CS 4770: Cryptography

CS 6750: Cryptography and Communication Security

Alina Oprea
Associate Professor, CCIS
Northeastern University

February 1 2018

Review

- Encryption in practice
  - Block ciphers: PRFs
  - Stream ciphers: PRGs
- PRGs
  - Functions applied to a secret seed that produce output strings indistinguishable from random strings of same length
- PRFs
  - Family of functions (indexed by secret key) that are indistinguishable from random functions
  - Adversary can query inputs and get function outputs
  - Oracle queries (polynomial number)

Encryption in Practice

stream ciphers ≈ pseudorandom generators

block ciphers ≈ pseudorandom functions /permutations

- Practical encryption
  - Good **block ciphers** that withstood the test of time (3DES, AES)
    - Widely used in many practical applications
    - More scrutiny from the community
  - Several recent constructions of **stream ciphers** (eStream)

Cryptographic PRG

outputs:

a random string **r**

or

G(s) (where **s** random)

→

0 if he thinks it's **r**

1 if he thinks it's **G(s)**

Should not be able to distinguish...

Definition

n – a parameter

S – a variable distributed uniformly over $\{0,1\}^n$

r – a variable distributed uniformly over $\{0,1\}^{\ell(n)}$

Definition: **G** is a **cryptographic** **PRG** if for every PPT algorithm **D** we have:

$$| \; P[\, D(G(s)) = 1 \,] - P[\, D(r) = 1 \,] \; |$$

is negligible in **n**.

Scenario $\mathbf{1}$

security parameter
n

distinguisher D

challenger
chooses a random $k \in \{0,1\}^n$.

$x_1 \in \{0,1\}^u$ →

← $F_k(x_1)$

$x_2 \in \{0,1\}^u$ →

← $F_k(x_2)$

⋮

$x_t \in \{0,1\}^u$ →

← $F_k(x_t)$

Scenario 0

security parameter
n

distinguisher D

challenger
chooses a random function
$f : \{0,1\}^u \rightarrow \{0,1\}^v$

$x_1 \in \{0,1\}^u$ →

← $f(x_1)$

$x_2 \in \{0,1\}^u$ →

← $f(x_2)$

⋮

$x_t \in \{0,1\}^u$ →

← $f(x_t)$

outputs **$b \in \{0,1\}$**

Pseudorandom Functions (definition)

- We say that F is a **pseudorandom function (PRF) family** if for all **PPT distinguisher D** the probability to correctly distinguish **scenario 0** from **scenario 1** is **negligible**.

Formally: For all PPT distinguisher **D**:

| Pr[D outputs "1" in scenario 1] – Pr[D outputs "1" in scenario 0] |
is negligible in **n**

$$|Pr[D^{F_k(\cdot)}(n) = 1] - Pr[D^{f(\cdot)}(n) = 1]| \leq \ negl(n)$$

Polynomial number of queries to oracle

An easy application: PRF $\Rightarrow$ PRG

Let $F: K \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF.

Then the following $G: K \to \{0,1\}^{nt}$ is a secure PRG:

$$G(k) = F(k,1) \parallel F(k,2) \parallel \cdots \parallel F(k,t)$$

Key property: parallelizable

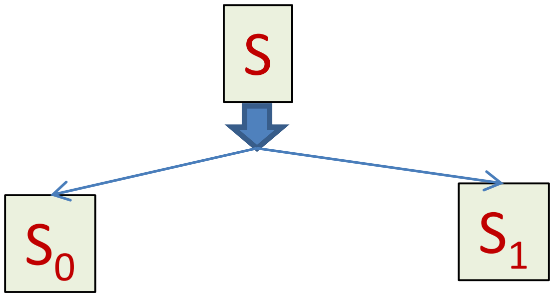
Security from PRF property: $F(k, \cdot)$ indist. from random function $f(\cdot)$

Outline

- Relation between PRF and PRG
  - Construct PRF from PRG (GGM construction)
- Pseudorandom permutations
- Definitions of security for encryption
  - CPA/CCA security
  - Relations between definitions
- CPA-secure construction
  - Security proof
  - Reduction to PRF

Constructing a 1-bit PRF from PRG

- Let $G : \{0,1\}^n \rightarrow \{0,1\}^{2n}$ be a PRG.



$(S_0, S_1) = G(S)$

- Define PRF: $F_s(x) = S_x$

Reduction proof

- Assume, by contradiction, that F is not a secure PRF. There exists a distinguisher D such that:

$$|\Pr[D^{F_k(\cdot)} = 1] - \Pr[D^{f(\cdot)} = 1]| = \epsilon(n)$$

- We build A a distinguisher for G
- A is given access to string $u = u_0||u_1$
  - $u = r$ random in world 0
  - $u = G(s) = s_0||s_1$ in world 1
- A runs D; when D makes a query for bit $x \in \{0,1\}$ A outputs u_x
- A outputs what D outputs

Reduction proof

- Assume, by contradiction, that F is not a secure PRF. There exists a distinguisher D such that:

$$|\Pr[D^{F_k(\cdot)} = 1] - \Pr[D^{f(\cdot)} = 1]| = \epsilon(n)$$

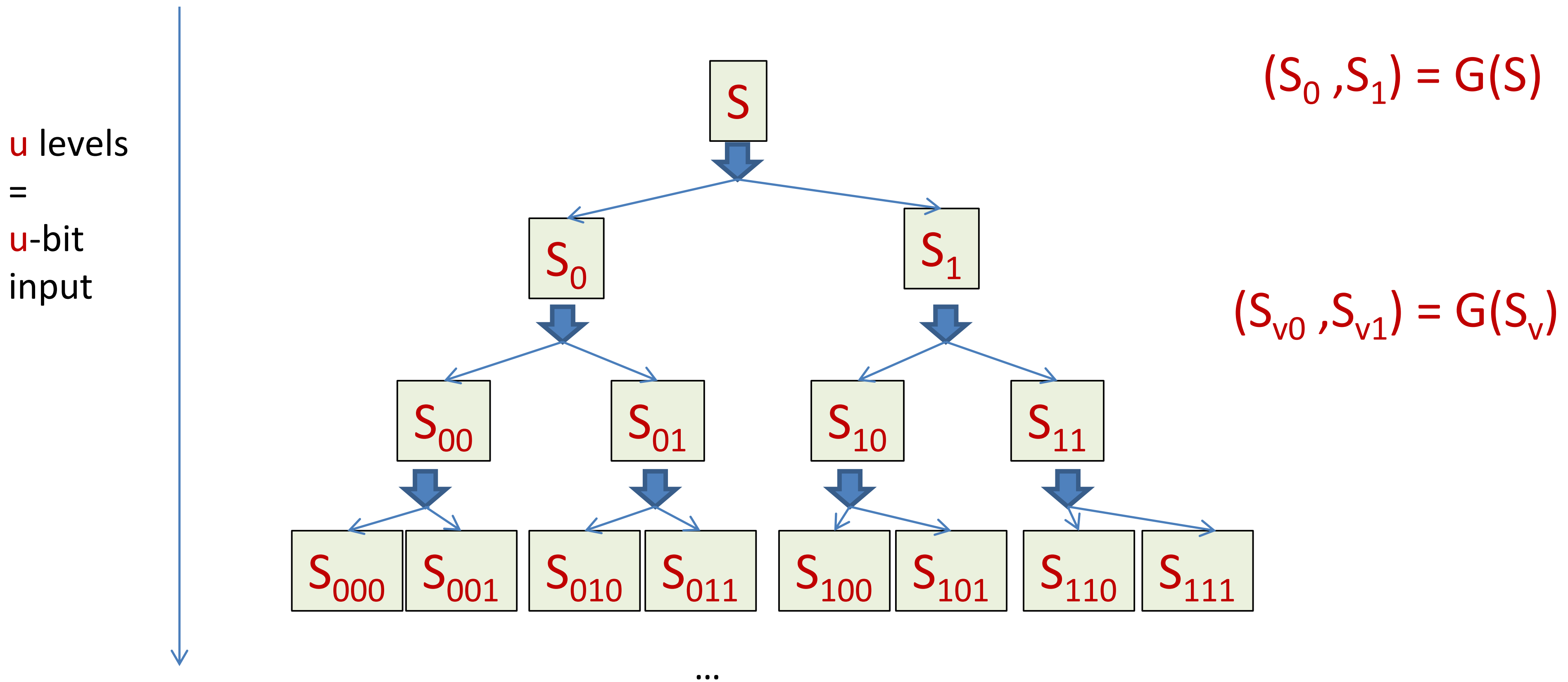
- We build A a distinguisher for G
- In world 0, $\Pr[A(r) = 1] = \Pr[D^{f(\cdot)} = 1]$
- In world 1, $\Pr[A(G(s)) = 1] = \Pr[D^{S_0,S_1} = 1]$
$= \Pr[D^{F_k(\cdot)} = 1]$

$$|\Pr[A(r) = 1] - \Pr[A(G(s)) = 1]| = |\Pr[D^{F_k(\cdot)} = 1] - \Pr[D^{f(\cdot)} = 1]| = \epsilon(n)$$

Constructing a PRF from PRG

[Goldreich-Goldwasser-Micali]

- Let $G : \{0,1\}^n \to \{0,1\}^{2n}$ be a PRG.



- Define PRF: $F_s(x) = S_x$

Pseudorandom Permutations (PRP)

- Sometimes, useful to have a PRF that's also a permutation $F_k(x) : \{0,1\}^u \rightarrow \{0,1\}^u$.
- Can efficiently compute inverse

  $F_k^{-1}(y)$ such that $F_k^{-1}(F_k(x)) = x$.
- Security of PRP: Attacker sees $F_k(x)$ and $F_k^{-1}(y)$ for various values x, y. Cannot distinguish from seeing $R(x), R^{-1}(y)$ for completely random permutation R.

Pseudorandom permutations (definition)

- We say that F is a **pseudorandom function (PRF) family** if for all **PPT distinguisher D** the probability to correctly distinguish **scenario 0** from **scenario 1** is **negligible**.

Formally: For all PPT distinguisher **D**:

| Pr[D outputs "1" in scenario 0] – Pr[D outputs "1" in scenario 1] |

is negligible in **n**

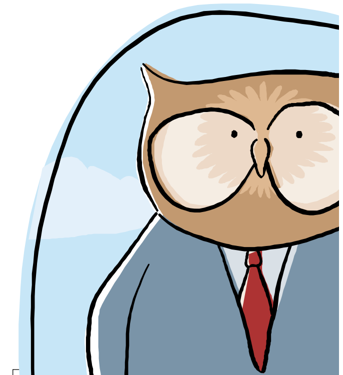
$$|Pr\left[D^{F_k(\cdot),F_k^{-1}(\cdot)}(n)=1\right]-Pr[D^{f(\cdot),f^{-1}(\cdot)}(n)=1]| \leq negl(n)$$

Polynomial number of queries to oracle

Security Game

Π= (Enc,Dec): an encryption scheme

security parameter
n



PPT Adversary A

Challenger

chooses $\mathbf{m_0,m_1}$ such that $\mathbf{|m_0|=|m_1|}$

→ $\mathbf{m_0,m_1}$ →

1. Choose random $\mathbf{k} \leftarrow \mathbf{\{0,1\}^n}$
2. chooses random $\mathbf{b} \leftarrow \mathbf{\{0,1\}}$
3. calculate $\mathbf{c} \leftarrow \mathbf{Enc(k,m_b)}$

← **c** ←

Makes a guess **b'**

Security definition:
We say that **(Enc,Dec)** is **indistinguishable against eavesdropping (EAV-secure)** if any **polynomial time** adversary, $\mathbf{|\ Pr[\ b'=b\ ] - ½\ |}$ is negligible in n.

Ciphertext-only attack

The security definition

- Experiment $\mathrm{Exp}^{\mathrm{EAV}}_{\Pi,A}(n)$:
  1. Choose $k \leftarrow^{R} Gen(n)$
  2. $m_0, m_1 \leftarrow A_1\,(\cdot)$
  3. $b \leftarrow^{R} \{0,1\}; c \leftarrow Enc_k(m_b)$
  4. $b' \leftarrow A_2\,(m_0, m_1, c)$
  5. Output 1 if $b = b'$ and 0 otherwise

We say that **(Enc,Dec)** is **EAV-secure** (secure against eavesdropping) if

For every **PPT** adversary $A = (A_1, A_2)$:

|Pr[$\mathrm{Exp}^{\mathrm{EAV}}_{\Pi,A}(n)$ = 1]- ½ | negligible in n

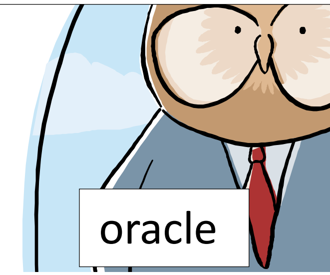
Stronger notions

- CPA security (security against chosen plaintext attacks)
  - Adversary can submit messages and get back ciphertexts
- CCA security (security against chosen ciphertext attacks)
  - Adversary can additionally submit ciphertexts and receive decryptions
  - E.g., find out if ciphertext has valid format

A chosen-plaintext attack (CPA)

security parameter
n

1. selects random **$k \in \{0,1\}^n$**
2. chooses a random **$b = 0,1$**



chooses **m'_1** → **m'_1** →

← **$c_1 = Enc(k,m'_1)$** ←

⋮

chooses **m'_t** → **m'_t** →

← **$c_t = Enc(k, m'_t)$** ←

challenge phase:

chooses **m_0,m_1** → **m_0,m_1** →

← **$c = Enc(k,m_b)$** ←

the interaction continues . . .

makes a guess **b'**

CPA security definition

- **Experiment $\mathrm{Exp}_{\Pi,A}^{\mathrm{CPA}}(n)$:**
  1. Choose $k \leftarrow^R Gen(1^n)$
  2. $m_0, m_1 \leftarrow A_1^{Enc_k(\cdot)}(\cdot)$
  3. $b \leftarrow^R \{0,1\}; c \leftarrow Enc_k(m_b)$
  4. $b' \leftarrow A_2^{Enc_k(\cdot)}(m_0, m_1, c)$
  5. Output 1 if $b = b'$ and 0 otherwise

We say that **(Enc,Dec)** is **chosen-plaintext attack (CPA) secure** if

For every **PPT** adversary $A = (A_1, A_2)$:

$|\Pr[\mathrm{Exp}_{\Pi,A}^{\mathrm{CPA}}(n) = 1] - ½ |$ negligible in n

CCA security definition

- Experiment $\text{Exp}_{\Pi,A}^{\text{CCA}}(n)$:
  1. Choose $k \leftarrow^{R} Gen(1^n)$
  2. $m_0, m_1 \leftarrow A_1^{Enc_k(\cdot),Dec_k(\cdot)}(\cdot)$
  3. $b \leftarrow^{R} \{0,1\}; c \leftarrow Enc_k(m_b)$
  4. $b' \leftarrow A_2^{Enc_k(\cdot),Dec_k(\cdot)}(m_0, m_1, c)$
  5. Output 1 if $b = b'$ and 0 otherwise

Adversary can not submit c to decryption oracle

We say that **(Enc,Dec)** is **chosen-ciphertext attack (CCA) secure** if

For every **PPT** adversary $A = (A_1, A_2)$:

|Pr[$\text{Exp}_{\Pi,A}^{\text{CCA}}(n)$ = 1]- ½ | negligible in n

Relation between security notions

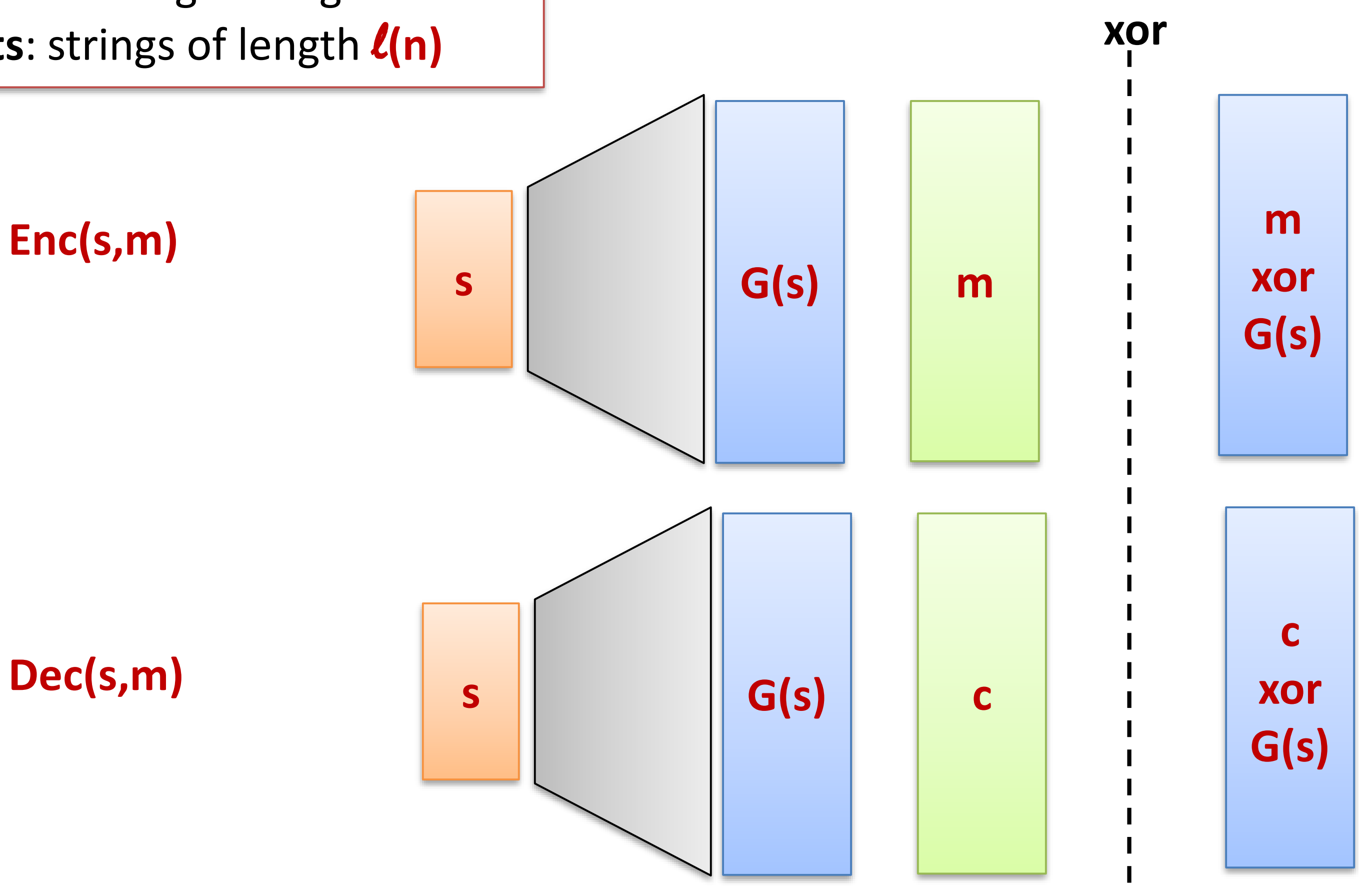
- CPA security implies EAV security
- CCA security implies CPA security
- EAV security does not imply CPA security
  - Will see an example soon

CPA security strictly stronger than EAV security
CCA security strictly stronger than CPA security

EAV-secure encryption from PRG

Use PRGs to "shorten" the key in the one time pad

Key: random string of length **n**
Plaintexts: strings of length ***ℓ*(n)**



Is it CPA secure?

CPA Security Requires Randomness

- **Theorem:** Any CPA secure encryption scheme has to either:
  - Keep state (encryption changes the key).
  - Have a randomized encryption procedure (for a fixed k, m the output of Enc(k,m) cannot be deterministic).

- Why?
  - Otherwise, easy to tell if the same message is encrypted twice!

https://xkcd.com/257/

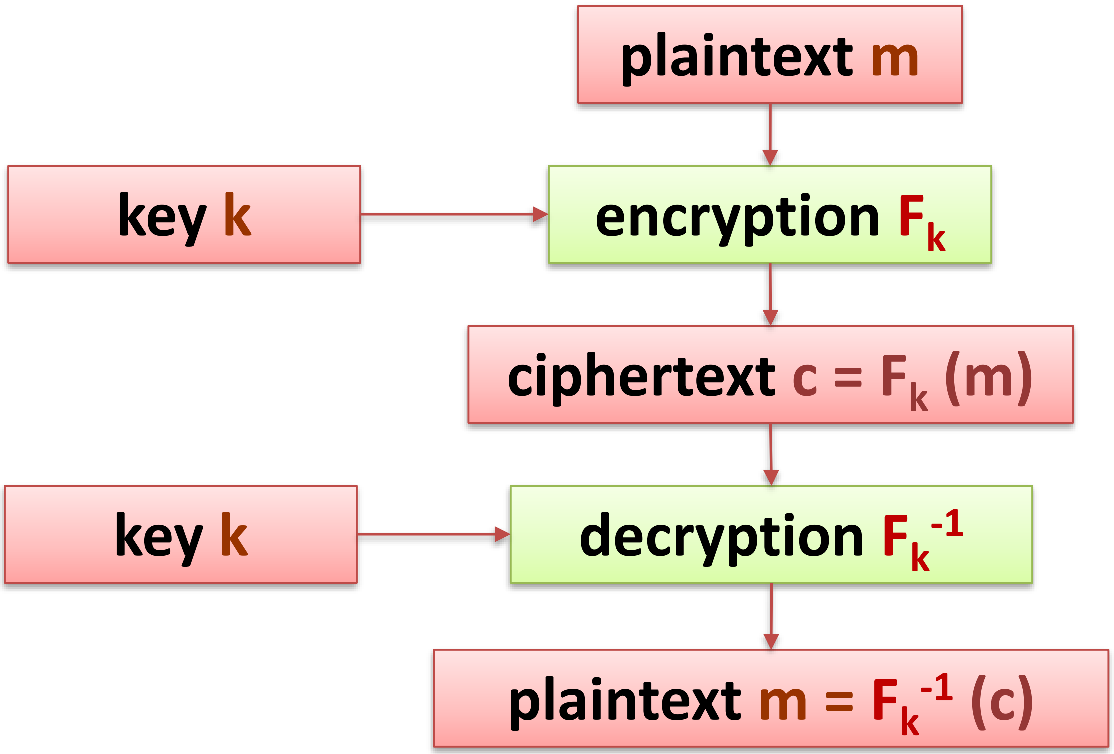
Code Talkers

|< < Prev Random Next > >|

|< < Prev Random Next > >|

How to encrypt using PRF/PRP?

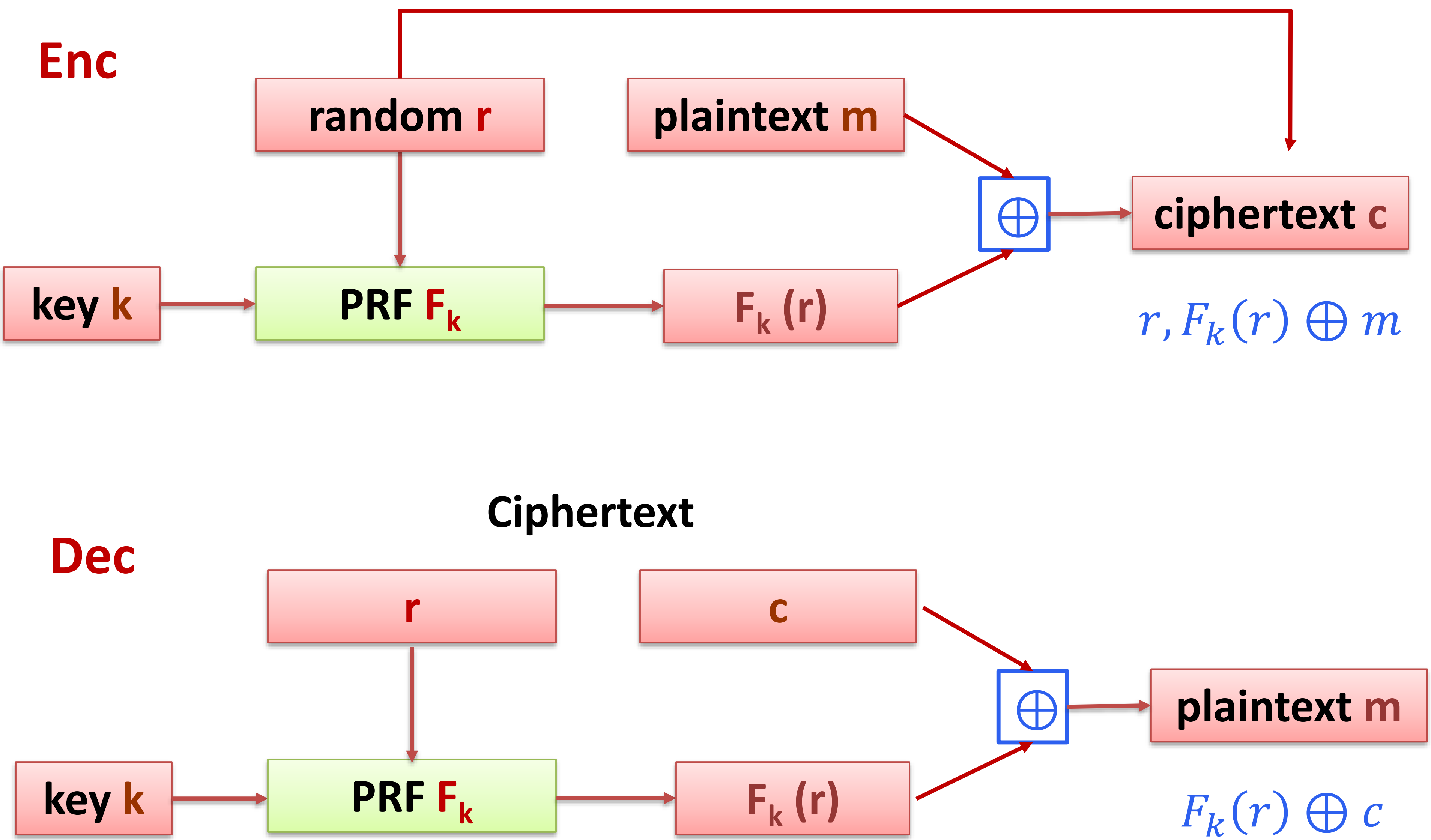
A naive idea:



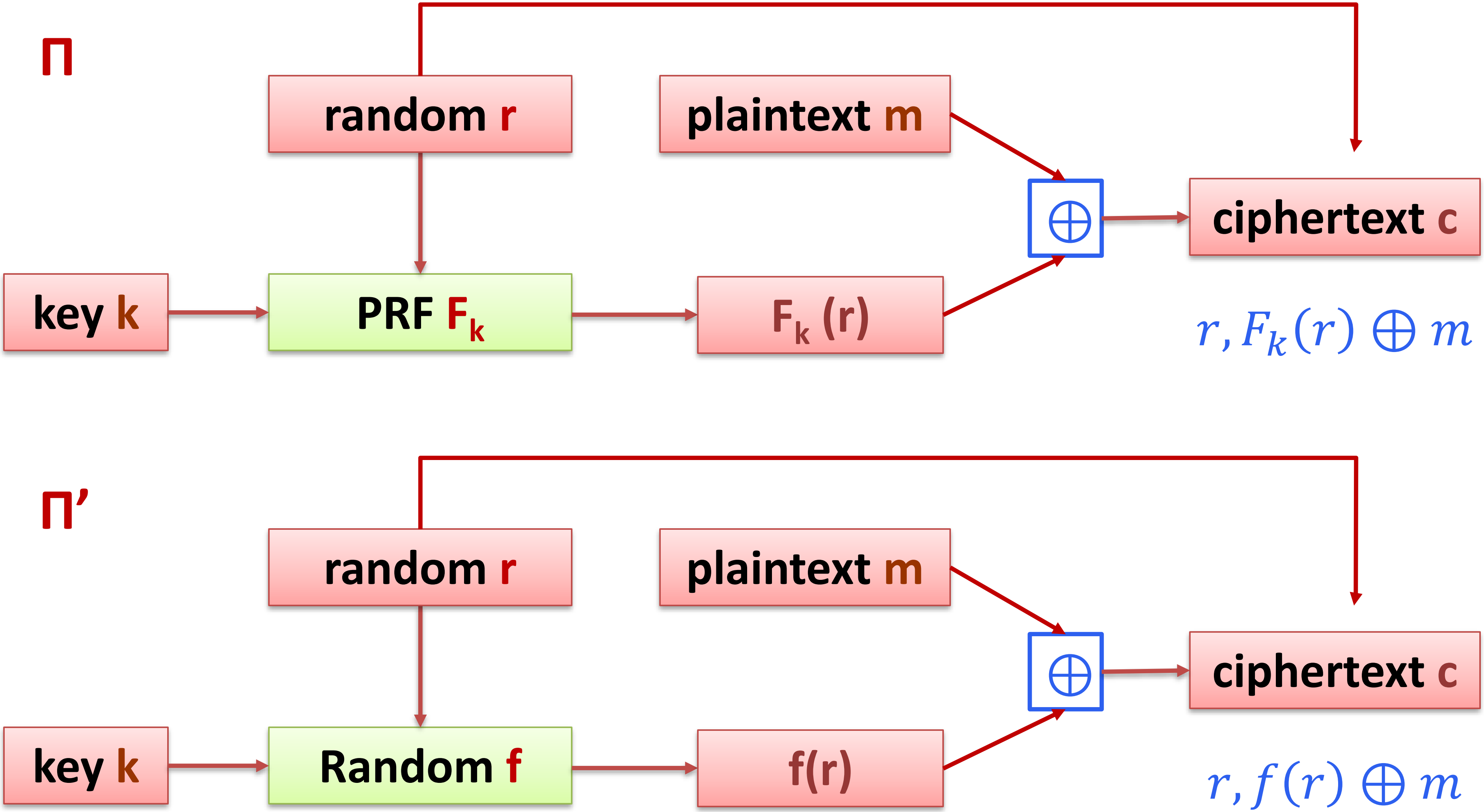
Problems:

1. it is **deterministic** and **has no state**, so it **cannot be CPA-secure**.
2. the messages have to be short

How to encrypt using PRF?



Proof of security - Intuition



Proof of security - Intuition

Π

Enc

$c = (r, F_k(r) \oplus m)$

Dec

$c = (r, \mathrm{s})$

$m = F_k(r) \oplus s$

1. Success of adversary to break **Π** and **Π'** in CPA game is similar

Under the assumption that F is a PRF!

Π'

Enc

$c = (r, f(r) \oplus m)$

Dec

$c = (r, \mathrm{s})$

$m = f(r) \oplus s$

2. Success of adversary to break **Π'** in CPA game is negligible

Proof of security – step 1

1. Success of adversary to break **Π** and **Π’** in CPA game is similar

Assume that F is PRF.

For any PPT adversary A that makes q(n) encryption queries:

$$\left|\mathbf{Pr}[\mathrm{Exp}_{\Pi,A}^{\mathrm{CPA}}(n) = \mathbf{1}] - \mathbf{Pr}[\mathrm{Exp}_{\Pi',A}^{\mathrm{CPA}}(n) = \mathbf{1}]\right| \leq \mathrm{negl(n)}$$

- Let A be a PPT adversary in CPA game for Π st

$$\left|\mathbf{Pr}[\mathrm{Exp}_{\Pi,A}^{\mathrm{CPA}}(n) = \mathbf{1}] - \mathbf{Pr}[\mathrm{Exp}_{\Pi',A}^{\mathrm{CPA}}(n) = \mathbf{1}]\right| = \epsilon(\mathrm{n})$$

and $\epsilon(\mathrm{n})$ is non-negligible

- We build D a distinguisher for PRF
- D is given access to oracle O (in world 0: $O = F_k(\cdot)$ and in world 1: $O = f(\cdot)$)

Proof of security – step 1

1. Success of adversary to break **Π** and **Π’** in CPA game is similar

Assume that F is PRF.

For any PPT adversary A that makes q(n) encryption queries:

$$\left|\mathbf{Pr}[\mathrm{Exp}_{\Pi,A}^{\mathrm{CPA}}(n) = 1] - \mathbf{Pr}[\mathrm{Exp}_{\Pi',A}^{\mathrm{CPA}}(n) = 1]\right| \leq \mathrm{negl}(n)$$

- When A queries Enc oracle with message m, D outputs $c = (r, O(r) \oplus m)$
- When A chooses 2 messages m_0, m_1, D chooses $b \leftarrow \{0,1\}$ and responds with $c = (r, O(r) \oplus m_b)$
- D outputs what A outputs

Proof of security – step 1

1. Success of adversary to break **Π** and **Π'** in CPA game is similar

Assume that F is PRF.

For any PPT adversary A that makes q(n) encryption queries:

$$|\Pr[\mathrm{Exp}^{\mathrm{CPA}}_{\Pi,A}(n) = 1] - \Pr[\mathrm{Exp}^{\mathrm{CPA}}_{\Pi',A}(n) = 1]| \leq \mathrm{negl}(n)$$

- In world 1

$$\Pr[D^{F_k(\cdot)}(n) = 1] = \Pr[\mathrm{Exp}^{\mathrm{CPA}}_{\Pi,A}(n) = 1]$$

- In world 0

$$\Pr[D^{f(\cdot)}(n) = 1] = \Pr[\mathrm{Exp}^{\mathrm{CPA}}_{\Pi',A}(n) = 1]$$

$$|\Pr[D^{F_k(\cdot)}(n) = 1] - \Pr[D^{f(\cdot)}(n) = 1]| =$$

$$|\Pr[\mathrm{Exp}^{\mathrm{CPA}}_{\Pi,A}(n) = 1] - \Pr[\mathrm{Exp}^{\mathrm{CPA}}_{\Pi',A}(n) = 1]| = \epsilon(\mathbf{n})$$

Key takeaways

- Stronger notions of security for encryption
  - CPA security strictly stronger than EAV security
  - CCA security strictly stronger than CPA security
- CPA-secure encryption needs to be randomized
- CPA-secure construction from PRF F
  - Works for small messages
  - Expands the ciphertext by a factor of 2
  - Will discuss how to expand to longer messages with minimal ciphertext expansion

Acknowledgement

Some of the slides and slide contents are taken from
http://www.crypto.edu.pl/Dziembowski/teaching
and fall under the following:

We have also used slides from Prof. Dan Boneh online cryptography course at Stanford University:

http://crypto.stanford.edu/~dabo/courses/OnlineCrypto/