a part of $K$, then for some pair of ciphertexts, the adversary can learn the XOR of (part of) the two messages. For example if $k = n$,

$$C \oplus C' = (M \oplus K) \oplus (M' \oplus K) = M \oplus M'$$

A one-time pad is a vehicle that allows one to time-shift face-to-face communication and thus has been used in scenarios in which parties exchange extensive code-books and then later communicate over insecure channels. One difficulty of this doing this by hand is the potential to forget

to update the counter which could end up with re-use of the pad which would therefore reveal the XOR of two parts of a message stream. It has been recently revealed (in the archives of the Venona project at the NSA) that such errors were used to decrypt parts of messages about nuclear secrets sent by Julius Rosenberg in the early 1950's and were part of the evidence against him.

However, in terms of information theory, if used only once, Shannon showed that one-time pads have perfect security (see below).

# 7 Symmetric authentication

This is known as a MAC (message authentication code) which involves a "tag" for each message. We don't want the adversary to be able to convince the receiver that sender's message was anything else. In this case the sender must deal with an adversary that actively interferes with the channel and may corrupt what is sent on it.

**Message Authentication Protocol:**
Given that the key $K$ has been chosen:

- Sender to send message $M$ computes $tag \leftarrow MAC_K(M)$ and sends $\langle M, tag \rangle$
- Receiver gets the corrupted $\langle M', tag' \rangle$ and computes $tag'' \leftarrow MAC_K(M')$. Rejects iff $tag' \neq tag''$.

We can combine secrecy and authentication by inserting encryption before sending the message/tag pair and decryption after receiving the message/tag pair.

# 8 Asymmetric setting

Two keys for each party (A for Alice, B for Bob).

- public keys $PK_A, PK_B$
- secret keys $SK_A, SK_B$

## 8.1 Asymmetric Encryption

Only requires the keys for the receiver (Bob). This is required otherwise Eve would have the same information as Bob.

- Sender computes $C \leftarrow E(PK_B, M)$ and sends it.
- Receiver gets $C'$ and computes $M' \leftarrow D(SK_B, C)$
- Adversary gets receiver's public keys and the algorithms $E$ and $D$

Note that we assume that the encryption method $E$ is part of the sender. In some circumstances the sender has to use a separate device to do encryption so an adversary could potentially intercept the message in between the sender and the algorithm $E$. This is a vulnerability that is part of ensuring general security but is outside the purview of this course.

Asymmetric encryption is usually used to exchange symmetric secret keys, which are faster for the bulk of communication (SSL works this way).

The original Diffie-Hellman paper noted the close connection between the schemes for asymmetric encryption and asymmetric authentication.

# 9 Asymmetric authentication

Known as "digital signatures," uses only the keys for the sender (Alice). (This is necessary else Alice and Eve would have the information.)

- Sender computes $S \leftarrow Sign(SK_A, M)$ and sends $\langle M, S \rangle$
- Receiver gets $\langle M', S' \rangle$ and computes $S'' \leftarrow Verify(PK_A, M')$. Rejects iff $S'' \neq S'$.

# 10 Security definition

Shannon in 1949 gave an information-theoretic definition of security.

**Definition 10.1.** *A symmetric encryption scheme is **perfectly secure** iff*

$$\forall distributions on \mathcal{M} \forall M \in \mathcal{M} : \Pr_{K \in \mathcal{K}, M \in \mathcal{M}}[M \mid C = E_K(M)] = \Pr_{K \in \mathcal{K}, M \in \mathcal{M}}[M]$$

*In other words, the seeing the ciphertext gives no additional information about what the message is, provided the key is secret. Equivalently:*

$$\Pr_{M,K}[C \mid M] = \Pr_{M,K}[C]$$

We will prove the following theorems next time.

**Theorem 10.2.** *If a symmetric encryption scheme is perfectly secure, then $|\mathcal{M}| \leq |\mathcal{K}|$.*

**Theorem 10.3.** *For $n \leq k$, a one-time pad is perfectly secure.*