## 1 Announcements

- Course webpage is now up: `www.ece.umd.edu/˜danadach/Crypto_Physical2013/coursepage.html`
- Sign up sheet for paper presentations will be up on Canvas in the next couple of days.

## 2 Recap and Overview

Previous lecture:

- Discussed Pseudorandomness
- Gave definition for pseudorandom generators (PRG).

This lecture: Private Key Encryption Schemes.

- Definition of private key encryption schemes
- Security against eavesdroppers for private key encryption schemes
- Constructing private key encryption secure against eavesdroppers from PRG.
- Stream Ciphers
- CPA security for private key encryption schemes.

## 3 Review: Definition of PRG

A pseudorandom generator is a deterministic algorithm that receives a short truly random seed and stretches it into a long string that is pseudorandom. $n$ is the length of the seed that is input and $\ell(n)$ is the output length.

**Definition 1.** *Let $\ell(\cdot)$ be a polynomial and let $G$ be a deterministic polynomial-time algorithm such that for any input $s \in \{0,1\}^n$, algorithm $G$ outputs a string of length $\ell(n)$. We say that $G$ is a pseudorandom generator if the following two conditions hold:*

1. *(Expansion:) For every $n$ it holds that $\ell(n) > n$.*
2. *(Pseudorandomness:) For all ppt distinguishers $D$, there exists a negligible function* neg *such that*

$$|\Pr[D(r) = 1] - \Pr[D(G(s)) = 1]| \leq \text{neg}(n)$$

*where $r$ is chosen uniformly at random from $\{0,1\}^{\ell(n)}$, the seed $s$ is chosen uniformly at random from $\{0,1\}^n$, and the probabilities are taken over the random coins used by $D$ and the choice of $r$ and $s$.*

*The function $\ell(\cdot)$ is called the expansion factor of $G$.*

## 4 Tangent: The One-Time Pad

1. Fix an integer $\ell > 0$. Then the message space $\mathcal{M}$, key space $\mathcal{K}$, and ciphertext space $\mathcal{C}$ are all equal to $\{0,1\}^\ell$.
2. The key-generation algorithm $\mathsf{Gen}$ works by choosing a string from $\mathcal{K} = \{0,1\}^\ell$ according to the uniform distribution.
3. Encryption $\mathsf{Enc}$ works as follows: given a key $\textsc{sk} \in \{0,1\}^\ell$ and a message $m \in \{0,1\}^\ell$, output $C := \textsc{sk} \oplus m$.
4. Decryption $\mathsf{Dec}$ works as follows: given a key $\textsc{sk} \in \{0,1\}^\ell$ and a ciphertext $C \in \{0,1\}^\ell$, output $m := \textsc{sk} \oplus C$.

See Introduction to Modern Cryptography (page 35) for proof of perfect secrecy.

## 5 Defining Computationally-Secure Encryption

**Definition 2.** *A* private-key encryption scheme *is a tuple of probabilistic polynomial-time algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *such that:*

1. *The* key-generation algorithm $\mathsf{Gen}$ *takes as input the security paratmeter* $1^n$ *and outputs a key* $\mathtt{SK}$*; we write this as* $\mathtt{SK} \leftarrow \mathsf{Gen}(1^n)$*. We will assume without loss of generality that any key* $\mathtt{SK}$ *output by* $\mathsf{Gen}(1^n)$ *satisfies* $|\mathtt{SK}| \geq n$*.*
2. *The* encryption algorithm $\mathsf{Enc}$ *takes as input a key* $\mathtt{SK}$ *and a plaintext message* $m \in \{0,1\}^*$*, and outputs a ciphertext* $C$*. Since* $\mathsf{Enc}$ *may be randomized, we write this as* $c \leftarrow \mathsf{Enc}_{\mathtt{SK}}(m)$*.*
3. *The* decryption algorithm $\mathsf{Dec}$ *takes as input a key* $\mathtt{SK}$ *and a ciphertext* $C$ *and outputs a message* $m$*. We assume that* $\mathsf{Dec}$ *is deterministic and so write this as* $m = \mathsf{Dec}_{\mathtt{SK}}(C)$*.*

*It is required that for every* $n$*, every key* $\mathtt{SK}$ *output by* $\mathsf{Gen}(1^n)$*, and every* $m \in \{0,1\}^*$*, it holds that* $\mathsf{Dec}_{\mathtt{SK}}(\mathsf{Enc}_{\mathtt{SK}}(m)) = m$*.*

*If* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is such that for* $\mathtt{SK}$ *output by* $\mathsf{Gen}(1^n)$*, algorithm* $\mathsf{Enc}_{\mathtt{SK}}$ *is only defined for messages* $m \in \{0,1\}^{\ell(n)}$*, then we say that* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a* fixed-length *private-key encryption scheme for messages of length* $\ell(n)$*.*

### 5.1 The Basic Definition of Security

Motivating the definition: Any definition of security consists of two distinct components: a specification of the assumed power of the adversary, and a description of what constitutes a "break" of the scheme. We consider the case of an *eavesdropping adversary* who observes the encryption of a single message. Adversary is computationally bounded and limited to running in polynomial time.

Note: No assumptions made about the adversary's *strategy*. This is crucial for obtaining meaningful notions of security because it is impossible to predict all possible strategies.

Idea behind the definition: An adversary should be unable to learn *any partial information* about the plaintext from the ciphertext. The definition of *semantic security* directly formalizes exactly this notion. Definition of semantic security is complex and difficult to work with. Fortunately, there is an equivalent definition in terms of *indistinguishability* which is much simpler.

**Indistinguishability in the presence of an eavesdropper.** We now give the formal definition. The experiment is defined for any private-key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, any adversary $\mathcal{A}$, and any value $n$ for the security parameter:

The eavesdropping indistinguishability experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\mathcal{E}}(n)$:

1. The adversary $\mathcal{A}$ is given input $1^n$, and outputs a pair of messages $m_0, m_1$ of the same length.
2. A key $\mathtt{SK}$ is generated by running $\mathsf{Gen}(1^n)$, and a random bit $b \leftarrow \{0,1\}$ is chosen. A ciphertext $C \leftarrow \mathsf{Enc}_{\mathtt{SK}}(m_b)$ is computed and given to $\mathcal{A}$. We call $C$ the *challenge ciphertext*.
3. $\mathcal{A}$ outputs a bit $b'$.
4. The output of the experiment is defined to be 1 if $b' = b$ and 0 otherwise. if $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\mathcal{E}}(n) = 1$, we say that $\mathcal{A}$ *succeeded.*

Note that in the definition, there is no limitation on the length of the messages $m_0, m_1$ as long as they are the same. If $\mathcal{E}$ is a fixed-length scheme for messages of length $\ell(n)$, the above experiment is modified by requiring $m_0, m_1 \in \{0,1\}^{\ell(n)}$.

**Definition 3.** *A private-key encryption scheme* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has* indistinguishable encryptions in the presence of an eavesdropper *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$ *there exists a negligible function* neg *such that*

$$\Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\mathcal{E}}(n) = 1] \leq \frac{1}{2} + \mathrm{neg}(n),$$

*where the probability is taken over the random coins used by* $\mathcal{A}$*, as well as the random coins used in the experiment.*

### 5.2 The Power of the Indistinguishability Definition

We begin by showing that indistinguishability implies that no single bit of a *randomly chosen* plaintext can be guessed with probability significantly better than $1/2$.

*Claim.* Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a private-key encryption scheme that has indistinguishable encryptions in the presence of an eavesdropper. Then for all ppt adversaries $\mathcal{A}$ and all $i$, there exists a negligible function $\mathrm{neg}$ such that:

$$\Pr[\mathcal{A}(1^n, \mathsf{Enc}_{\textsc{sk}}(m)) = m^i] \leq \frac{1}{2} + \mathrm{neg}(n).$$

Sketched the proof in class, for full proof, see Introduction to Modern Cryptography, pages 64-67.

In fact, loosely speaking, no ppt adversary can learn *any* function of the plaintext message given the ciphertext, and furthermore this holds regardless of the a priori distribution over the message being sent. This is the definition of Semantic Security.

See Introduction to Modern Cryptography, page 68.

**Definition 4.** *A private-key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is* semantically secure in the presence of an eavesdropper *if for every probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a probabilistic polynomial-time algorithm $\mathcal{A}'$ such that for all efficiently-sampleable distributions $X = (X_1, \ldots)$ and all polynomial-time computable functions $f$ and $h$, there exists a negligible function* $\mathrm{neg}$ *such that*

$$|\Pr[\mathcal{A}(1^n, \mathsf{Enc}_{\textsc{sk}}(m), h(m)) = f(m)] - \Pr[\mathcal{A}'(1^n, h(m)) = f(m)]| \leq \mathrm{neg}(n),$$

*where $m$ is chosen according to distribution $X_n$ and the probabilities are taken over the choice of $m$ and the key $k$, and any random coins used by $\mathcal{A}$, $\mathcal{A}'$, and the encryption process.*

## 6 Constructing Secure Encryption Schemes

Picture:

Construction:

Let $G$ be a pseudorandom generator with expansion factor $\ell$. Define a private-key encryption scheme for messages of length $\ell$ as follows:

- $\mathsf{Gen}$ : on input $1^n$, choose $\textsc{sk} \leftarrow \{0,1\}^n$ uniformly at random and output it as the key.
- $\mathsf{Enc}$ : on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^{\ell(n)}$, output the ciphertext

$$C := G(\textsc{sk}) \oplus m.$$

- $\mathsf{Dec}$ : on input a key $\textsc{sk} \in \{0,1\}^n$ and a ciphertext $C \in \{0,1\}^{\ell(n)}$, output the plaintext message

$$m := G(\textsc{sk}) \oplus C.$$

We now prove that the given encryption scheme has indistinguishable encryptions in the presence of an eavesdropper, under the *assumption* that $G$ is a pseudorandom generator.

**Theorem 1.** *If $G$ is a pseudorandom generator, then the construction above is a fixed-length private-key encryption scheme that has indistinguishable encryptions in the presence of an eavesdropper.*

*Proof.* Let $\mathcal{A}$ be a ppt adversary "breaking" $\mathcal{E}$. I.e.:

$$\varepsilon(n) = \Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\mathcal{E}}(n) = 1] - \frac{1}{2},$$

and $\varepsilon(\cdot)$ is a non-negligible function. We use $\mathcal{A}$ to construct a distinguisher $D$ for the pseudorandom generator $G$, which succeeds with probability $\varepsilon(n)$ (thus contradicting the security of $G$).

The distinguisher is given a string $w$ as input, and its goal is to determine whether $w$ was chosen uniformly at random or whether $w$ was generated by choosing a random $\textsc{sk}$ and computing $w := G(\textsc{sk})$. $D$ emulates the eavesdropping experiment for $\mathcal{A}$ in the manner described below.

**Distinguisher $D$:** $D$ is given as input a string $w \in \{0,1\}^{\ell(n)}$.

1. Run $\mathcal{A}(1^n)$ to obtain a pair of messages $m_0, m_1 \in \{0,1\}^{\ell(n)}$.
2. Choose a random bit $b \leftarrow \{0,1\}$. Set $C := w \oplus m_b$.
3. Give $C$ to $\mathcal{A}$ and obtain output $b'$. Output 1 if $b' = b$, and output 0 otherwise.

Now, if $w$ is chosen uniformly at random from $\{0,1\}^{\ell(n)}$, the view of $\mathcal{A}$ consists of $C := w \oplus m_b$, which is also just a uniformly random string when $w$ is uniformly random. Therefore, when $w$ is chosen uniformly at random, $\mathcal{A}$ succeeds with probability *exactly* $\frac{1}{2}$. Thus, $\Pr[D(w) = 1] = \frac{1}{2}$.

On the other hand, if $w = G(\textsc{sk})$, then the view of $\mathcal{A}$ generated by $D$ is distributed *identically* to the view of $\mathcal{A}$ in experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\mathcal{E}}(n)$. Thus, $\Pr[D(w) = 1] = \Pr[D(G(\textsc{sk}) = 1] = \Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\mathcal{E}}(n) = 1] = \frac{1}{2} + \varepsilon(n)$.

Therefore,

$$|\Pr[D(w) = 1] - \Pr[D(G(\textsc{sk}) = 1]| = \varepsilon(n).$$

If $\varepsilon(n)$ is non-negligible, then this contradicts the security of the prg $G$.

How is this an improvement over one-time pad?

## 7 Stream Ciphers and Multiple Encryption

In the cryptographic literature, an encryption scheme of the type presented in the previous two sections is often called a *stream cipher*. This is due to the fact that encryption is carried out by first generating a *stream* of pseudorandom bits, and then XORing this stream with the plaintext. By stream cipher, we refer to the *algorithm* that generates the stream (i.e. the pseudorandom generator $G$). Secure stream cipher should satisfy the definition of a variable output-length pseudorandom generator.

### 7.1 Stream Ciphers in Practice

(See Introduction to Modern Cryptography, pages 77-78). There are a number of practical constructions of stream ciphers available, and these are typically extraordinarily fast. A popular example is the stream cipher RC4 which is widely considered to be secure when used appropriately.

Linear feedback shift registers (LFSRs) have, historically, also been popular as stream ciphers. Have been shown to be insecure.

May have heard of stream cipher vs. block cipher. Block cipher is "more secure", but stream ciphers are faster, more efficient.

## 7.2 Security for Multiple Encryptions

The experiment is defined for any private-key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, any adversary $\mathcal{A}$, and any value $n$ for the security parameter:

The multiple-message eavesdropping experiment $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\mathcal{E}}(n)$:

1. The adversary $\mathcal{A}$ is given input $1^n$, and outputs a pair of *vectors* of messages $\overline{M}_0 = (m_0^1, \ldots, m_0^t)$ and $\overline{M}_1 = (m_1^1, \ldots, m_1^t)$ with $|m_0^i| = |m_1^i|$ for all $i$.
2. A key $\mathsf{SK}$ is generated by running $\mathsf{Gen}(1^n)$, and a random bit $b \leftarrow \{0, 1\}$ is chosen. For all $i$, the ciphertext $C^i = \mathsf{Enc}_{\mathsf{SK}}(m_b^i)$ is computed and the vector of ciphertexts $\overline{C} = (c^1, \ldots, c^t)$ is given to $\mathcal{A}$.
3. $\mathcal{A}$ outputs a bit $b'$.
4. The output of the experiment is defined to be $1$ if $b' = b$ and $0$ otherwise.

**Definition 5.** *A private-key encryption scheme* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has* indistinguishable multiple encryptions in the presence of an eavesdropper *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$ *there exists a negligible function* neg *such that*

$$\Pr[\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\mathcal{E}}(n) = 1] \leq \frac{1}{2} + \mathrm{neg}(n),$$

*where the probability is taken over the random coins used by* $\mathcal{A}$*, as well as the random coins used in the experiment.*

Crucial observation: Security for a *single* encryption does not imply security under *multiple* encryptions.

***EXAMPLE:*** Consider the following adversary $\mathcal{A}$ attacking $\mathcal{E}$. $\mathcal{A}$ outputs the vectors $\overline{M}_0 =$, $\overline{M}_1 =$. Let $\overline{C} = (c^1, c^2)$ be the vector of ciphertexts that $\mathcal{A}$ receives. If $c^1 = c^2$, $\mathcal{A}$ outputs 0; otherwise, $\mathcal{A}$ outputs 1.

Main point: Construction $\mathcal{E}$ is *deterministic*, so if the same message is encrypted multiple times then the same ciphertext results each time.

The mere knowledge that the same message has been re-sent can provide significant information and has historically been very useful to cryptanalysts.

## 7.3 Secure Multiple Encryptions Using a Stream Cipher

There are typically two ways in which a stream cipher/pseudorandom generator is used in practice to securely encrypt multiple plaintexts:

1. **Synchronized mode:** In this mode, the communicating parties use a different part of the stream output by the stream cipher in order to encrypt each messsage. This mode is "synchronized" because both parties need to know which parts of the stream have already been used in order to prevent re-use.

2. **Unsynchronized mode:** In this mode, encryptions are carried out independently of one another and the parties do not need to maintain state.

   Require stronger property of the pseudorandom generator.

## 8 Security Against Chosen-Plaintext Attacks (CPA)

Until now we have considered a relatively weak adversary who only passively eavesdrops on the communication between two honest parties. Now consider a more powerful type of adversarial attack, alled a *chosen-plaintext attack (CPA)*.

The experiment is defined for any private-key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, any adversary $\mathcal{A}$, and any value $n$ for the security parameter:

The CPA indistinguishability experiment $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\mathcal{E}}(n)$:

1. A key $\mathsf{SK}$ is generated by running $\mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ is given input $1^n$ and oracle access to $\mathsf{Enc}_{\mathsf{SK}}(\cdot)$ and outputs a pair of messages $m_0, m_1$ of the same length.
3. A random bit $b \leftarrow \{0, 1\}$ is chosen and then a ciphertext $C \leftarrow \mathsf{Enc}_{\mathsf{SK}}(m_b)$ is computed and given to $\mathcal{A}$. We call $C$ the *challenge ciphertext*.
4. The adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_{\mathsf{SK}}(\cdot)$, and outputs a bit $b'$.
5. The output of the experiment is defined to be 1 if $b' = b$ and 0 otherwise. if $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\mathcal{E}}(n) = 1$, we say that $\mathcal{A}$ *succeeded*.

**Definition 6.** *A private-key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has* indistinguishable encryptions under a chosen-plaintext attack *if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there exists a negligible function* neg *such that*

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\mathcal{E}}(n) = 1] \leq \frac{1}{2} + \mathrm{neg}(n),$$

*where the probability is taken over the random coins used by $\mathcal{A}$, as well as the random coins used in the experiment.*

**Proposition 1.** *Any private-key encryption scheme that has indistinguishable encryptions under a chosen-plaintext attack also has indistinguishable* multiple *encryptions under a chosen-plaintext attack.*

## 9 Pseudorandom Functions

As we have seen, pseudorandom generators can be used to obtain security in the presence of eavesdropping adversaries. The notion of peudorandomness is also instrumental in obtaining security against chosen-plaintext attacks. Now, however, instead of considering pseudorandom *strings*, we consider pseudorandom *functions*.

Specifically interested in: Pseudorandom functions mapping $n$-bit strings to $n$-bit strings. Does not make sense to say that any *fixed* function is pseudrandom. Thus, we must technically refer to the pseudorandomness of a *distribution* on functions. An easy way to do this is to consider *keyed functions*, defined next.

Keyed Functions: A keyed function $F$ is a two-input function $F : \{0, 1\}^* \rightarrow \{0, 1\}^*$, where the first input is called the *key* and denoted $\mathsf{SK}$, and the second input is just called the input. In general, the key $\mathsf{SK}$ will be chosen and then *fixed*, and we will then be interested in the single-input function $F_{\mathsf{SK}} : \{0, 1\}^* \rightarrow \{0, 1\}^*$ defined by $F_{\mathsf{SK}}(x) = F(\mathsf{SK}, x)$. We say that $F$ is *efficient* if there is a deterministic polynomial-time algorithm that computes $F(\mathsf{SK}, x)$ given $\mathsf{SK}, x$ as input. We will only be interested in functions $F$ that are efficient.

Keyed function induces a natrual distribution on functions.

Intuitively, we call $F$ *pseudorandom* if the function $F_{\mathsf{SK}}$ is indistinguishable from a function chosen uniformly at random from the set of all functions having the same domain and range.

Definition of $\mathsf{Func}_n$. How large is $\mathsf{Func}_n$? $(2^n)^{2^n} = 2^{n \cdot 2^n}$. Pseuorandom functions are chosen from a distribution over $2^n$ functions. Despite this, the "behavior" of these functions must look the same to any polynomial-time distinguisher.

Cannot give distinguisher $D$ access to the "code" of the function $F_{\mathsf{SK}}$ vs $f$. Actual definition gives $D$ *oracle access* to the function in question $F_{\mathsf{SK}}$ or $f$.

Oracle will always return the same response when queried twice on the same input.

**Definition 7.** *Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length-preserving, keyed function. We say that $F$ is a* pseudorandom function *if for all probabilistic polynomial-time distinguishers $D$, there exists a negligible function* $\mathrm{neg}$ *such that:*

$$\left|\Pr[D^{F_{\mathrm{SK}}(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1]\right| \leq \mathrm{neg}(n),$$

*where $\mathtt{SK} \leftarrow \{0,1\}^n$ is chosen uniformly at random and $f$ is chosen uniformly at random from the set of functions mapping $n$-bit strings to $n$-bit strings.*

Note: The key $\mathtt{SK}$ is not given to $D$
Note: Pseudorandom functions exist iff pseudorandom generators exist.

## 10 CPA-Secure Encryption Schemes from Pseudorandom Functions

Picture:

Construction:
Let $F$ be a pseudorandom function. Define a private-key encryption scheme for messages of length $n$ as follows:

- $\mathsf{Gen}$ : on input $1^n$, choose $\mathtt{SK} \leftarrow \{0,1\}^n$ uniformly at random and output as the key.
- $\mathsf{Enc}$ : on input a key $\mathtt{SK} \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, choose $r \leftarrow \{0,1\}^n$ uniformly at random and output the ciphertext
$$C := \langle r, F_{\mathrm{SK}}(r) \oplus m \rangle.$$
- $\mathsf{Dec}$ : on input a key $k \in \{0,1\}^n$ and a ciphertext $c = \langle r, s \rangle$, output the plaintext message
$$m := F_{\mathrm{SK}}(r) \oplus s.$$