Midterm this Thursday, Oct. 19.

• In class
• Open book (including textbook), no electronic devices
• Will cover material through Diffie-Hellman and key exchange, but not public key encryption.

Solution sets for PS #6 and last year’s midterm are now posted on ELMS.

For last year’s midterm, I strongly recommend doing your best to try the midterm problems before looking at the solutions.
Pseudorandomness

Pseudorandom generator $G(y)$

- One input $y$, “seed”
- Output looks like a random string when $s$ unknown
- Output should be longer than the seed
- Generally only good for EAV security
- Stream cipher is a more flexible version

Pseudorandom function $F_k(r)$

- Two inputs: $k$ (key) and $r$
- For fixed but unknown $k$, looks like a random function of $r$
- Output can be the same size or smaller than the input
- Useful for CPA security
- Block cipher is a fixed-size version, but must be a permutation (with computable inverse for known $k$)
**Stream Ciphers**

**Pseudorandom generator** $G(y)$
- Input only seed $y$, which has length $s$
- Single output string $G(s)$ of length $\ell(s)$

**Stream cipher**
- Input seed $x$, but *may also* take initial value IV as input
- With IV, can provide CPA security
- Has two component functions Init and Next
- Each time Next is called, the stream cipher outputs a fixed number of bits. I.e., output can be any length
- Function $y \mapsto (x_0, x_1, x_2, \ldots, x_\ell)$ is a pseudorandom generator

```
\[ y \rightarrow \text{Init} \rightarrow \text{Next} \rightarrow \text{Next} \rightarrow \text{Next} \rightarrow \ldots \]
```

This class is being recorded
Suppose we have a message \( m = m_1, m_2, m_3, \ldots, m_a \) of length \( a \) and we wish to encrypt using a stream cipher to get EAV security using key \( k \):

1. Run `Init` using input \( k \) and no IV or fixed IV.
2. Run `Next` \( a \) times to get \( x_1, x_2, x_3, \ldots, x_a \).
3. Ciphertext is \( c = (m_1 \oplus x_1, m_2 \oplus x_2, \ldots, m_a \oplus x_a) \).

For CPA security:

1. Choose random IV. Run `Init` using input \( k \) and IV.
2. Run `Next` \( a \) times to get \( x_1, x_2, x_3, \ldots, x_a \).
3. Ciphertext is \( c = (IV, m_1 \oplus x_1, m_2 \oplus x_2, \ldots, m_a \oplus x_a) \).
Block Ciphers

Pseudorandom function $F_k(r)$

- Inputs $k$ (key) and $r$ have length $n$, a variable (although $k$ could have a different length than $r$)
- For fixed but unknown $k$, looks like a random function of $r$
- Output can be the same size or smaller than the input

Block cipher $F_k(r)$

- Also inputs $k$ (key) and $r$ but now they are fixed length
- For fixed but unknown $k$, looks like a random function of $r$
- Output must be same size as $r$ is
- Is a permutation: has computable inverse for known $k$
Break up message $m$ into blocks of fixed size: $m_1, m_2, m_3, \ldots$. 

**CTR Mode**

- Message: 
  - $m_1$
  - $m_2$
  - $m_3$

- Ciphertext: 
  - $c_1$
  - $c_2$
  - $c_3$

**CBC Mode**

- Message: 
  - $m_1$
  - $m_2$
  - $m_3$

- Ciphertext: 
  - $c_1$
  - $c_2$
  - $c_3$
• CTR and CBC modes are both CPA-secure.
• IV is needed to provide randomness as part of the encoding so that the same message doesn’t always have the same ciphertext.
• IV must be sent as part of the ciphertext so that Bob can decode.
• Another mode, ECB mode (below) is insecure, even against EAV attacks.
Goals of Block Cipher Design

- Must be invertible to use with CBC mode (i.e., pseudorandom permutation rather than pseudorandom function).
- Even when the inputs are related, the outputs should be very different.

In particular, the change of even a single bit of the input should result in a totally different output. This is known as the “avalanche effect.” It is often achieved by having multiple rounds, each of which magnifies small changes.
A **Feistel network** consists of a sequence of rounds sequentially acting on the message, which is split into a left and right half.

In each round, the current right half is fed into a **round function** $f$ with a key for the round and then XORed with the left half. The modified left half and old right half are then switched.
Variants of substitution-permutation networks are used for both the DES mangler function and for AES.

1. State is mixed with round key.
2. Substitution step using small invertible S-boxes.
3. Permute bits.
The **S-boxes** introduce **confusion**: They change their inputs into totally different strings and magnify single-bit changes. However, the S-box is small and acts on only a few bits, so the confusion is **only local**.

Then the **permutation step** causes **diffusion**: whatever local confusion was introduced by the S-boxes spreads out to many different locations.

> Multiple rounds of substitution and permutation cause the confusion to be magnified further and continue to spread around.

**We need both to get an avalanche effect.**

**You also need key mixing:** This is a permutation, and without the key, Eve can just trace the permutation backwards to get the input.
Imagine you have a disease spreading. It starts with one patient.

**Confusion:** The disease infects additional people in the same city as someone who is sick.

**Diffusion:** Some people travel to different cities.
Imagine you have a disease spreading. It starts with one patient.

**Confusion:** The disease infects additional people in the same city as someone who is sick.

**Diffusion:** Some people travel to different cities.

1. Confusion

![Diagram showing disease spread]

This class is being recorded
Imagine you have a disease spreading. It starts with one patient.

**Confusion:** The disease infects additional people in the same city as someone who is sick.

**Diffusion:** Some people travel to different cities.
Imagine you have a disease spreading. It starts with one patient.

**Confusion:** The disease infects additional people in the same city as someone who is sick.

**Diffusion:** Some people travel to different cities.
Imagine you have a disease spreading. It starts with one patient.

**Confusion:** The disease infects additional people in the same city as someone who is sick.

**Diffusion:** Some people travel to different cities.
DES Overview

DES is a Feistel network.

The mangler function $f$ is a substitution-permutation network.
The AES permutation takes a 128-bit input represented as a 4 x 4 matrix of bytes:

AES is basically a substitution-permutation network.