CIS 5560

Cryptography
Lecture 7

Course website:

pratyushmishra.com/classes/cis-5560-s24/

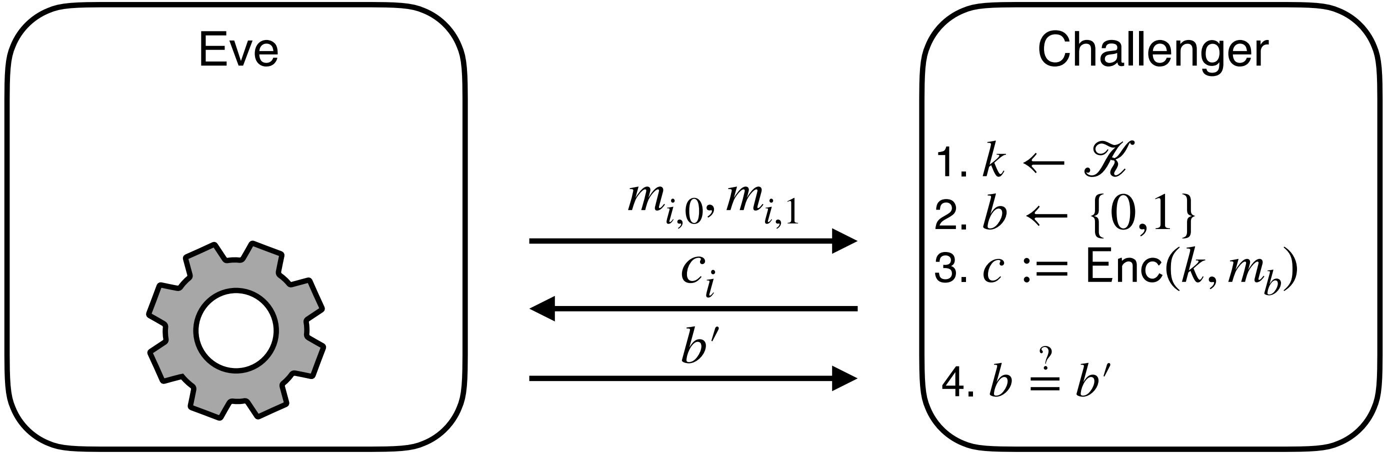
Slides adapted from Dan Boneh and Vinod Vaikuntanathan

Announcements

- **HW 3 out after lecture**
    - Due **Tuesday**, Feb 13 at 1PM on Gradescope
    - Covers PRGs, OWFs
- Converting Matan's OH to a **Homework Party**
    - Work on homework problems with other students
        - (Still have to write up your own answers!)
    - TA(s) and I and will be around for answering questions
    - Good way to meet other students in class and make friends =)

Recap of last lecture

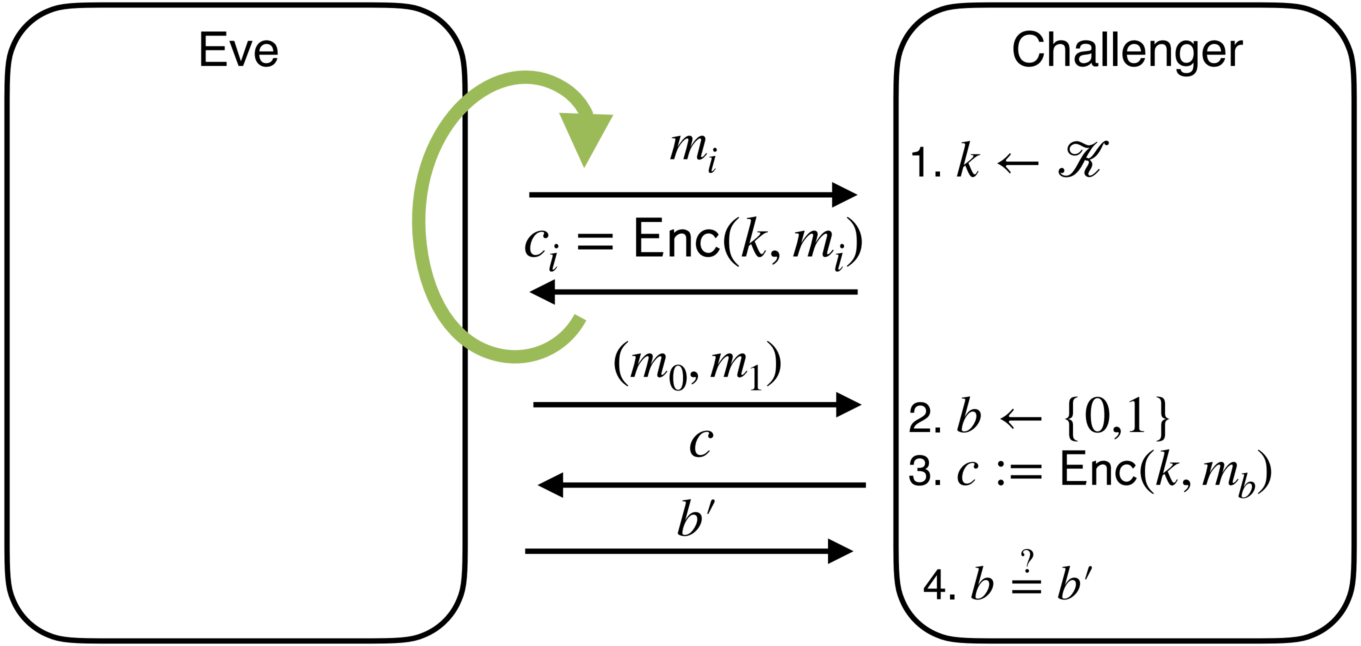
Semantic Security for Many Msgs



For every **PPT** Eve, there exists a negligible fn ε,

$$\Pr\left[\mathsf{Eve}(c_q) = b \;\middle|\; \begin{array}{r} k \leftarrow \mathcal{K} \\ b \leftarrow \{0,1\} \\ \text{For } i \text{ in } 1,\ldots,q: \\ (m_{i,0}, m_{i,1}) \leftarrow \mathsf{Eve}(c_{i-1}) \\ c_i = \mathsf{Enc}(k, m_{i,b}) \end{array}\right] < \frac{1}{2} + \varepsilon(n)$$

Alternate (Stronger?) definition



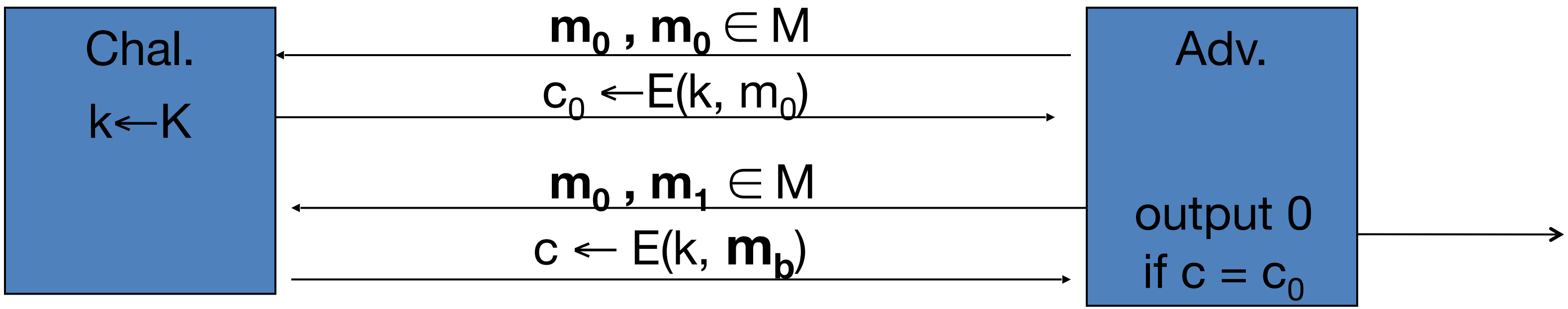
Also called "IND-CPA": Indistinguishability under Chosen-Plaintext Attacks

Equivalent to previous definition: just set $m_{i,0} = m_{i,1} = m_i$

Stream Ciphers insecure under CPA

Problem: E(k,m) outputs same ciphertext for msg m.

Then:



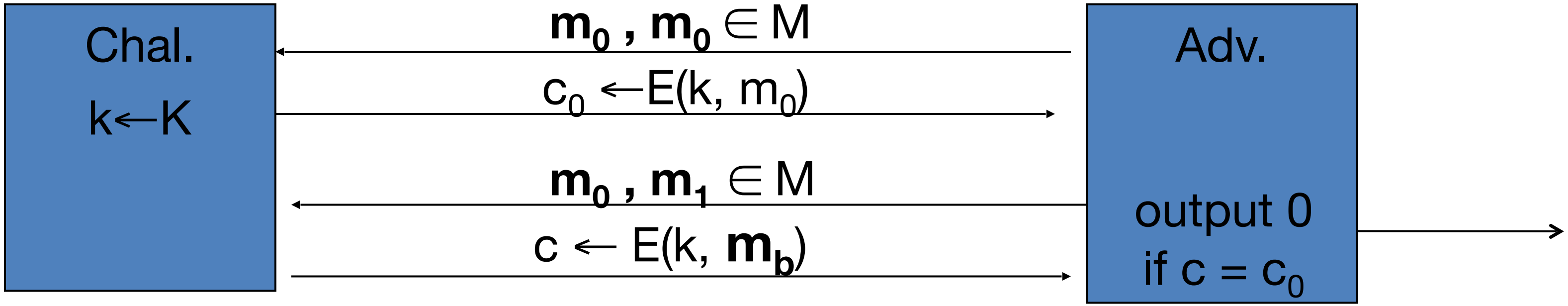
So what? an attacker can learn that two encrypted files are the same, two encrypted packets are the same, etc.

- Leads to significant attacks when message space M is small

Stream Ciphers insecure under CPA

Problem: E(k,m) always outputs same ciphertext for msg m.

Then:



If secret key is to be used multiple times ⇒

given the same plaintext message twice, encryption must produce different outputs.

Today's Lecture

- Deeper look at PRFs
- PRFs → multi-message encryption
- Hybrid argument
- PRGs → PRFs

Pseudorandom Functions

Collection of functions $\mathscr{F}_\ell = \{F_k : \{0,1\}^\ell \rightarrow \{0,1\}^m\}_{k \in \{0,1\}^n}$

- indexed by a key k
- n: key length, ℓ: input length, m: output length.
- Independent parameters, all poly(sec-param) = poly(n)
- #functions in $\mathscr{F}_\ell \leq 2^n$ (singly exponential in n)

$\mathbf{Gen}(1^n)$: Generate a random n-bit key k.

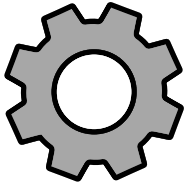
$\mathbf{Eval}(k, x)$ is a poly-time algorithm that outputs $F_k(x)$

How to define security?

Let’s try to build it up like the PRG security definition

PRG Security

Adv $\mathscr{A}$



y ←

b' →

Challenger

1. $b \leftarrow \{0,1\}$
2. If b = 0
   1. Sample $y \leftarrow \{0,1\}^m$
3. If b = 1
   1. Sample $s \leftarrow \{0,1\}^n$
   2. Set $y := G(s)$
4. $b \stackrel{?}{=} b'$

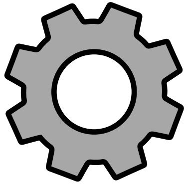
$$\Pr[b = b'] = 1/2 + \mathsf{negl}(n)$$

PRG vs PRF

- So, for PRG security, we give the adversary either a random string or a pseudorandom string, and ask it to figure out which one it is
- Can the same strategy work for PRFs?

PRF Security - Attempt 1

Adv $\mathscr{A}$



$\xrightarrow{x}$

$\xleftarrow{f}$

$\xrightarrow{b'}$

Challenger

1. $b \leftarrow \{0,1\}$
2. If b = 0
   1. Sample $f \leftarrow \mathsf{Fns}[X, Y]$
3. If b = 1
   1. Sample $k \leftarrow \mathscr{K}$
   2. Set $f(\,\cdot\,) = F(k, \cdot\,)$
4. $b \stackrel{?}{=} b'$

$$\Pr[b = b'] = 1/2 + \mathsf{negl}(n)$$

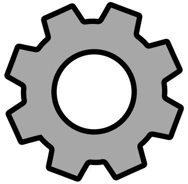
PRF Security - Attempt 1

- What's the problem with this?
- Hint: What does a random function look like?
  - Is it efficiently evaluatable?
  - Does it have a short description?
  - It maps inputs to random values (example on board)

- **Ans: we can't easily send a random function!**
- **So: how about we give the challenger "oracle" access**

PRF Security - Attempt 2

Adv $\mathscr{A}$



Challenger

1. $b \leftarrow \{0,1\}$
2. If b = 0
   1. Sample $f \leftarrow \mathsf{Fns}[X, Y]$
3. If b = 1
   1. Sample $k \leftarrow \mathscr{K}$
   2. Set $f(\cdot) := F_k(\cdot)$

$\mathscr{A} \xrightarrow{x}$ Challenger

4. Set $y := f(x)$

Challenger $\xrightarrow{y} \mathscr{A}$

$\mathscr{A} \xrightarrow{b'}$ Challenger

5. $b \stackrel{?}{=} b'$

$$\Pr[b = b'] = 1/2 + \mathsf{negl}(n)$$

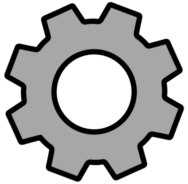
PRF Security - Attempt 1

- Q: How many questions should the adversary be allowed to ask?
  - 1
  - 2
  - poly(n)
  - exp(n)

- Why is 1 insufficient? Can't tell any information from 1 query
- Why is exp(n) too many? Adv will run in exponential time!

PRF Security - Attempt 2

Adv $\mathscr{A}$



Poly(n) queries

x

y

b'

Challenger

1. $b \leftarrow \{0,1\}$
2. If b = 0
    1. Sample $f \leftarrow$ Fns$[X, Y]$
3. If b = 1
    1. Sample $k \leftarrow \mathscr{K}$
    2. Set $f(\cdot) := F_k(\cdot)$
4. Set $y := f(x)$
5. $b \stackrel{?}{=} b'$

$$\Pr[b = b'] = 1/2 + \mathsf{negl}(n)$$

PRFs → multi-message encryption

Ideas for multi-message encryption

- State? (e.g. counter of num msgs)
- Randomness?

Stateful encryption w/ PRFs

- Gen$(1^n) \to k$:
  - Sample an n-bit string at random.

- Enc$(k, m, \mathbf{st}) \to c$:
  1. Interpret $\mathbf{st}$ as number ℓ of messages encrypted so far.
  2. Output $c = F_k(\ell) \oplus m$

- Dec$(k, c, \mathbf{st}) \to m$:
  1. Interpret $\mathbf{st}$ as number ℓ of messages encrypted so far.
  - Output $m = F_k(\ell) \oplus c$

Does this work?

Ans: Yes!

Pros:

- Relies on existing tools
- Generally fast
- No need to run PRF from start!

Cons:

- Must maintain counter of encrypted messages
  - (Just like PRG solution)

Ideas for multi-message encryption

- State? (e.g. counter of num msgs)
- Randomness?

Randomized encryption w/ PRFs

Gen(1^n): Generate a random n-bit key k that defines

$$F_k : \{0,1\}^\ell \to \{0,1\}^m$$

Enc(k, m): Pick a random x and
let the ciphertext c be the pair $(x, y = F_k(x) \oplus m)$

Dec$(k, c = (x, y))$:

Output $F_k(x) \oplus c$

Does this work?

Ans: Yes!

Proof: next

Pros:

- Relies on existing tools
- Generally fast
- No need to run PRF from start!

Cons:

- Need good randomness during encryption

Security of Randomized Encryption

Enc(k, m): Pick a random x and output $(x, y = F_k(x) \oplus m)$

Dec$(k, c = (x, y))$: Output $F_k(x) \oplus c$

- **Proof strategy:** Focusing on 1msg security first
- **We will introduce two new tools:**
  - Indistinguishability of distributions
  - The hybrid lemma/argument

Indistinguishable distributions

Definition: Two distributions X and Y are *computationally indistinguishable* if for every efficient distinguisher

$$\left|\Pr[D(x) = 1 \mid x \leftarrow X] - \Pr[D(y) = 1 \mid y \leftarrow Y]\right| = \mathrm{negl}(n)$$

Denoted by $X \approx Y$

Eg: PRG security says that $X := \{G(x) | x \leftarrow \{0,1\}^n\} \approx Y := \{y | y \leftarrow \{0,1\}^m\}$

Eg: Single msg security says that

$$\{c \leftarrow \mathsf{Enc}(k, m_0) \mid k \leftarrow \mathcal{K}\} \approx \{c \leftarrow \mathsf{Enc}(k, m_1) \mid k \leftarrow \mathcal{K}\}$$

Proof by hybrid argument

Enc(k, m): Pick a random x and output $(x, y = F_k(x) \oplus m)$

Dec$(k, c = (x, y))$: Output $F_k(x) \oplus c$

Single msg security says that the following dists are indistinguishable.

$$\{c \leftarrow \text{Enc}(k, m_0) \mid k \leftarrow \mathcal{K}\} \text{ and } \{c \leftarrow \text{Enc}(k, m_1) \mid k \leftarrow \mathcal{K}\}$$

How to do this? Let's create more (supposedly) indistinguishable distributions:

$H_0 = \{c := (r, m_0 \oplus F_k(r) \mid r \leftarrow \{0,1\}^n; k \leftarrow \mathcal{K}\}$

$\approx$ by PRF security

$H_1 = \{c := (r, m_0 \oplus R(r) \mid r \leftarrow \{0,1\}^n; R \leftarrow \text{Fns}\}$

$\approx$ defn of random fn

$H_2 = \{c := (r, m_0 \oplus r' \mid r \leftarrow \{0,1\}^n; r' \leftarrow \{0,1\}^n\}$

$\approx$ one time pad

$H_3 = \{c := (r, m_1 \oplus r' \mid r \leftarrow \{0,1\}^n; r' \leftarrow \{0,1\}^n\}$

$\approx$ defn of random fn

$H_4 = \{c := (r, m_1 \oplus R(r) \mid r \leftarrow \{0,1\}^n; R \leftarrow \text{Fns}\}$

$\approx$ by PRF security

$H_5 = \{c := (r, m_1 \oplus F_k(r) \mid r \leftarrow \{0,1\}^n; k \leftarrow \mathcal{K}\}$

Hybrid argument

The key steps in a hybrid argument are:

1. Construct a sequence of poly many distributions b/w the two target distributions.
2. Argue that each pair of neighboring distributions are indistinguishable.
3. Conclude that the target distributions are indistinguishable via contradiction:
    A. Assume the target distributions are distinguishable
    B. Must be the case that an intermediate pair of distributions is distinguishable
    C. This contradicts 2 above.

Hybrid argument

B. Must be the case that an intermediate pair of distributions is distinguishable

Lemma: Let $\cancel{p_0, p_1, p_2, \ldots, p_m}$ be advantage of distinguishing $(H_0, H_1), (H_1, H_2), \ldots, (H_{n-1}, H_n)$

If $p_0 - p_m \geq \epsilon$ there is an index i such that $p_i - p_{i+1} \geq \epsilon/m$ **.**

Proof:

$$p_m - p_0 = (p_m - p_{m-1}) + (p_{m-1} - p_{m-2}) + \cdots + (p_1 - p_0) \geq \epsilon$$

At least one of the m terms has to be at least ε/m (averaging).

Security of Randomized Encryption

Enc(k, m): Pick a random x and output $(x, y = F_k(x) \oplus m)$

Dec$(k, c = (x, y))$: Output $F_k(x) \oplus c$

- **Proof strategy:**
  - 1msg security done.
  - What about multi-msg security?

Multi-msg security proof

Can be written as

$$\{(\mathsf{Enc}(k, m_0), \mathsf{Enc}(k, m_1), \ldots, \mathsf{Enc}(k, m_n)) \mid k \leftarrow \mathcal{K}\}$$
$$\approx \{(\mathsf{Enc}(k, m_0'), \mathsf{Enc}(k, m_1'), \ldots, \mathsf{Enc}(k, m_n')) \mid k \leftarrow \mathcal{K}\}$$

How to prove?

Hybrid argument!

$$H_0 = \{(\mathsf{Enc}(k, m_0), \mathsf{Enc}(k, m_1), \ldots, \mathsf{Enc}(k, m_n)) \mid k \leftarrow \mathcal{K}\}$$

$\approx$ single msg security

$$H_1 = \{(\mathsf{Enc}(k, m_0'), \mathsf{Enc}(k, m_1), \ldots, \mathsf{Enc}(k, m_n)) \mid k \leftarrow \mathcal{K}\}$$

$\approx$ single msg security

$$H_2 = \{(\mathsf{Enc}(k, m_0'), \mathsf{Enc}(k, m_1'), \ldots, \mathsf{Enc}(k, m_n)) \mid k \leftarrow \mathcal{K}\}$$

$\approx$ single msg security

$$\ldots$$

$\approx$ single msg security

$$H_{n-1} = \{(\mathsf{Enc}(k, m_0'), \mathsf{Enc}(k, m_1), \ldots, \mathsf{Enc}(k, m_n)) \mid k \leftarrow \mathcal{K}\}$$

$\approx$ single msg security

$$H_n = \{(\mathsf{Enc}(k, m_0'), \mathsf{Enc}(k, m_1'), \ldots, \mathsf{Enc}(k, m_n')) \mid k \leftarrow \mathcal{K}\}$$

So far

Multi-msg security via randomized encryption

Pros:

- Relies on existing tools
- Generally fast
- No need to run PRF from start!

Cons:

- Ciphertext is ~2x larger: $(r, m \oplus F_k(r))$
- Can only encrypt fixed-size n bit msg at a time
- Thus, sending a message of, say, $10n$ bits, requires $20n$-sized ciphertext

Multi-msg security for long msgs

New concept: modes of operation

Ideas?

Recall:

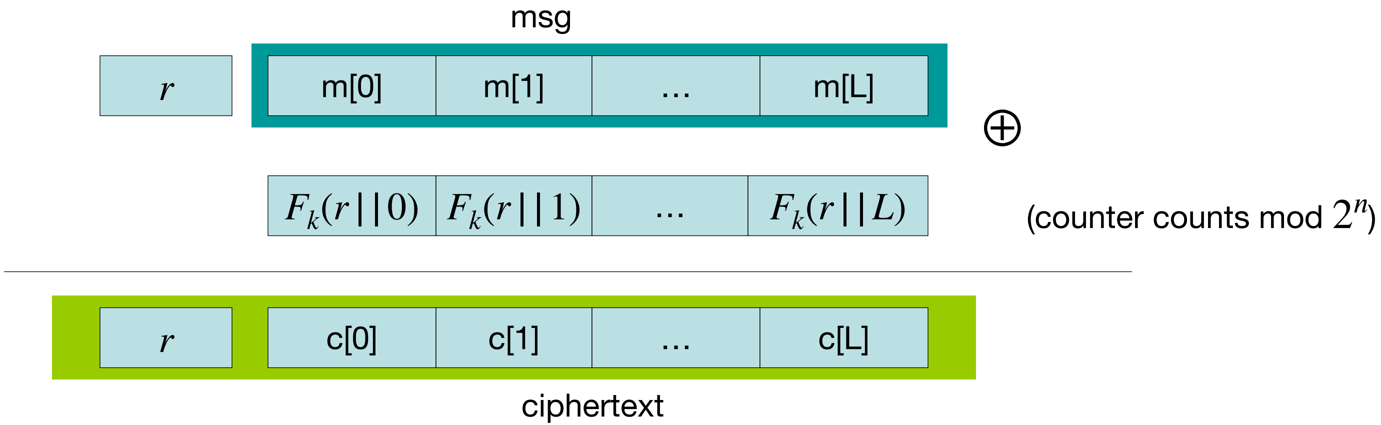
- Counter-based encryption
- Randomized encryption

Can we combine them?

Construction 2: rand ctr-mode

F: PRF defined over (K, X, Y) where $X = \{0,1\}^{2n}$ and $Y = \{0,1\}^n$

(e.g., n=128)



r - chosen at random for every message

note: parallelizable

rand ctr-mode: CPA analysis

Randomized counter mode: random IV.

<u>Counter-mode Theorem</u>: For any L>0,

If F is a secure PRF over (K,X,Y) then

E_{CTR} is IND-CPA-secure.

In particular, for a q-query adversary A attacking E_{CTR}

there exists a PRF adversary B s.t.:

$$\text{Adv}_{\text{CPA}}[A, E_{\text{CTR}}] \leq 2 \cdot \text{Adv}_{\text{PRF}}[B, F] + 2\, q^2 L / |X|$$

<u>Note</u>: ctr-mode only secure as long as $q^2 \cdot L \ll |X|$

Multi-msg security via randomized encryption

Pros:

- Pretty fast
- Ciphertext is ~ (1 + 1/L) larger → small for large L
- Parallelizable!

Cons:

- PRFs somewhat difficult to find, kind of slow

Good for us: Pseudorandom *Permutations* are easier to find!

PRPs and PRFs

- Pseudo Random Function (**PRF**) defined over (K,X,Y):

$$F: K \times X \to Y$$

such that exists "efficient" algorithm to evaluate F(k,x)

- Pseudo Random Permutation (**PRP**) defined over (K,X):

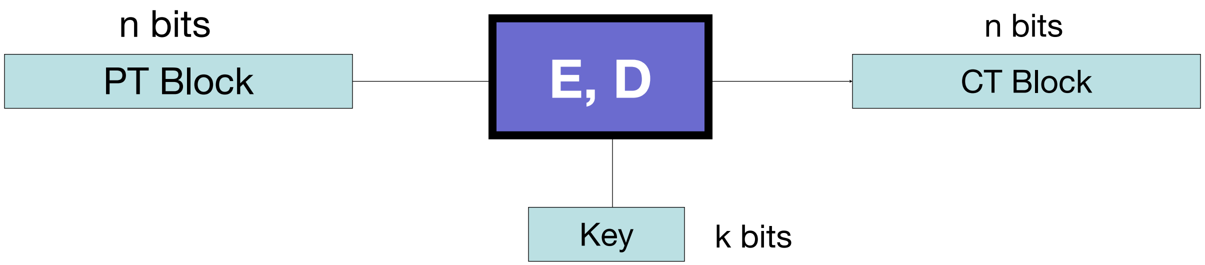
$$E: K \times X \to X$$

such that:

1. Exists "efficient" algorithm to evaluate E(k,x)
2. The function $E(k, \cdot)$ is one-to-one
3. Exists "efficient" inversion algorithm D(k,x)

Also called a Block Cipher

A **block cipher** is a pair of efficient algs. (E, D):



Canonical examples:

1. **AES**: n=128 bits, k = 128, 192, 256 bits
2. **3DES**: n= 64 bits, k = 168 bits (historical)

Running example

- Example PRPs: 3DES, AES, …

  AES128: $K \times X \rightarrow X$ where $K = X = \{0,1\}^{128}$

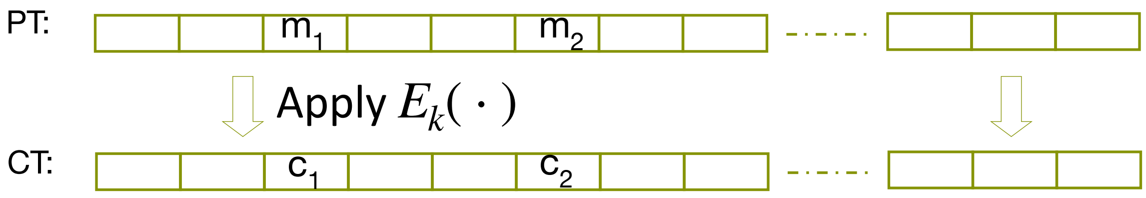
  DES: $K \times X \rightarrow X$ where $X = \{0,1\}^{64}$, $K = \{0,1\}^{56}$

  3DES: $K \times X \rightarrow X$ where $X = \{0,1\}^{64}$, $K = \{0,1\}^{168}$

- Functionally, any PRP where K and X are large is also a PRF.
  - A PRP is a PRF where X=Y and is efficiently invertible

Incorrect use of a PRP

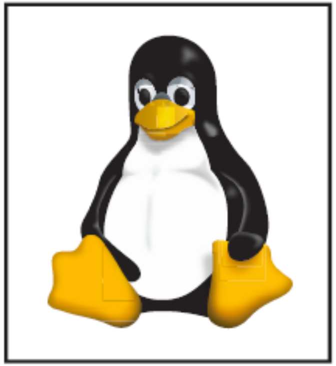
Electronic Code Book (ECB):



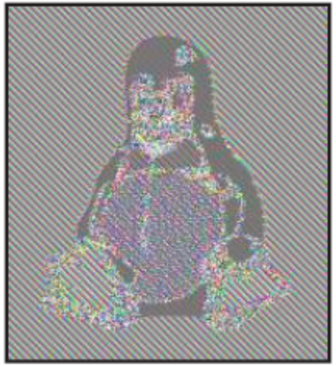
<u>Problem</u>:

– if $m_1 = m_2$ then $c_1 = c_2$

In pictures



Original penguin

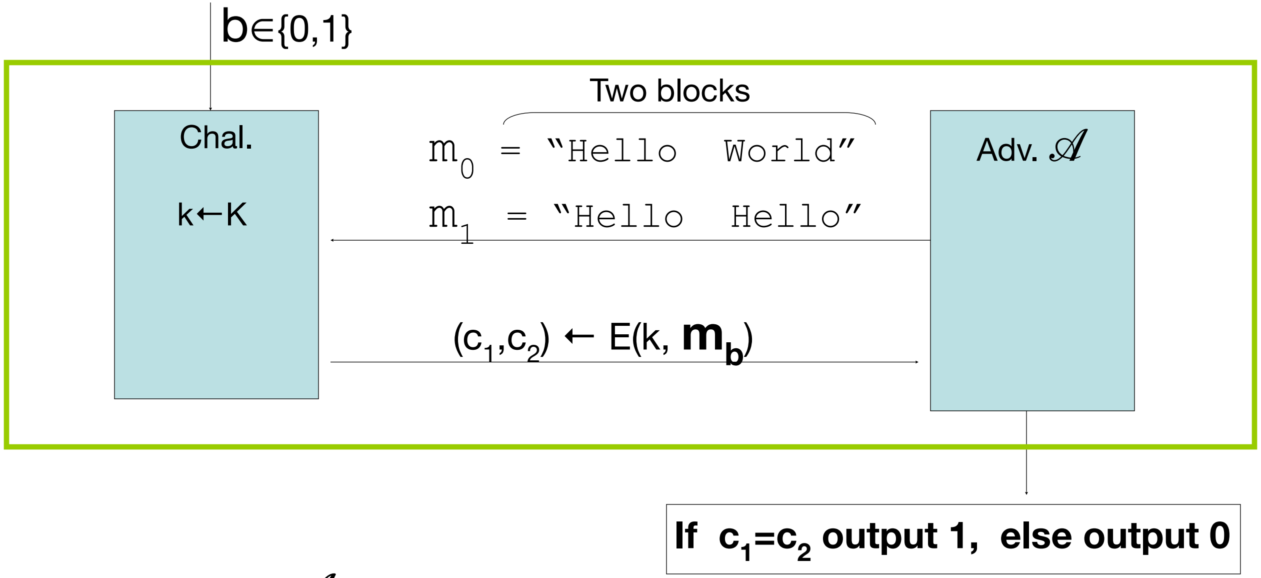


ECB encrypted penguin

(courtesy B. Preneel)

ECB is not Semantically Secure even for 1 msg

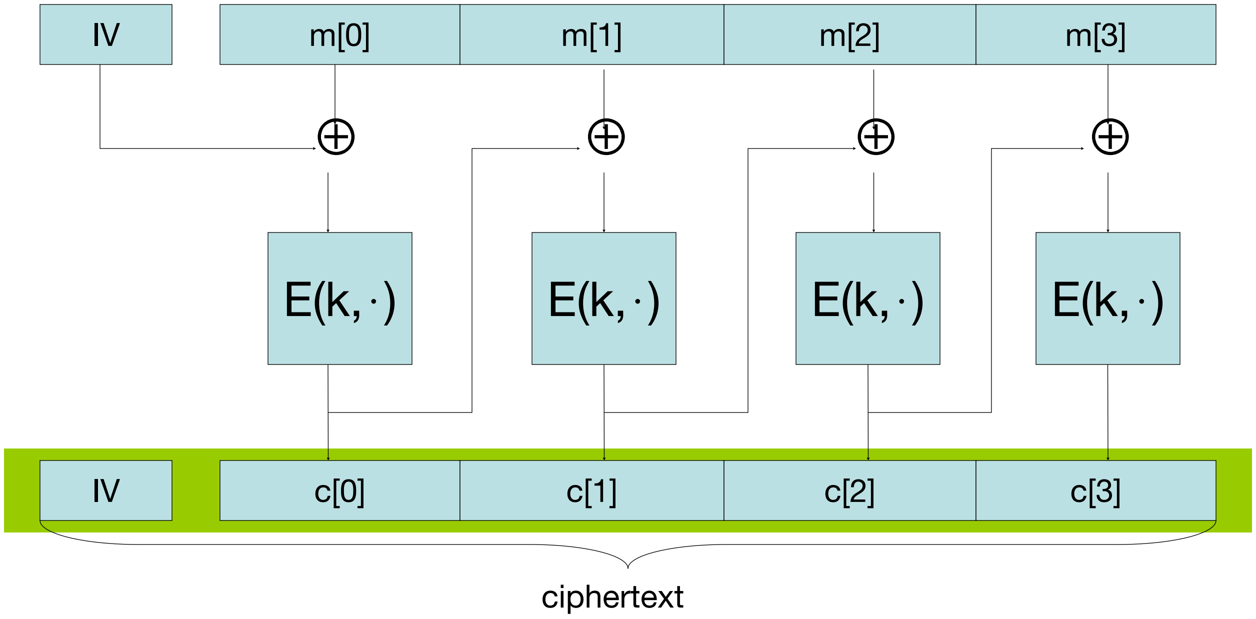
ECB is not semantically secure for messages that contain two or more blocks.



Then $\mathrm{Adv}_{\mathrm{SS}}[\mathscr{A}, \mathrm{ECB}] = 1$

Secure Construction 1: CBC with random nonce

Cipher block chaining with a <u>random</u> IV (IV = nonce)



note: CBC where attacker can predict the IV is not CPA-secure. HW.

CBC: CPA Analysis

<u>CBC Theorem</u>: For any L>0,

If E is a secure PRP over (K,X) then

E_{CBC} is a sem. sec. under CPA over (K, X^L, X^{L+1}).

In particular, for a q-query adversary A attacking E_{CBC}

there exists a PRP adversary B s.t.:

$$\mathrm{Adv}_{\mathrm{CPA}}[A, E_{\mathrm{CBC}}] \leq 2 \cdot \mathrm{Adv}_{\mathrm{PRP}}[B, E] + 2\, q^2 L^2 / |X|$$

Note: CBC is only secure as long as $q^2 \cdot L^2 \ll |X|$

messages enc. with key

max msg length

Next

HW

- Construct PRF from PRG!

Next Class:

- What happens if adversary can tamper with messages?