CIS 5560

Cryptography
Lecture 8

Course website:

pratyushmishra.com/classes/cis-5560-s25/

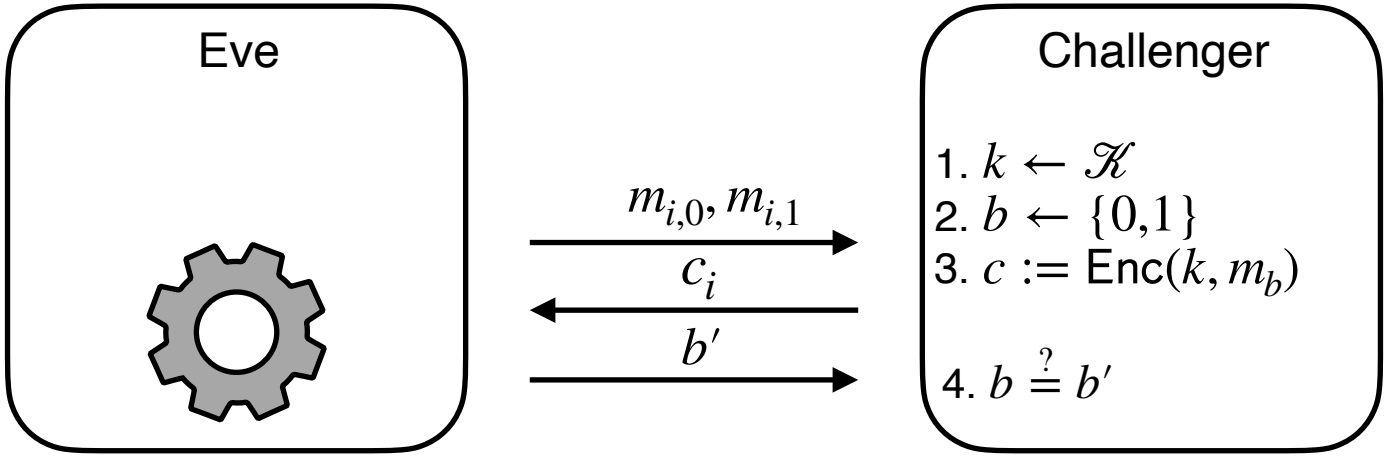
Slides adapted from Dan Boneh and Vinod Vaikuntanathan

Announcements

- **HW 3 out on Wednesday**
  - Due **Friday**, Feb 21 at 5PM on Gradescope
  - Covers PRFs, IND-CPA

Recap of last lecture

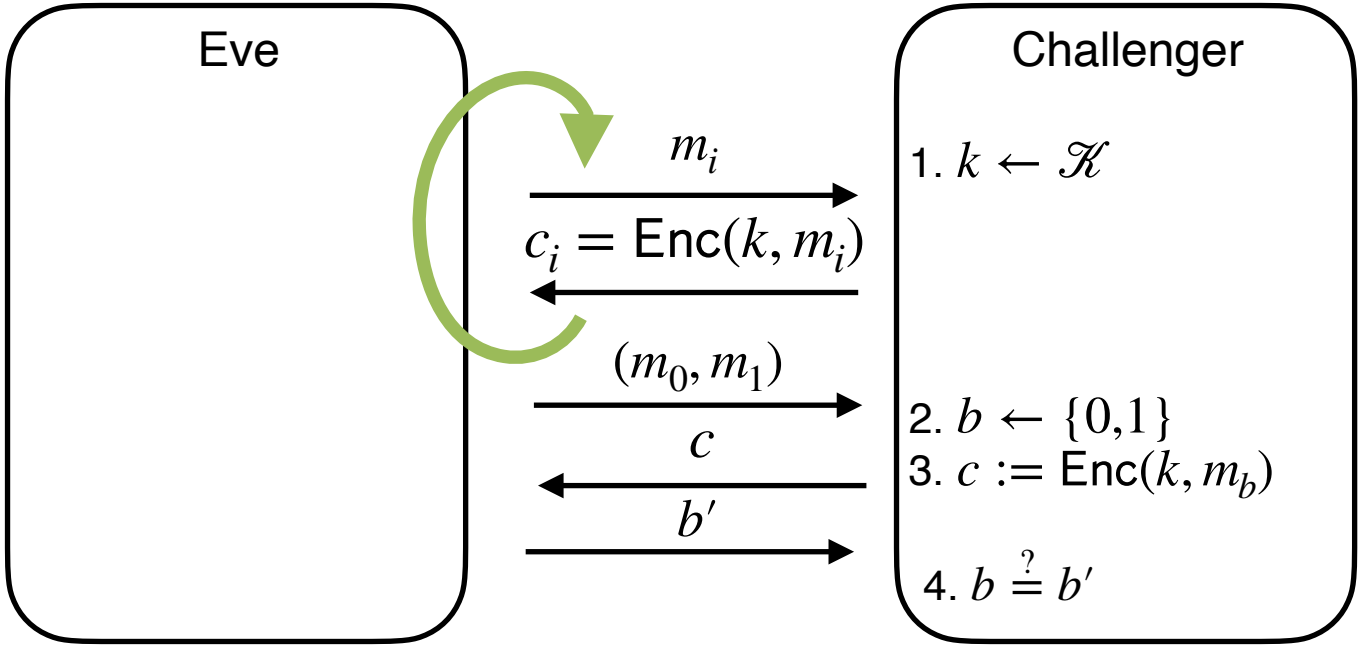
Semantic Security for Many Msgs



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 $m_{i,0} = m_{i,1} = m_i$

Pseudorandom Functions

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

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

Security: Cannot distinguish from random function

$$\left| \Pr\left[A^{f_k}(1^n) = 1 \mid k \leftarrow \{0,1\}^{\ell}\right] - \Pr\left[A^{F}(1^n) = 1 \mid F \leftarrow \mathsf{Fns}\right] \right| \leq \mathsf{negl}(n)\,.$$

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 $F_k(x) \oplus c$

Indistinguishable distributions

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

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

Denoted by $X \approx Y$

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

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.

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

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

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

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

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

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

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

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

Today's Lecture

- Multi-message secure encryption
- Block ciphers, PRPs, encryption for long messages
- PRGs → PRFs

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 $F_k(x) \oplus c$

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? Define $\mathsf{Enc2}(m) = (r, R(r) \oplus m)$ **for a random fn** R**, and** $\mathsf{Enc3}(m) = (r, r' \oplus m)$ **for a random** r'**.**

$$H_0 = \{(\mathsf{Enc}(k, m_0), \ \ldots, \ \mathsf{Enc}(k, m_n)) \mid k \leftarrow \mathcal{K}\}$$
$$H_1 = \{(\mathsf{Enc2}(m_0), \ \ldots, \ \mathsf{Enc2}(m_n)) \mid R \leftarrow \mathsf{Fns}\}$$
$$H_2 = \{(\mathsf{Enc3}(m_0), \ \ldots, \ \mathsf{Enc3}(m_n)) \mid r'_i \leftarrow \{0,1\}^n\}$$
$$H_3 = \{(\mathsf{Enc3}(m'_0), \ \ldots, \ \mathsf{Enc3}(m'_n)) \mid r'_i \leftarrow \{0,1\}^n\}$$
$$H_4 = \{(\mathsf{Enc2}(m'_0), \ \ldots, \ \mathsf{Enc2}(m'_n)) \mid R \leftarrow \mathsf{Fns}\}$$
$$H_n = \{(\mathsf{Enc}(k, m'_0), \ \ldots, \ \mathsf{Enc}(k, m'_n)) \mid k \leftarrow \mathcal{K}\}$$

$\approx$ PRF security

$=$ Defn of random fn

$=$ OTP security

$\approx$ Defn of random fn

$\approx$ PRF security

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:

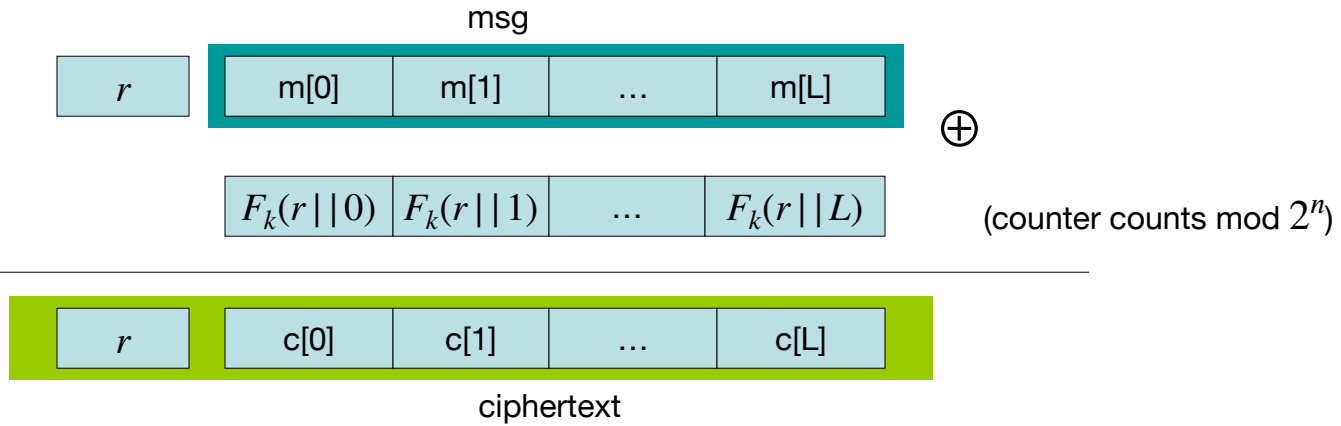
- Counter-based encryption
- Randomized encryption

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.

Counter-mode Theorem: 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.:

$$Adv_{CPA}[A, E_{CTR}] \le 2 \cdot Adv_{PRF}[B, F] + 2\, q^2\, L / |X|$$

Note: 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):

  $F: K \times X \rightarrow Y$

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

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

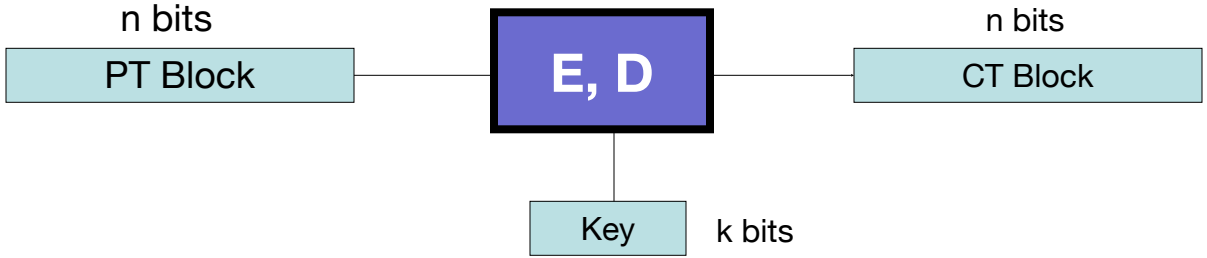
  $E: K \times X \rightarrow X$

  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 $K = X = \{0,1\}^{128}$

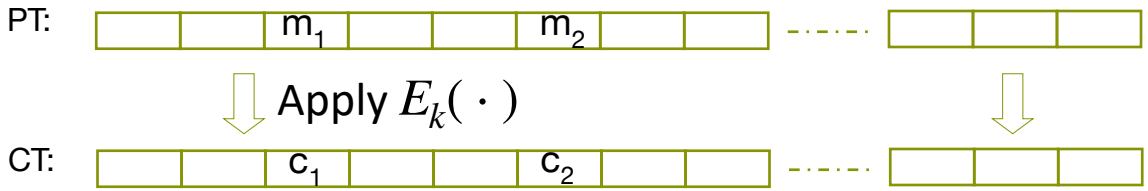
  DES: $K \times X \rightarrow X$ where $X = \{0,1\}^{64}$, $K = \{0,1\}^{56}$

  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

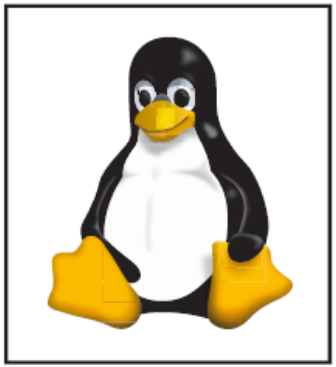
Electronic Code Book (ECB):



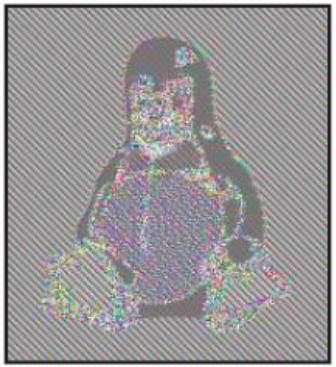
Problem:

- if $m_1=m_2$ then $c_1=c_2$

In pictures



Original penguin

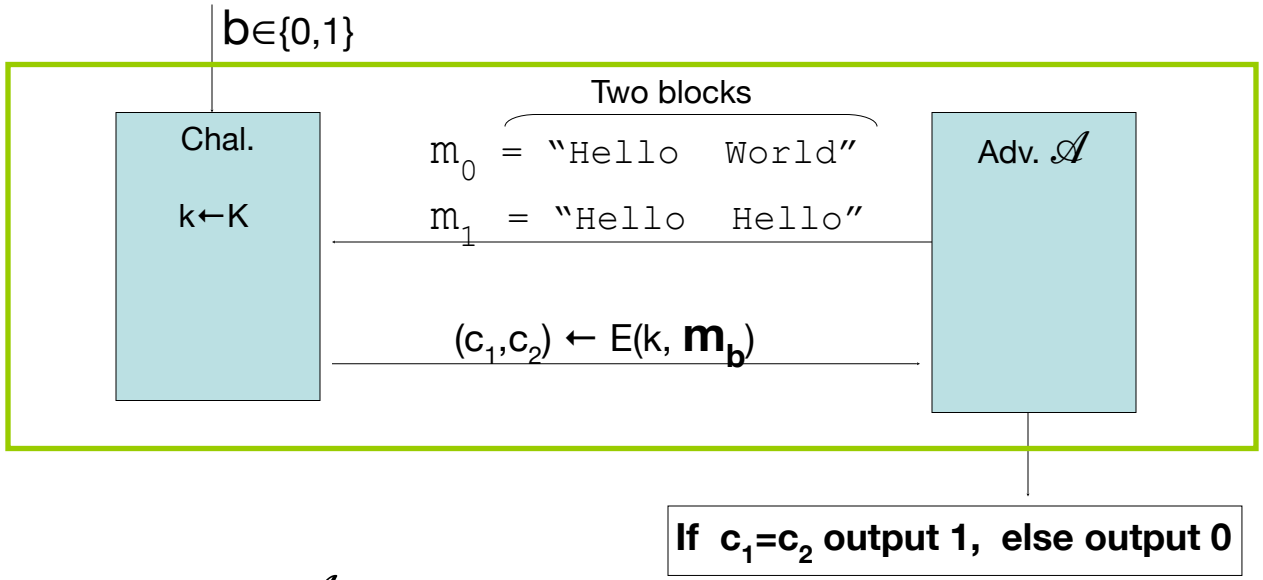


ECB encrypted penguin

(courtesy B. Preneel)

ECB is not Semantically Secure even for 1 msg

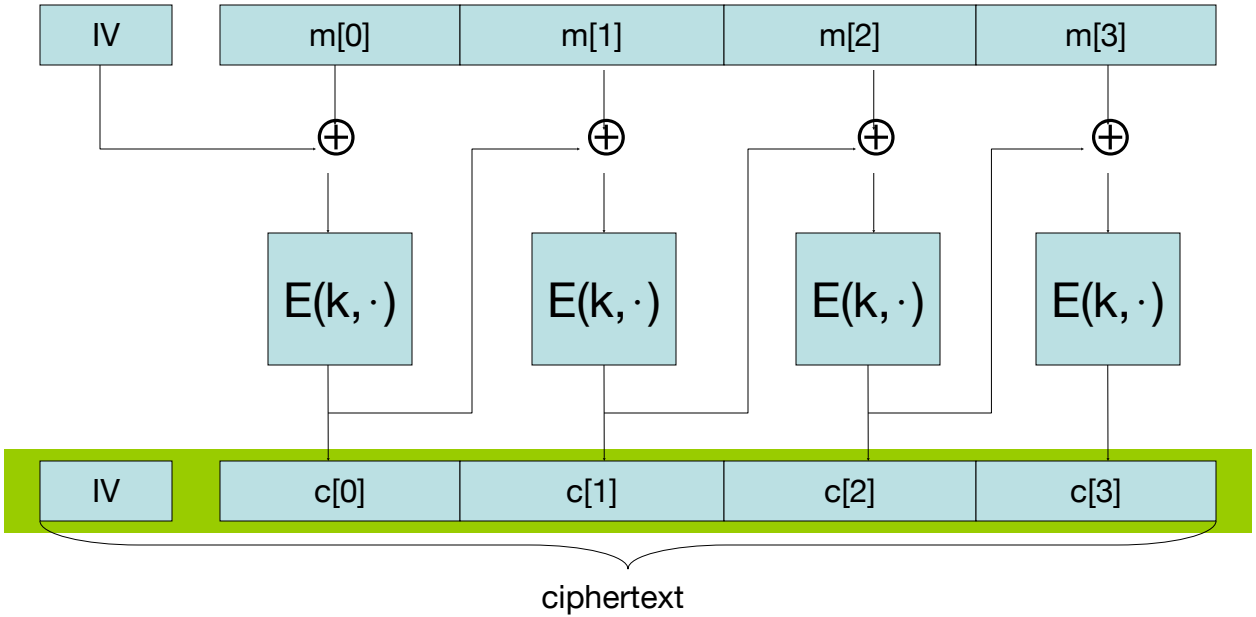
ECB is not semantically secure for messages that contain two or more blocks.



Then $Adv_{SS}[\mathscr{A}, ECB] = 1$

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

CBC Theorem: 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.:

$$Adv_{CPA}[A, E_{CBC}] \le 2 \cdot Adv_{PRP}[B, E] + 2 q^2 L^2 / |X|$$

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

messages enc. with key

max msg length

CBC: CPA Analysis

CBC Theorem: 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.:

$$Adv_{CPA}[A, E_{CBC}] \le 2 \cdot Adv_{PRP}[B, E] + 2 q^2 L^2 / |X|$$

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

messages enc. with key

max msg length

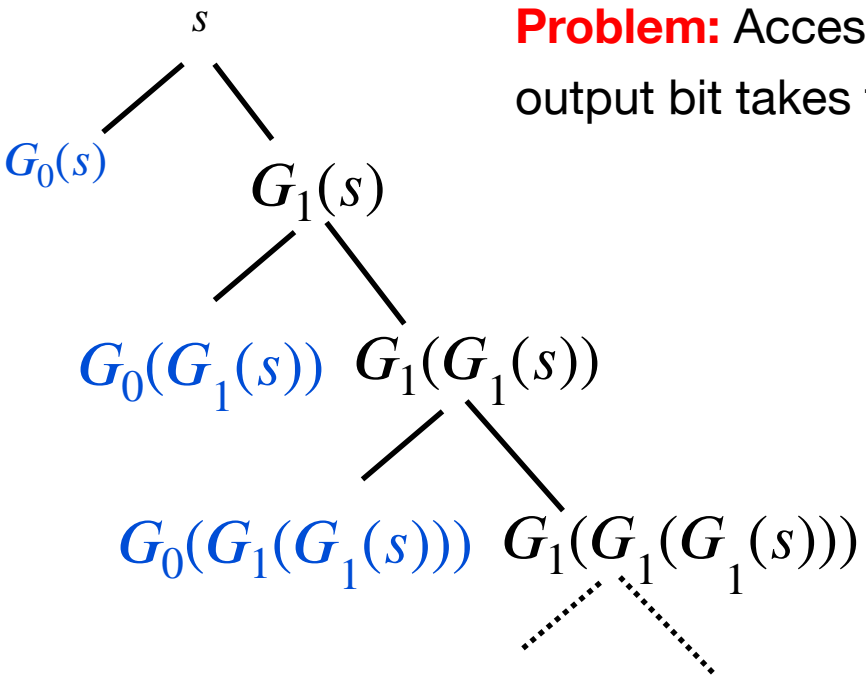
- PRPs and block cipher modes of operation
- PRGs → PRFs
- MACs, if we have time

Let's Look Back at Length Extension…

Theorem: Let G: $\{0,1\}^n \rightarrow \{0,1\}^{n+1}$ be a PRG. Then, for every polynomial m(n), there is a PRG G': $\{0,1\}^n \rightarrow \{0,1\}^{m(n)}$.

Let's Look Back at Length Extension…

Construction: Let G(s) = $G_0(s) \,||\, G_1(s)$ where $G_0(s)$ is 1 bit and $G_1(s)$ is n bits .



Problem: Accessing the i^{th} output bit takes time $\approx i$.

Goldreich-Goldwasser-Micali PRF

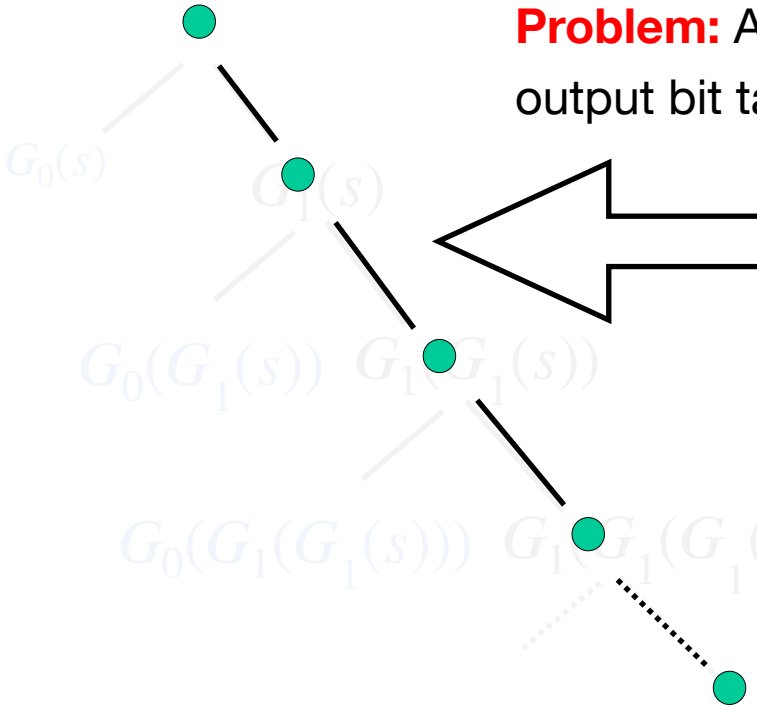
Theorem: Let G be a PRG. Then, for every polynomials $\ell = \ell(n)$, $m = m(n)$, there exists a PRF family $\mathcal{F}_\ell = \{f_s : \{0,1\}^\ell \to \{0,1\}^m\}_{s \in \{0,1\}^n}$.

Note: We will focus on $m = \ell$.
The output length could be made smaller (by truncation) or larger (by expansion with a PRG).

What is the standard way to improve

Let's Look Back at Length Extension…

Construction: Let G(s) = $G_0(s) \,||\, G_1(s)$ where $G_0(s)$ is 1 bit and $G_1(s)$ is n bits .



Problem: Accessing the i^{th} output bit takes time $\approx i$.

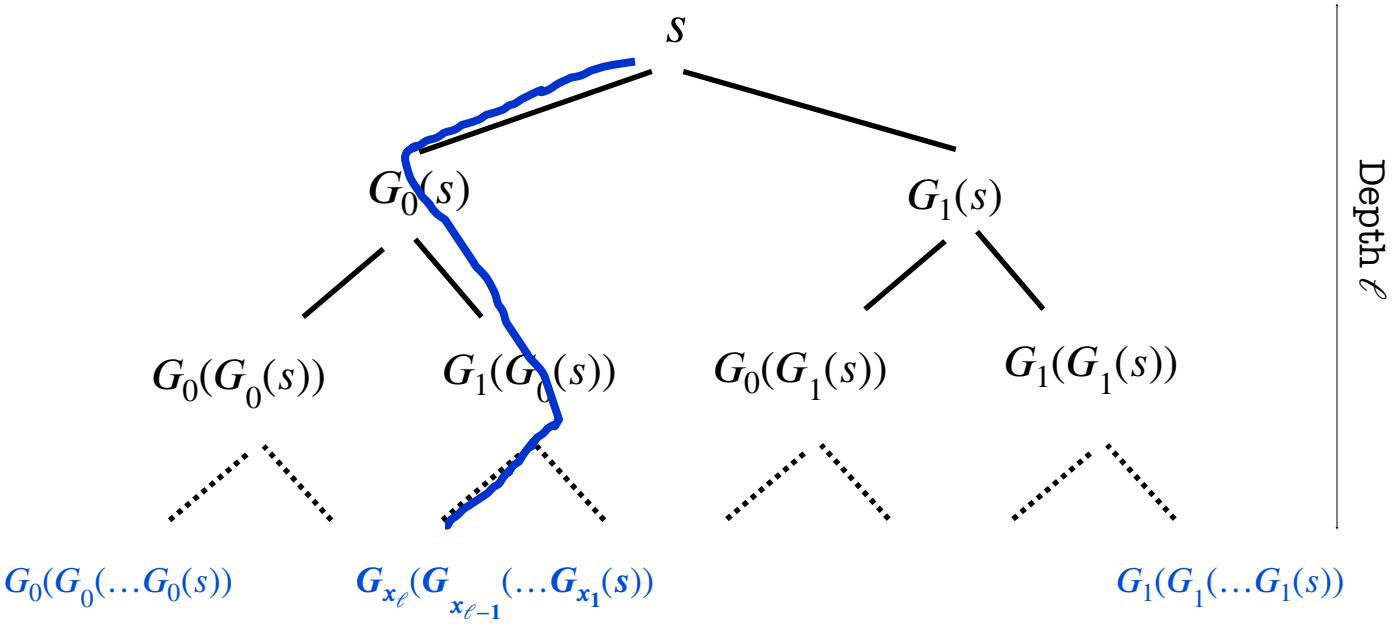
What data structure does this remind you of?

Ans: a list!
No wonder it's linear time!

What is the standard technique to do better?

Goldreich-Goldwasser-Micali PRF

Construction: Let G(s) = $G_0(s)||G_1(s)$ where $G_0(s)$ and $G_1(s)$ are both n bits each.



Each path/leaf labeled by $x \in \{0,1\}^\ell$ corresponds to $f_s(x)$.

Goldreich-Goldwasser-Micali PRF

Construction: Let G(s) = $G_0(s)||G_1(s)$ where $G_0(s)$ and $G_1(s)$ are both n bits each.

The pseudorandom function family $\mathcal{F}_\ell$ is defined by a collection of functions f_s where:

$$f_s(x_1 x_2 \ldots x_\ell) = \boldsymbol{G}_{x_\ell}(\underbrace{\boldsymbol{G}_{x_{\ell-1}}(\ldots \boldsymbol{G}_{x_1}(s))}_{\ell\text{-bit input}})$$

- f_s defines 2^ℓ pseudorandom bits.
- The x^{th} bit can be computed using ℓ evaluations of the PRG G (as opposed to $x \approx 2^\ell$ evaluations as before.)