CSCI 531:
Applied Cryptography

Block Ciphers: Modes of Operation

Prof. Tatyana Ryutov

Lecture Outline

- Review
- Using block ciphers
  - Security for many-time key
  - Modes of operation: many time key (CBC mode)
  - Modes of operation: many time key (CTR mode)
- Message Integrity
  - MAC security

Reminders

- HW2 is assigned
  - Due February 22, 11:59pm
- Programming assignment 1 is assigned
  - Due March 1, 11:59pm
- Please post your HW- and programming assignment-related questions on Piazza
- If you need help with the assignments, talk to our CPs

Presentation2

NEURO CRYPTANALYSIS

of DES and 3DES

An Introduction to Neural Network-Based Cryptanalytic Techniques

Ajit Dinkar Bhandarkar
Balaji Betadur

Presentation3

The Dining Cryptographers Problem

An Introduction to Anonymous Communication

Presenters: Yiwen Cai, Aayush Bothra

USC Viterbi School of Engineering

Your Review Topics

- PRF and PRP
- Confusion and diffusion
- Feistel network and its inverse
- 3DES
- The meet in the middle attack

Abstractions: PRPs and PRFs

*(precisely captures what a **block cipher** is)*

- To understand how to use block ciphers correctly, we need a clear abstraction of a block cipher
- 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):

  $E: K \times X \rightarrow X$ accurately captures what a block cipher is

  such that:

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

fix the key k

Secure PRFs

*(precisely captures what a **secure** cipher is)*

- Goal of PRF is to look like a **random function** form X to Y

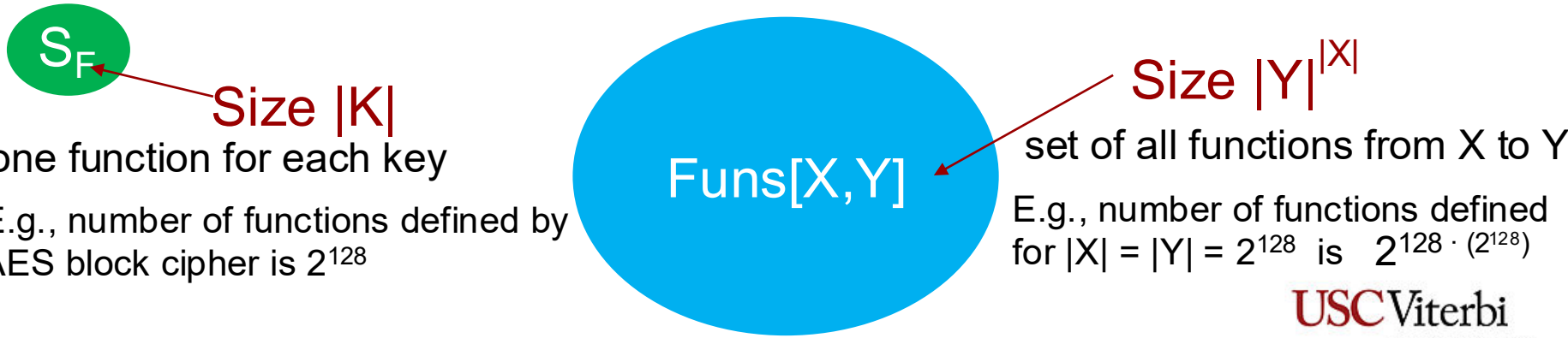
- Let $F: K \times X \rightarrow Y$ be a PRF

  Funs[X,Y]: the set of all functions from X to Y

  $S_F = \{F(k,\cdot): k \in K\} \subseteq$ Funs[X,Y] — the set of all functions from X to Y that are specified by the PRF S_F for particular key k

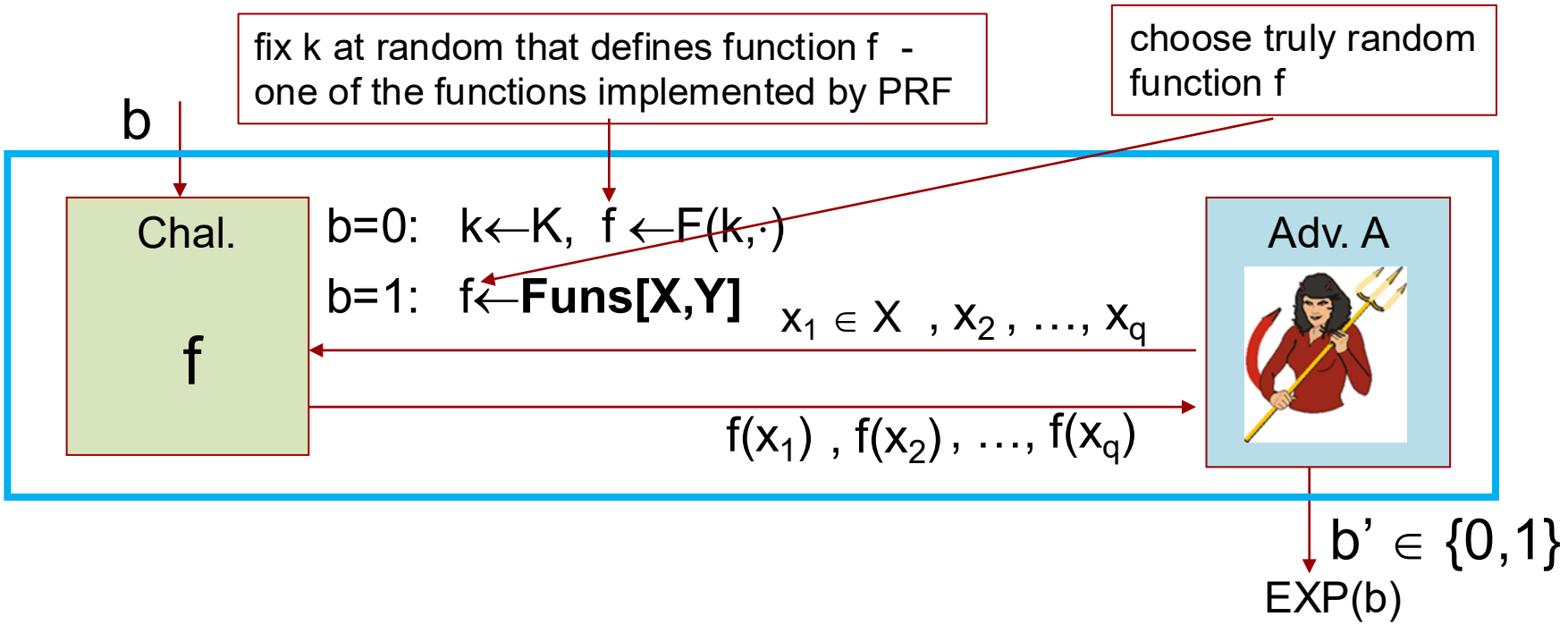
once we fix the key k, we obtain a function from X to Y

- Intuition: a PRF is secure if a random function in Funs[X,Y] is indistinguishable from a pseudorandom function in S_F



Secure PRF: Definition

- For b = 0,1 define experiment EXP(b) as:



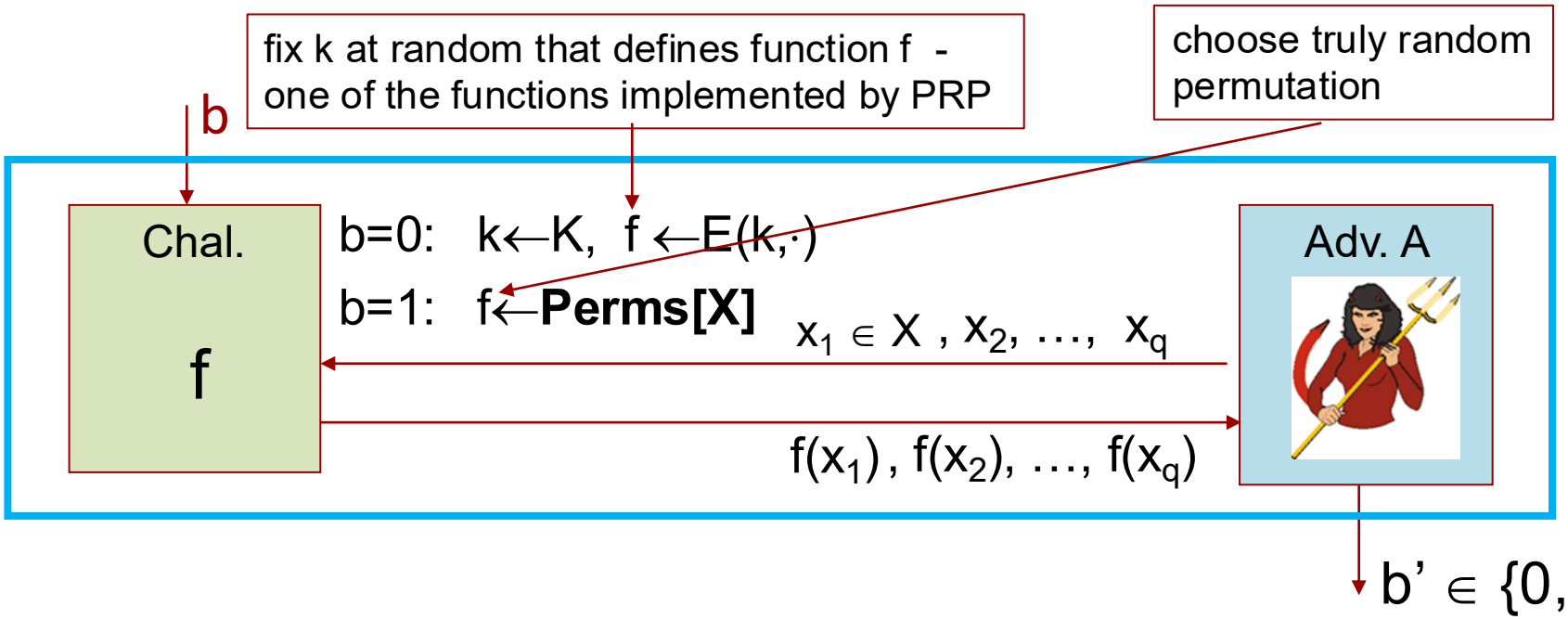
- Def: F is a secure PRF if for all "efficient" A:

$$Adv_{PRF}[A,F] := |Pr[EXP(0) = 1] - Pr[EXP(1) = 1]|$$

is "negligible"

Secure PRP: Definition
(secure block cipher)

- For b = 0,1 define experiment EXP(b) as:



- Def: E is a secure PRP if for all "efficient" A:

$$Adv_{PRP}[A,E] = \left|Pr[EXP(0) = 1] - Pr[EXP(1) = 1]\right|$$

is "negligible"

Application: PRF ⇒ PRG

(pseudorandom functions directly give us a pseudorandom generator)

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

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

$$G(k) = F(k,0) \,||\, F(k,1) \,||\, \cdots \,||\, F(k,t-1)$$

Key property: parallelizable stream cipher

Security follows from PRF property: F(k, ·) is indistinguishable from truly random function f(·)

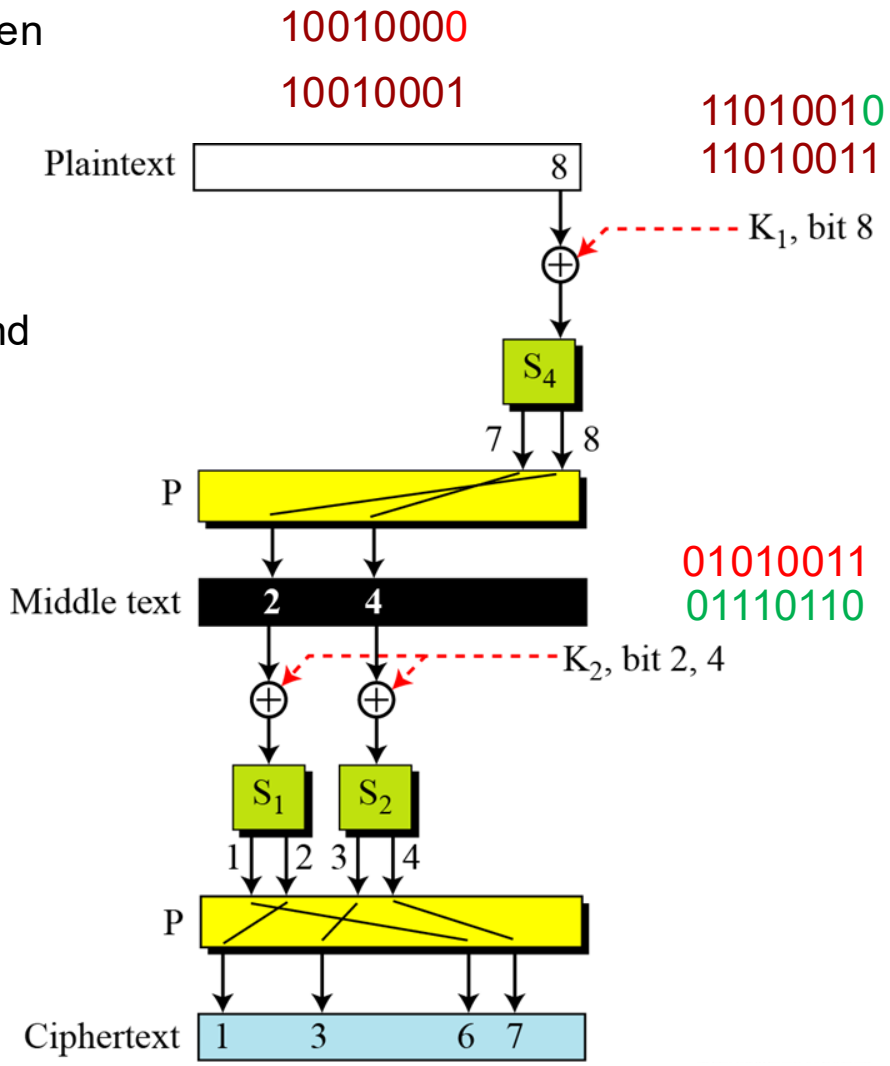
$$G(k) = f(0) \,||\, f(1) \,||\, \cdots \,||\, f(t)$$

Student's question: If an attacker sees the output for inputs 0, 1, and 2, then it would be reasonable to say the next input will be 3. This seems like it would violate unpredictability somehow, but knowing the next input (3) is useless without the key, right? Is it accurate to say that unpredictability comes from the PRP security and knowing the counter doesn't matter at all ?

Confusion and Diffusion

- Confusion (non –linearity)
  - Goal: Hide mathematical relationship between ciphertext and key
  - How?
    - S-boxes mask the key-ciphertext relationship
    - Break mathematical relationship
- Diffusion (spreading)
  - Goal: Hide relationship between plaintext and ciphertext
  - How?
    - P-boxes spread plaintext statistics
    - Avalanche effect

Student's question: why would confusion and diffusion work not as ad hoc constructions?

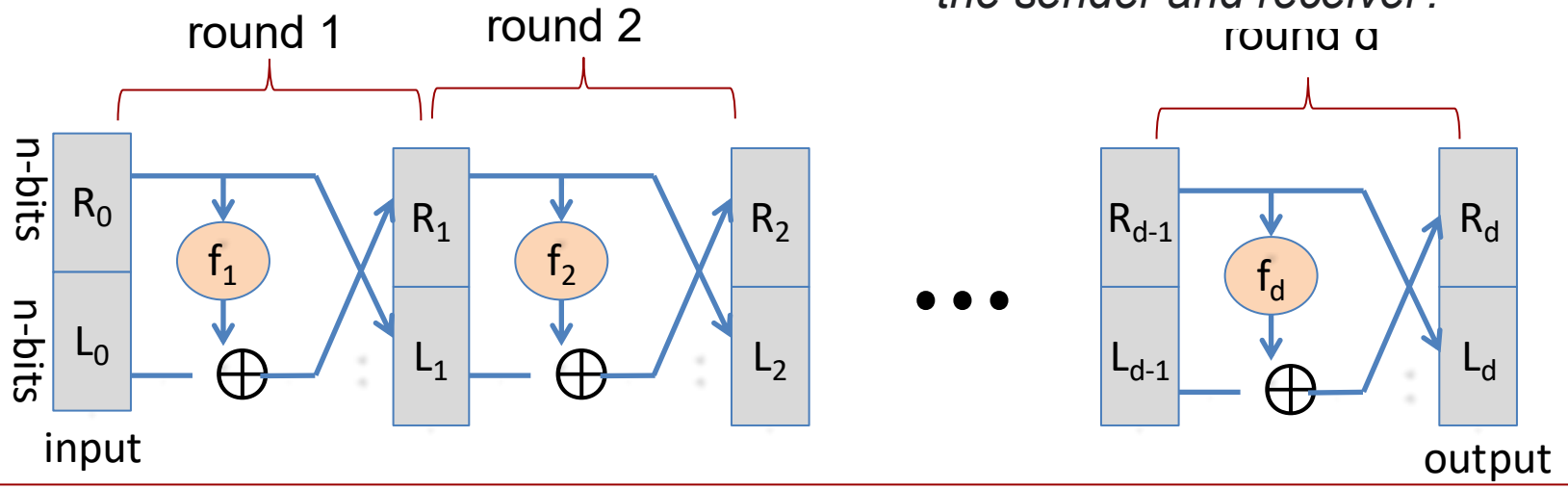


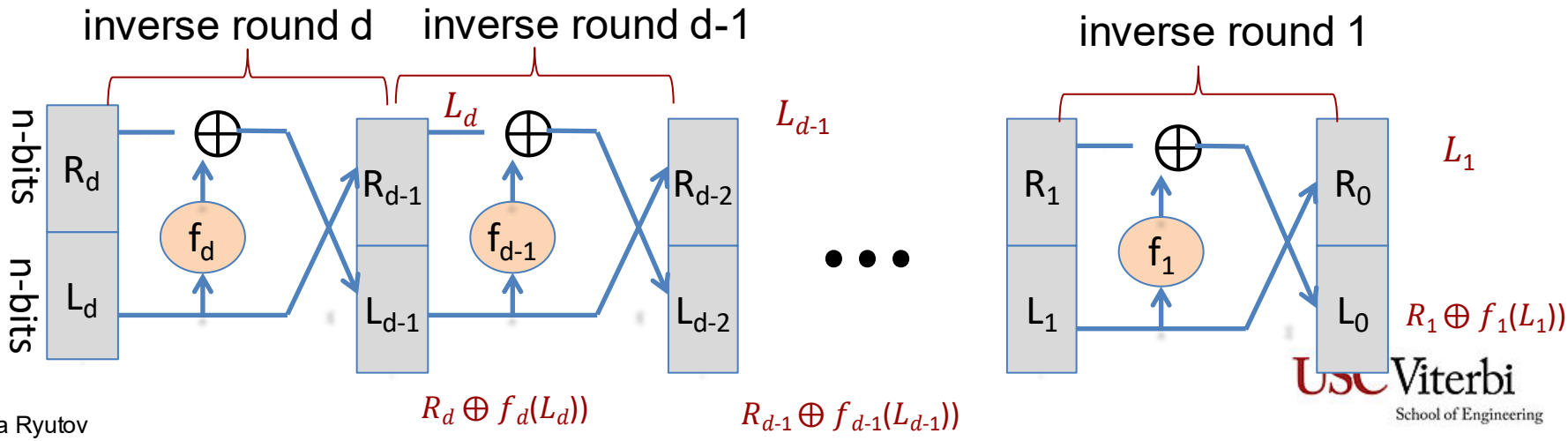
Feistel Network

Given arbitrary functions $f_1, \ldots, f_d$: $\{0,1\}^n$

Goal: build invertible function $F: \{0,1\}^{2n}$

Student's question: how would the functions be transferred between the sender and receiver?

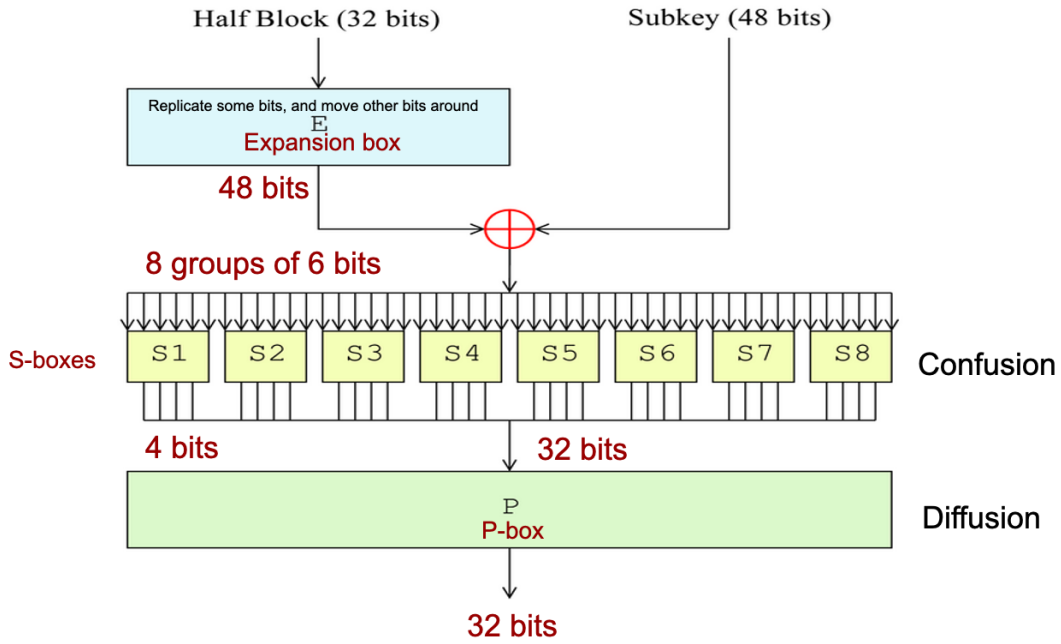




USC Viterbi School of Engineering

Students' Questions

- How confusion and diffusion is achieved in Feistel cipher?
- If 3 rounds of Feistel can guarantee randomness, why is DES using 16 rounds of Feistel in the implementation?
- Why was it important that the s-box mapping in DES was non-linear?



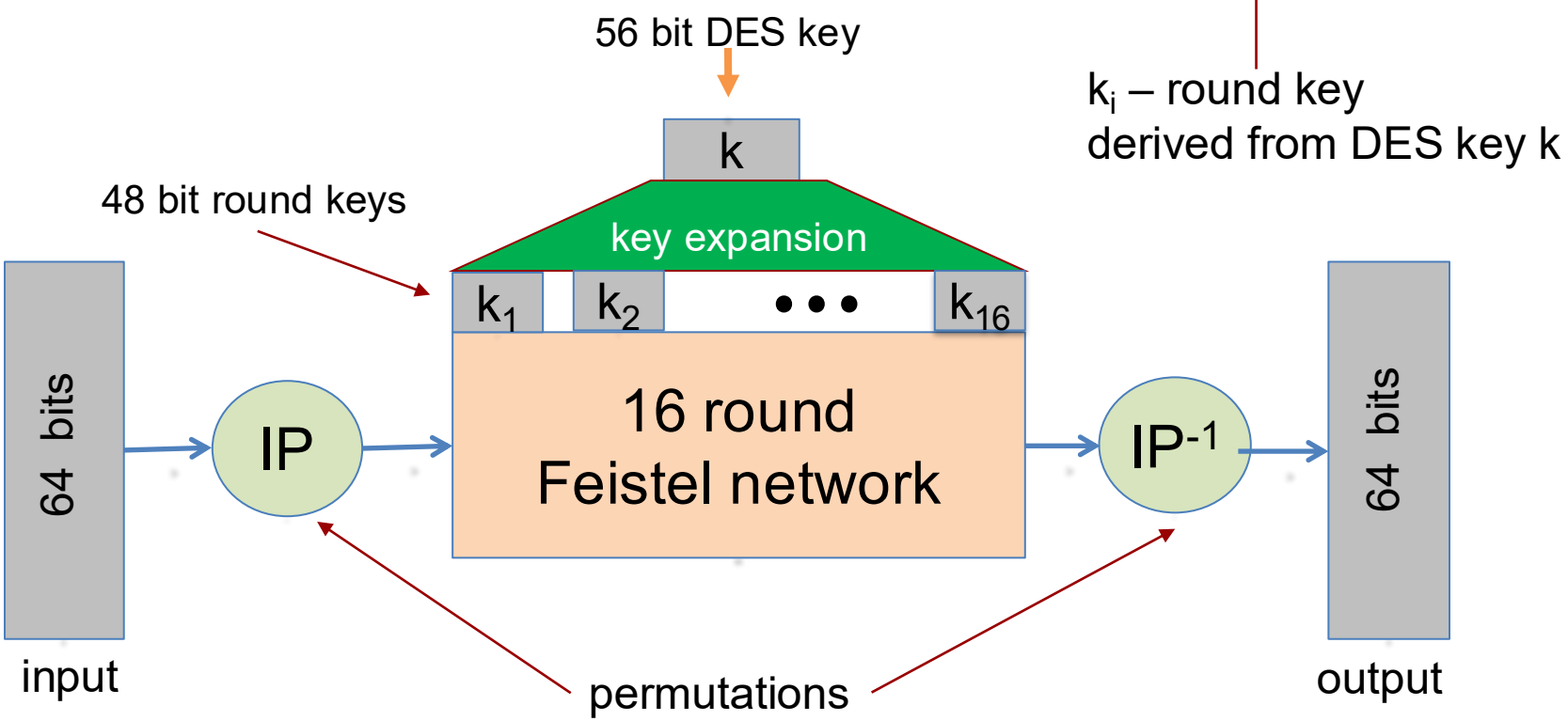
Theorem about theory of Feistel networks:

f: $K \times \{0,1\}^n \longrightarrow \{0,1\}^n$ a secure PRF (indistinguishable from random) $\Rightarrow$

3-round Feistel F: $K^3 \times \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$ a secure PRP

DES: 16 Round Feistel Network

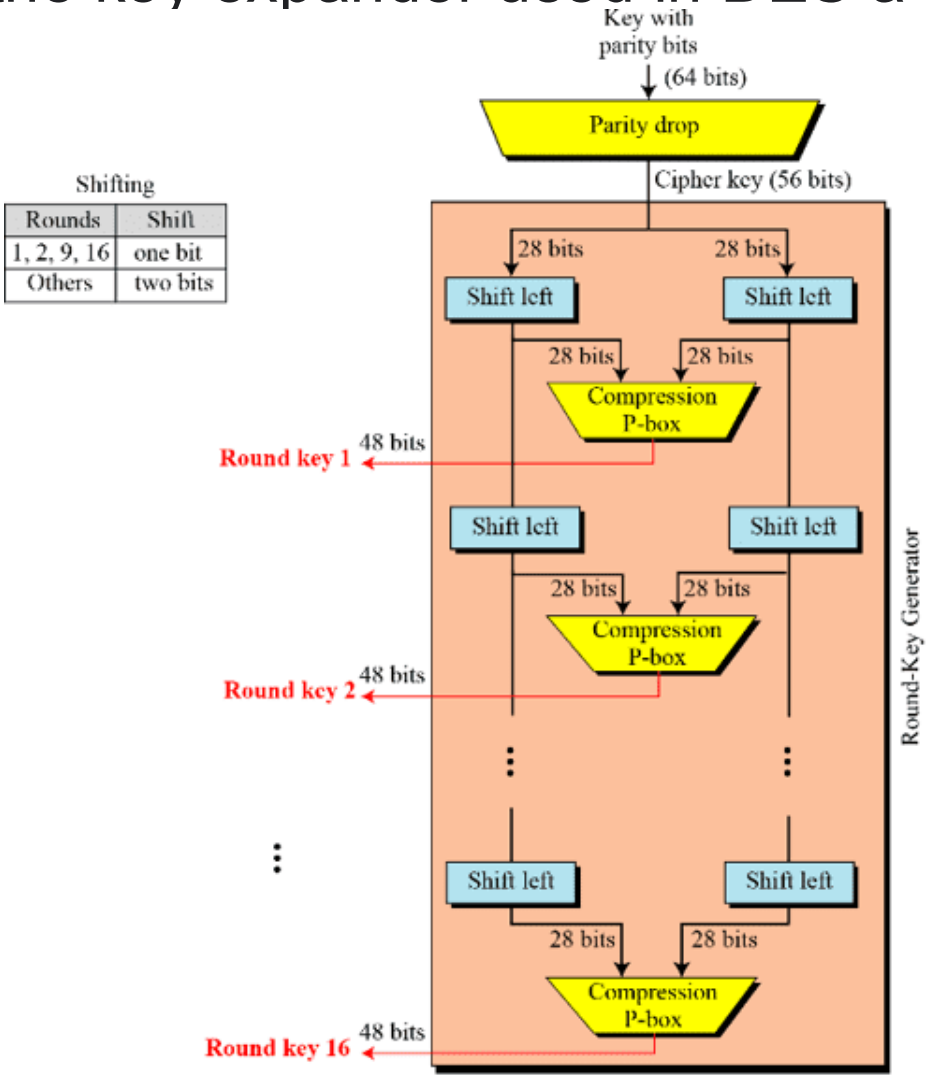
$f_1, \ldots, f_{16}: \{0,1\}^{32} \longrightarrow \{0,1\}^{32}$, $f_i(x) = \mathbf{F}(k_i, x)$



To invert, use the round keys in reverse order

Student's Question

- Is the key expander used in DES a PRG?



Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits

The initial key size is 64-bit which is reduced to the 56-bit key. This is done by discarding every 8_{th} bit from the 64-bit key

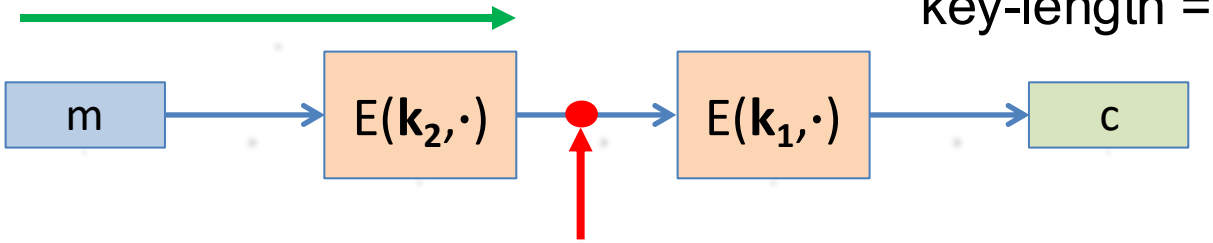
fixed permutations and cyclic shifts

USC Viterbi School of Engineering

What is Wrong with Double DES?

- Define $2E((k_1,k_2), m) = E(k_1, E(k_2, m))$

key-length = 112 bits for DES



Attack: $M = (m_1,\ldots, m_{10})$, $C = (c_1,\ldots,c_{10})$

separation of variables

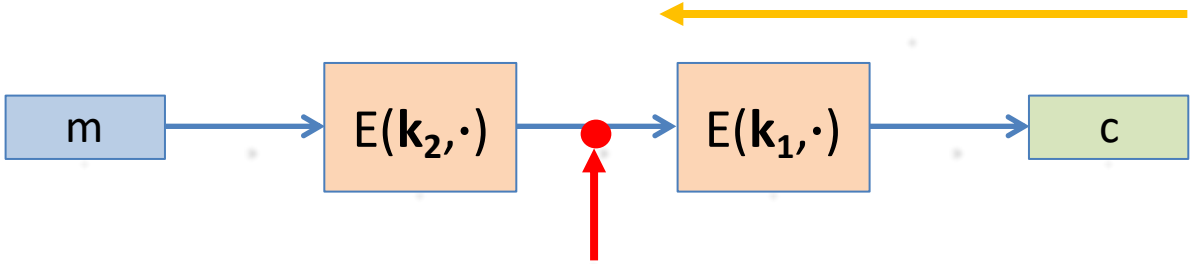
Find (k_1,k_2) such that $E(k_1, E(k_2, M)) = C$ Equivalently: $E(k_2, M) = D(k_1, C)$

Step 1: build table for all possible values of $k_2 \in \{0,1\}^{56}$ and sort on second column

$k^0 = 00\ldots00$	$E(k^0, M)$
$k^1 = 00\ldots01$	$E(k^1, M)$
$k^2 = 00\ldots10$	$E(k^2, M)$
$\vdots$	$\vdots$
$k^N = 11\ldots11$	$E(k^N, M)$

2^{56} entries

Meet in the Middle Attack



Attack: $M = (m_1,\ldots, m_{10})$, $C = (c_1,\ldots,c_{10})$

- Step 1: build table and sort
- Step 2: for all $k_1 \in \{0,1\}^{56}$ do:
  test if $D(k_1, C)$ is in 2nd column

if so then $E(k^i,M) = D(k_1,C) \Rightarrow (k^i,k_1) = (k_2,k_1)$

$k^0 = 00\ldots00$	$E(k^0, M)$
$k^1 = 00\ldots01$	$E(k^1, M)$
$k^2 = 00\ldots10$	$E(k^2, M)$
⋮	⋮
$k^N = 11\ldots11$	$E(k^N, M)$

$k^0 = 00\ldots00$	$D(k^0, C)$
$k^1 = 00\ldots01$	$D(k^1, C)$
$k^2 = 00\ldots10$	$D(k^2, C)$
⋮	⋮
$k^N = 11\ldots11$	$D(k^N, C)$

Student's question: how do we determine that the keys k1 and k2 are correct?

Lecture Outline

- Review
- Using block ciphers
  - Security for many-time key
  - Modes of operation: many time key (CBC mode)
  - Modes of operation: many time key (CTR mode)
- Message Integrity
  - MAC security

Recall: Use Cases

- Single use key: (one time key)
  - Key is only used to encrypt one message
    - Encrypted email: new key generated for every email
- Multi use key: (many time key)
  - Key used to encrypt multiple messages
    - Encrypted files: same key used to encrypt many files
  - Need more machinery than for one-time key

Using PRPs and PRFs

Goal: build "secure" encryption from secure PRP (e.g., AES)

First look at: one-time keys

1. Adversary's power:

   Adversary sees only one ciphertext (one-time key)

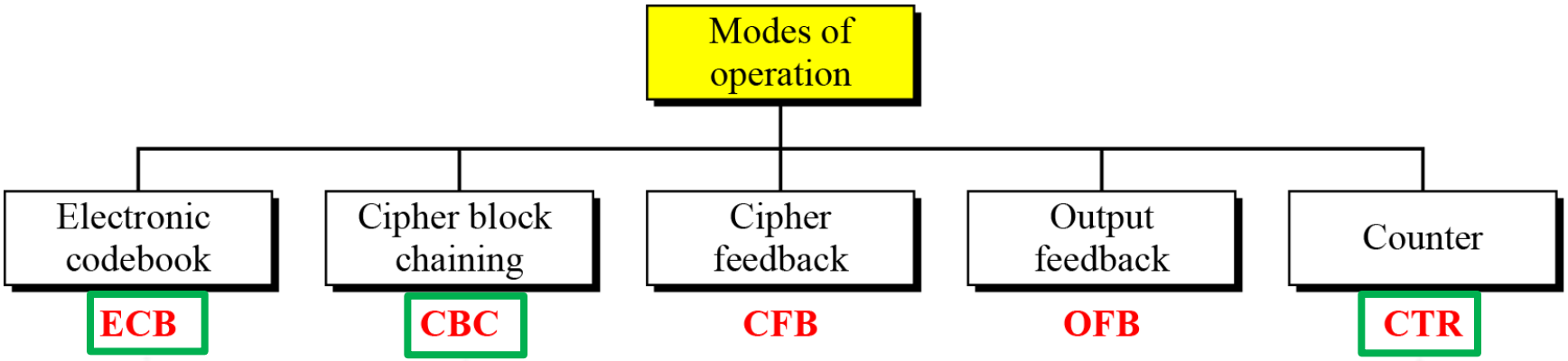
2. Adversary's goal:

   Learn info about plaintext from ciphertext (break semantic security)

Later: many-time keys (a.k.a chosen-plaintext security)

Block Cipher Modes of Operation

- Mode of operation - a technique for adapting the algorithm for an application
- Block ciphers encrypt only fixed-size blocks
  - If you want to encrypt something that isn't exactly one block long, you have to use a block cipher in a mode of operation



Block Cipher Modes of Operation

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key.	• Secure transmission of single values (e. g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.	• General-purpose block-oriented transmission • Authentication
Cipher Feedback (CFB)	Input is processed *j* bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	• General-purpose stream-oriented transmission • Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding DES output.	• Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	• General-purpose block-oriented transmission • Useful for high-speed requirements

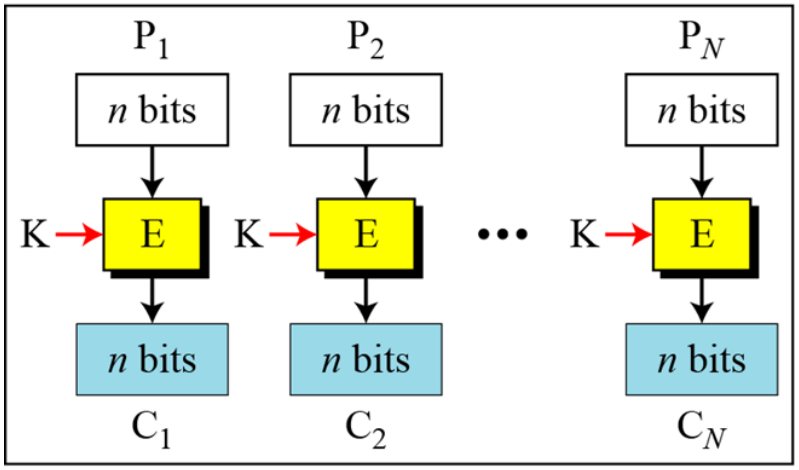
Electronic Codebook (ECB) Mode

- Message is broken into independent blocks which are encrypted
- Each block is a value which is substituted, like a codebook
- Each block is encoded independently of the other blocks

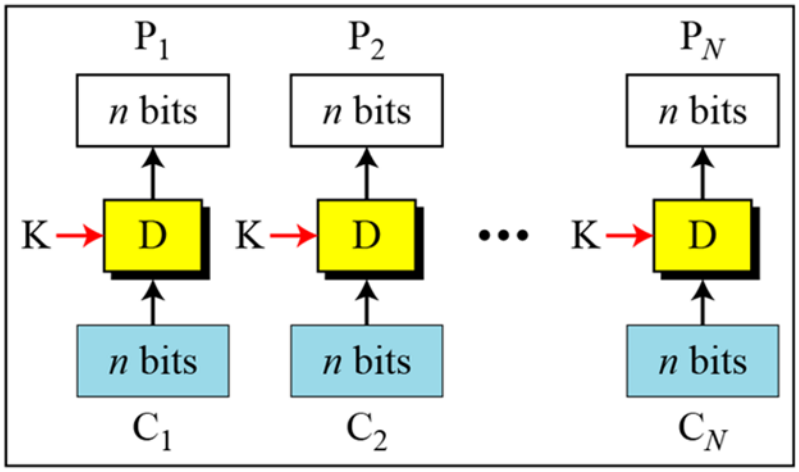
Encryption: $C_i = E_K(P_i)$

Decryption: $P_i = D_K(C_i)$

E: Encryption
D: Decryption
P_i: Plaintext block i
C_i: Ciphertext block i
K: Secret key



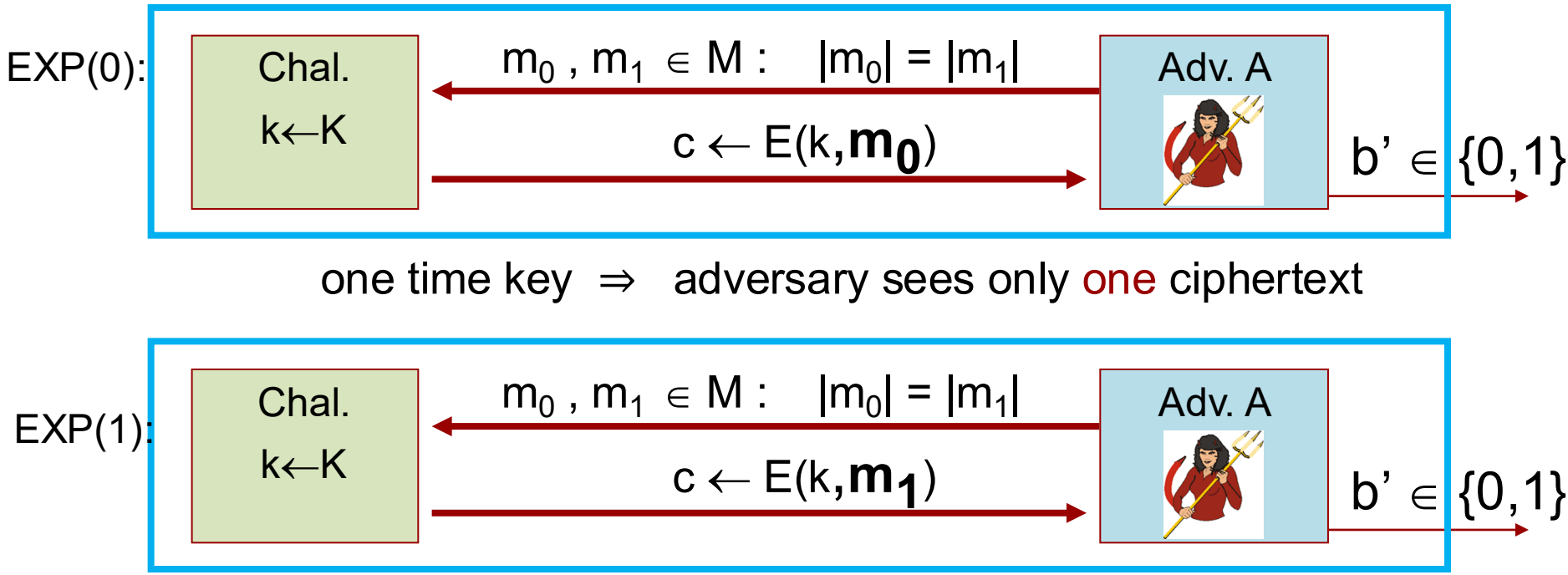
Encryption



Decryption

USC Viterbi School of Engineering

Recall: Semantic Security (One-time Key)

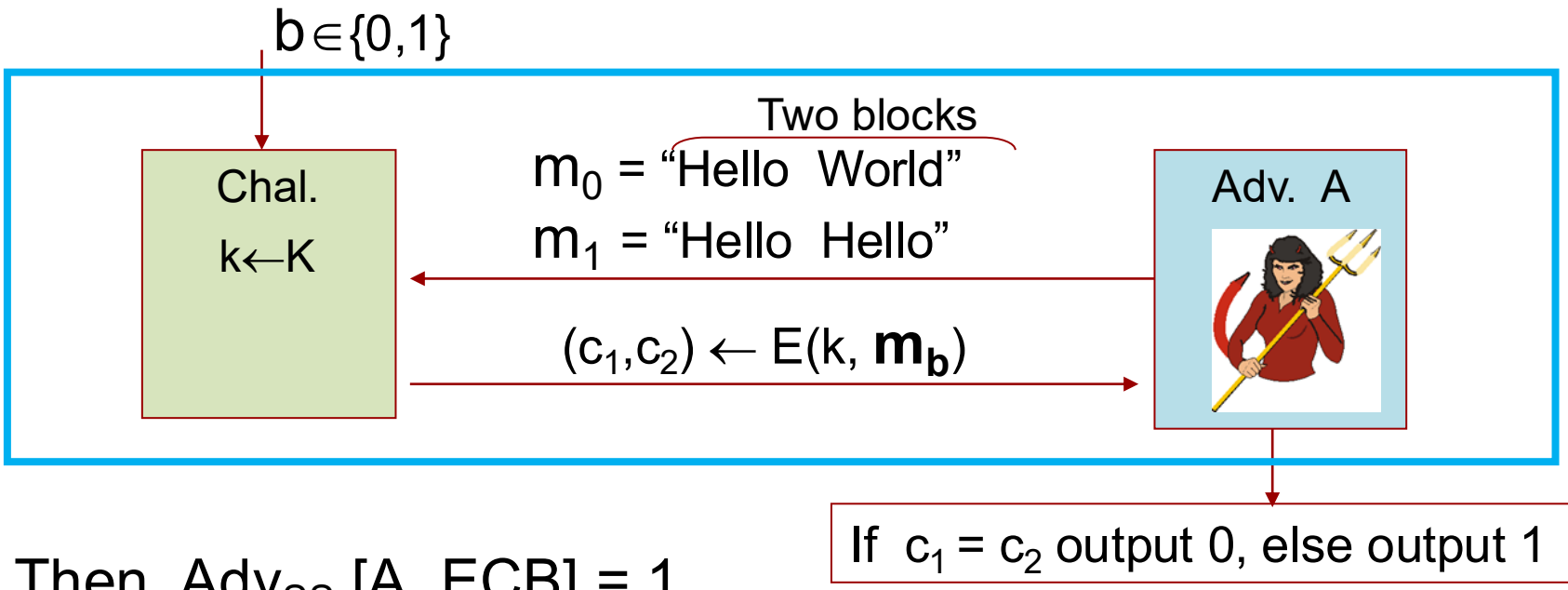


$Adv_{SS}[A,OTP] = \left| Pr[\ \mathbf{EXP(0)}=1\ ] - Pr[\ \mathbf{EXP(1)}=1\ ] \right|$ should be "negligible"

USC Viterbi School of Engineering

ECB is not Semantically Secure

- ECB is not semantically secure for messages that contain more than one block

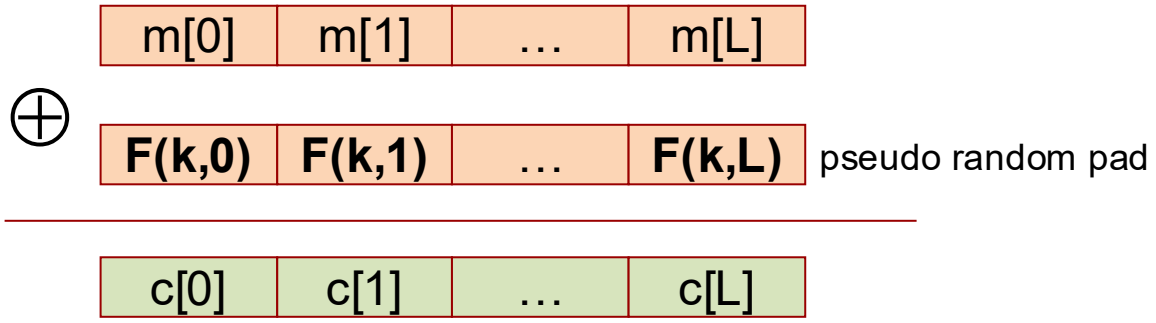


Then Adv_{SS} [A, ECB] = 1

Secure Construction 1
(build stream cipher out of block cipher)

Deterministic counter mode from a PRF F:

- $E_{DETCTR}(k, m)$ =



⇒ Stream cipher built from a PRF(e.g., AES, 3DES)

Deterministic Counter-mode Security

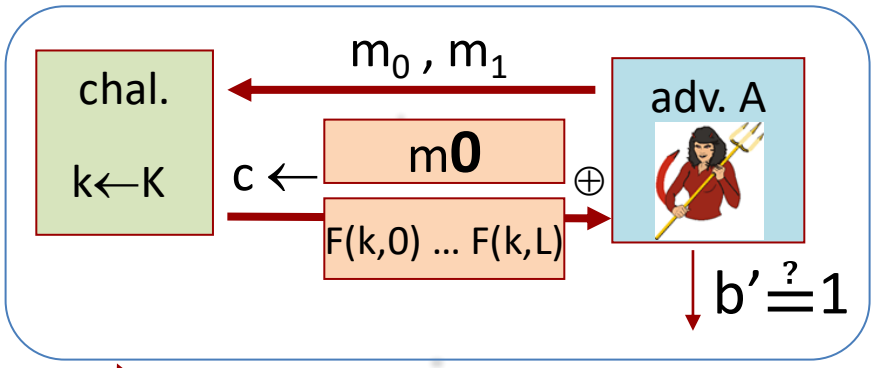
Theorem: For any L > 0,

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

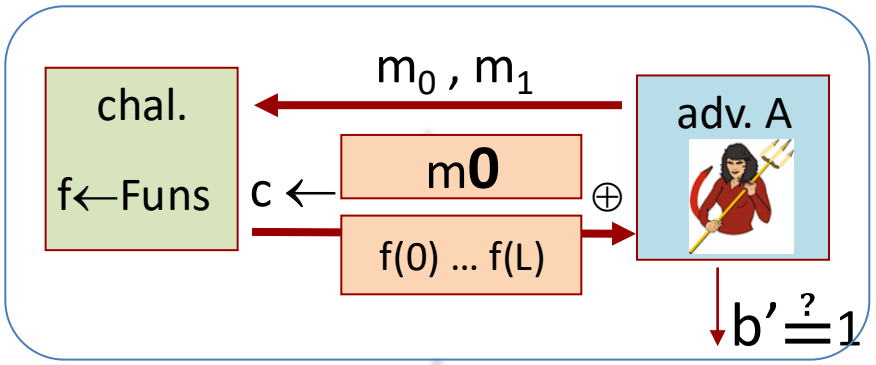
E_{DETCTR} is semantically secure cipher over (K,X^L,X^L)

DETCTR Security: Proof in Pictures

PRF is computationally indistinguishable from a truly random function



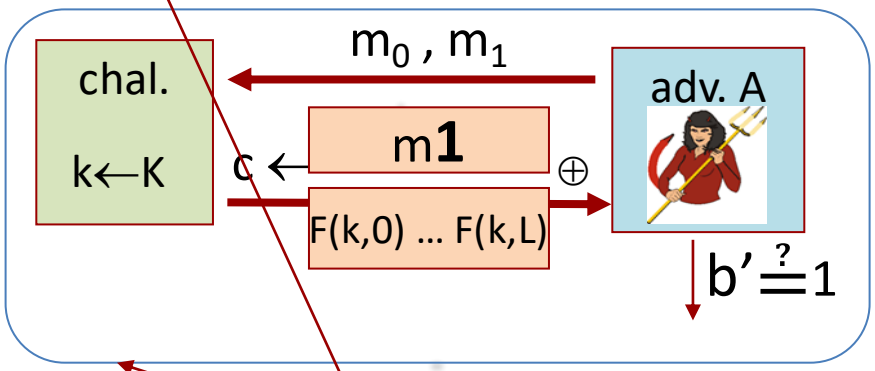
≈p



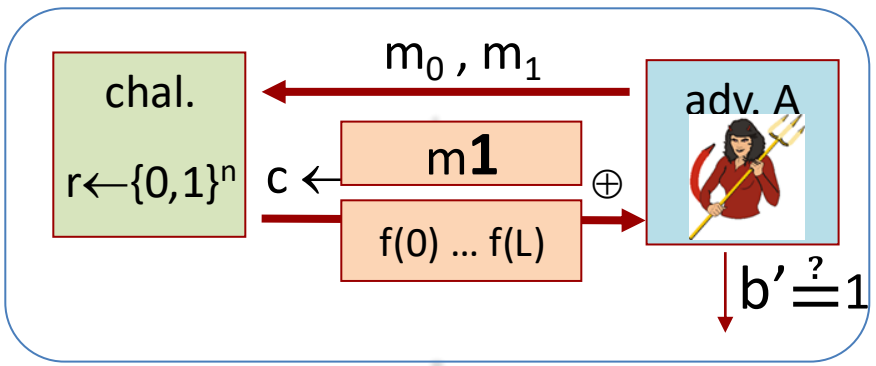
≈p

f is a truly random function, and the pad is OTP

=



≈p



Want to prove: these two distributions are computationally indistinguishable

Deterministic counter mode is semantically secure!

Semantic Security for Many-time Key

Key used more than once ⇒ adversary sees many ciphertexts encrypted with the same key

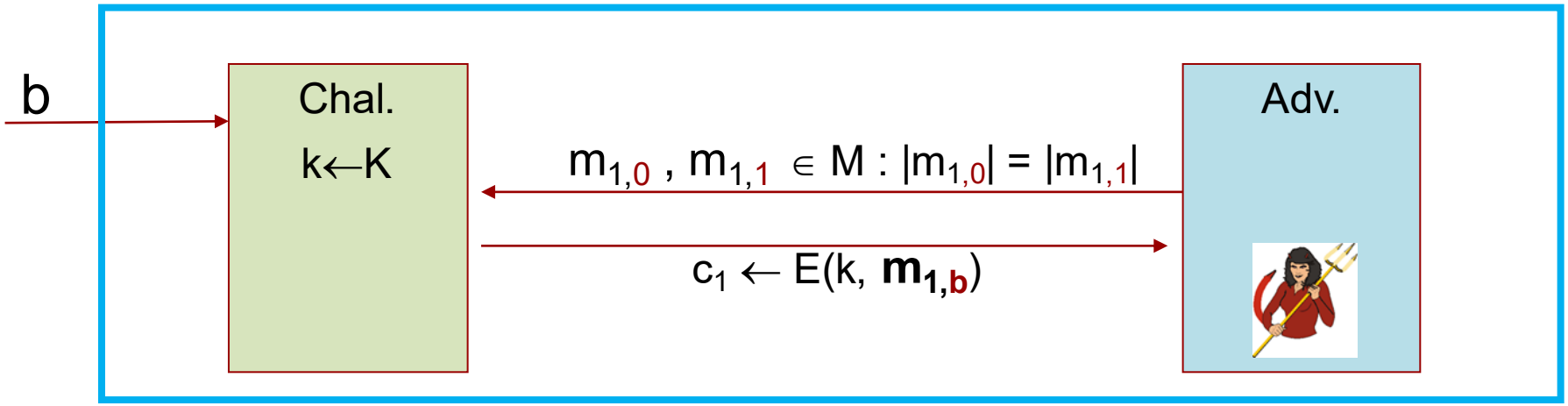
Adversary's power: chosen-plaintext attack (CPA)

- Can obtain the encryption of arbitrary messages of her choice

Adversary's goal: break sematic security

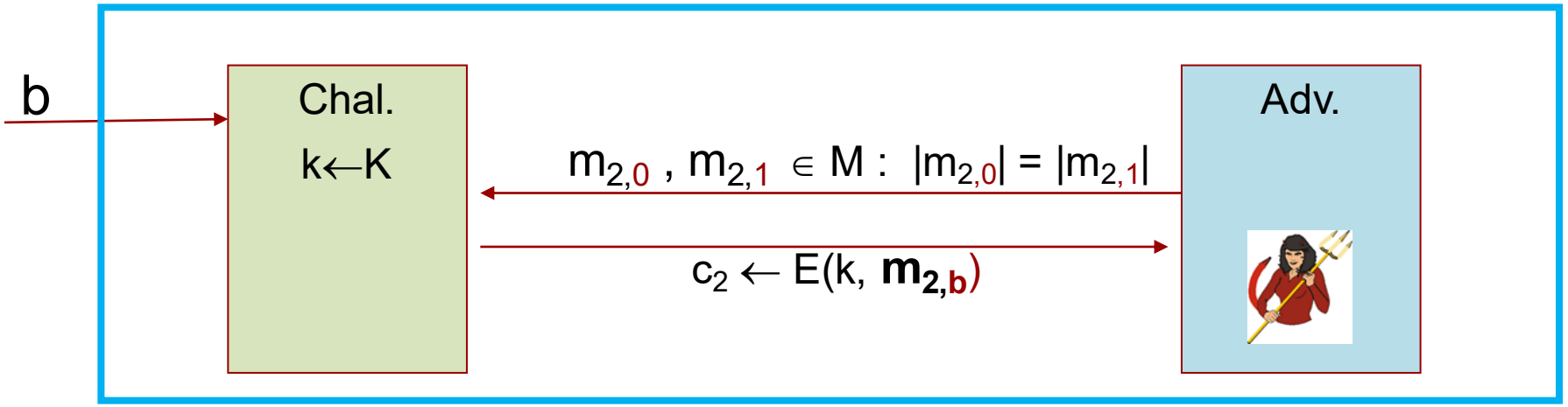
Semantic Security for Many-time Key

$\mathbb{E}$ = (E,D) a cipher defined over (K,M,C) For b = 0,1 define EXP(b) as:



Semantic Security for Many-time Key

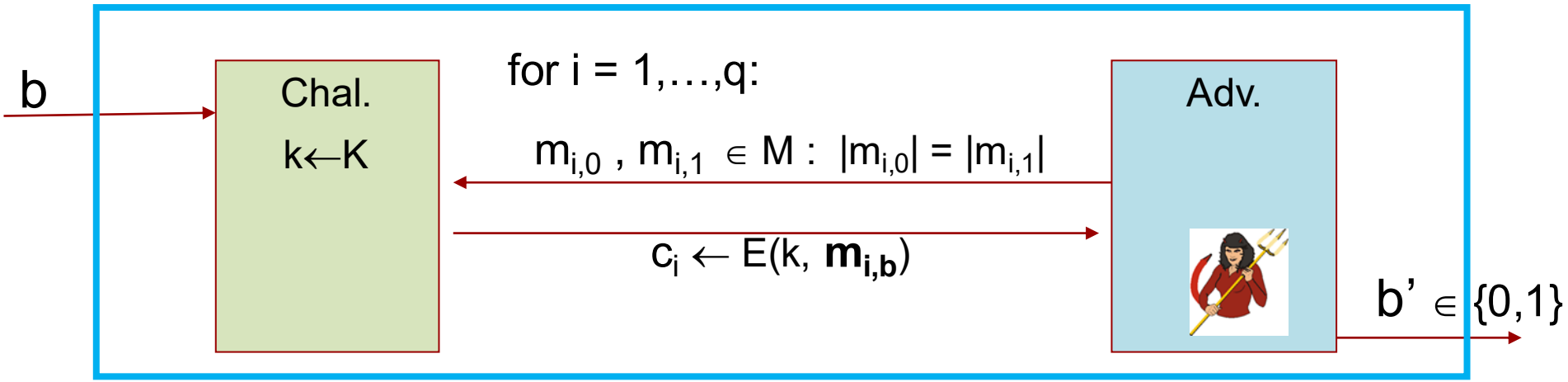
$\mathbb{E}$ = (E,D) a cipher defined over (K,M,C) For b = 0,1define EXP(b) as:



In a CPA attack the adversary can repeat the query

Semantic Security for Many-time Key (CPA Security)

$\mathbb{E}$ = (E,D) cipher defined over (K,M,C) For b = 0,1 define EXP(b):



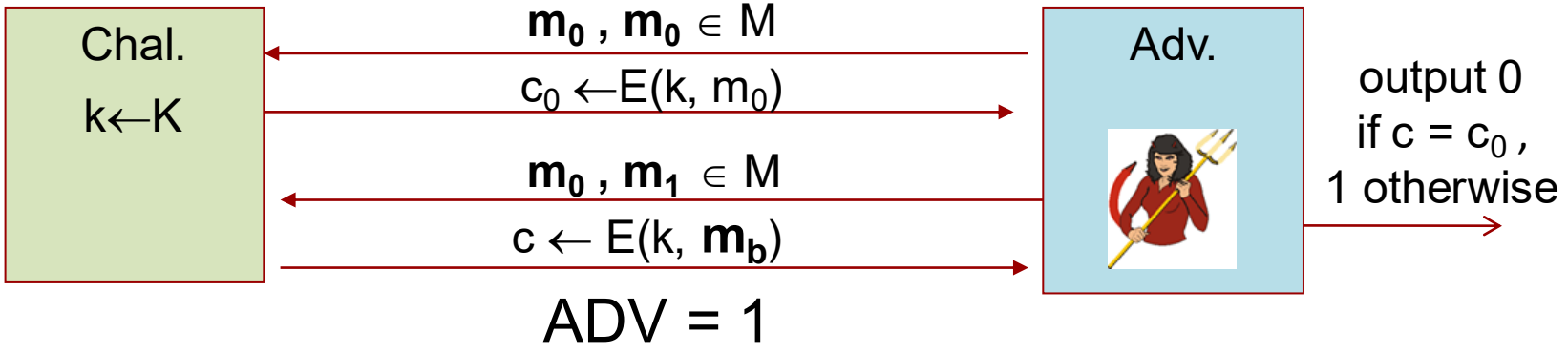
- CPA: if Eve wants encryption of a **particular message** c = E(k, m), she queries with $m_{j,0} = m_{j,1} = m$
- Given the encryption of arbitrary messages of her choice, her goal is to break semantic security for **some other** challenge ciphertexts

Definition: $\mathbb{E}$ is semantically security under CPA if for all "efficient" A:

$Adv_{CPA}[A,\mathbb{E}] = |Pr[EXP(0) = 1] - Pr[EXP(1) = 1]|$ is "negligible"

Ciphers Insecure under CPA (Deterministic Encryption)

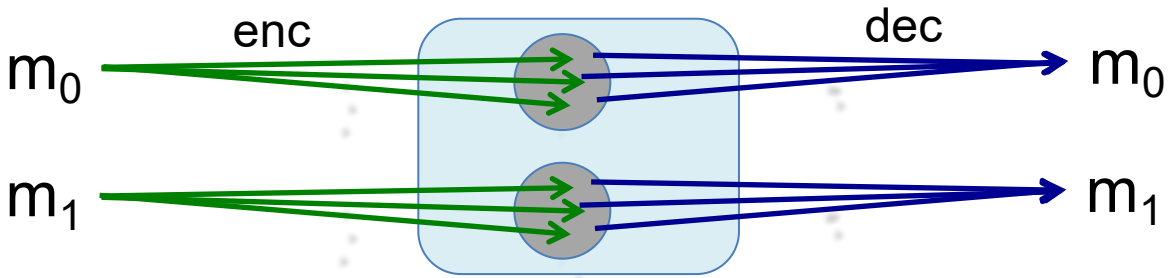
Suppose E(k,m) always outputs same ciphertext for message m, then:



- 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

Solution 1: Randomized Encryption

- E(k,m) is a randomized algorithm:

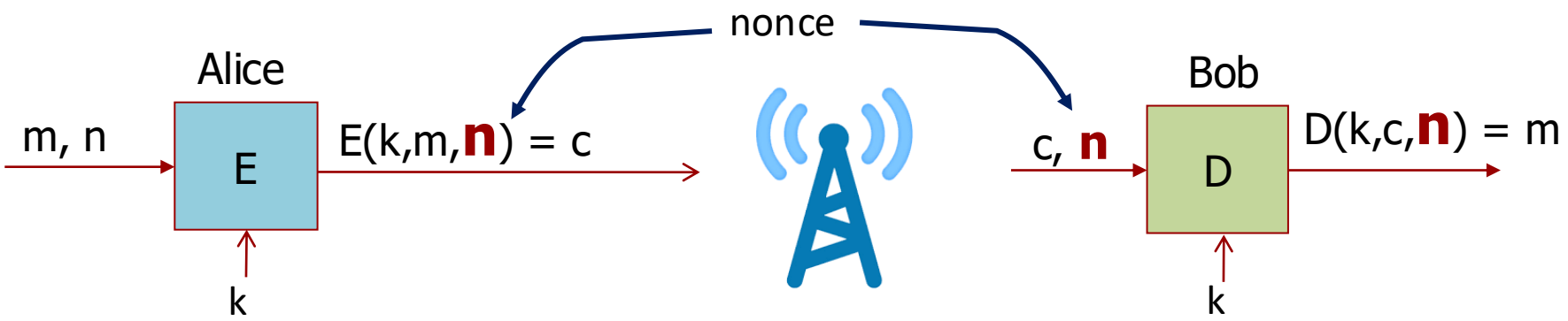


⇒ encrypting same message twice gives different ciphertexts (with high probability)

⇒ ciphertext must be longer than plaintext

Roughly: $size_{CT} = size_{PT}$ + "# random bits"

Solution 2: Nonce-based Encryption



- nonce n: value that changes from message to message (k,n) pair never used more than once

- Method: nonce is a counter (e.g., packet counter)

Lecture Outline

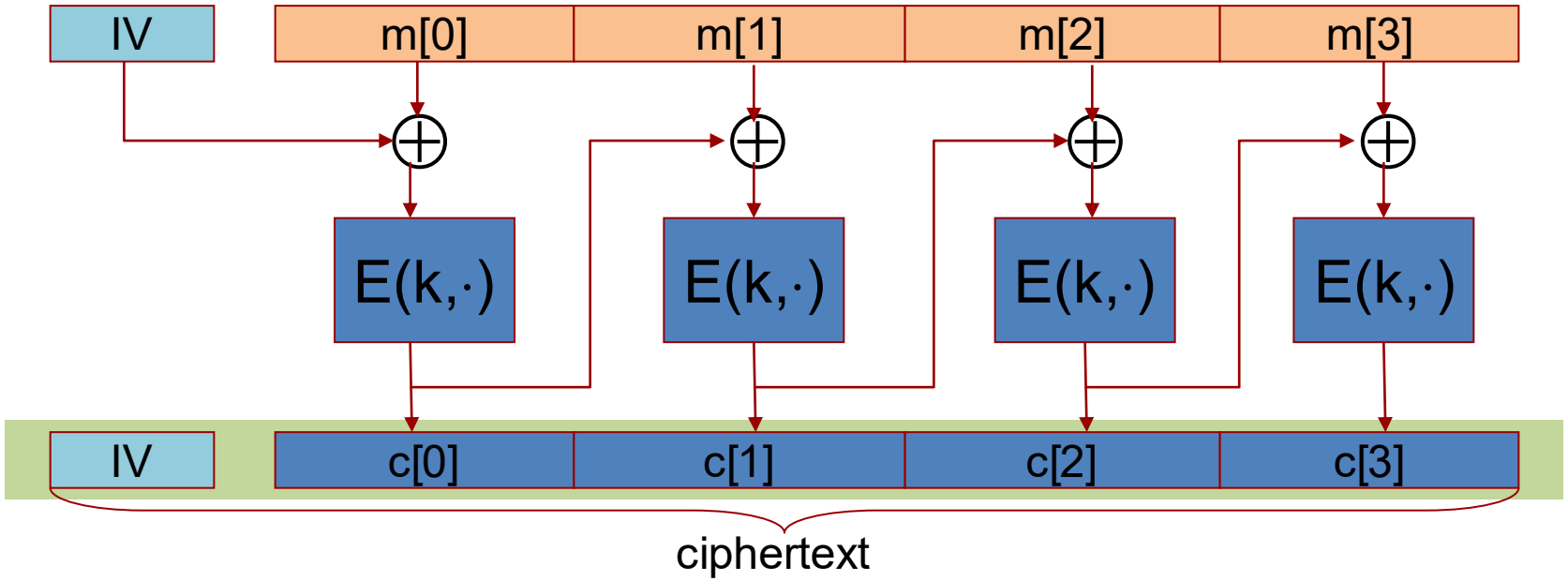
- Review
- Using block ciphers
  - Security for many-time key
  - Modes of operation: many time key (CBC mode)
  - Modes of operation: many time key (CTR mode)
- Message Integrity
  - MAC security

Cipher Block Chaining (CBC)

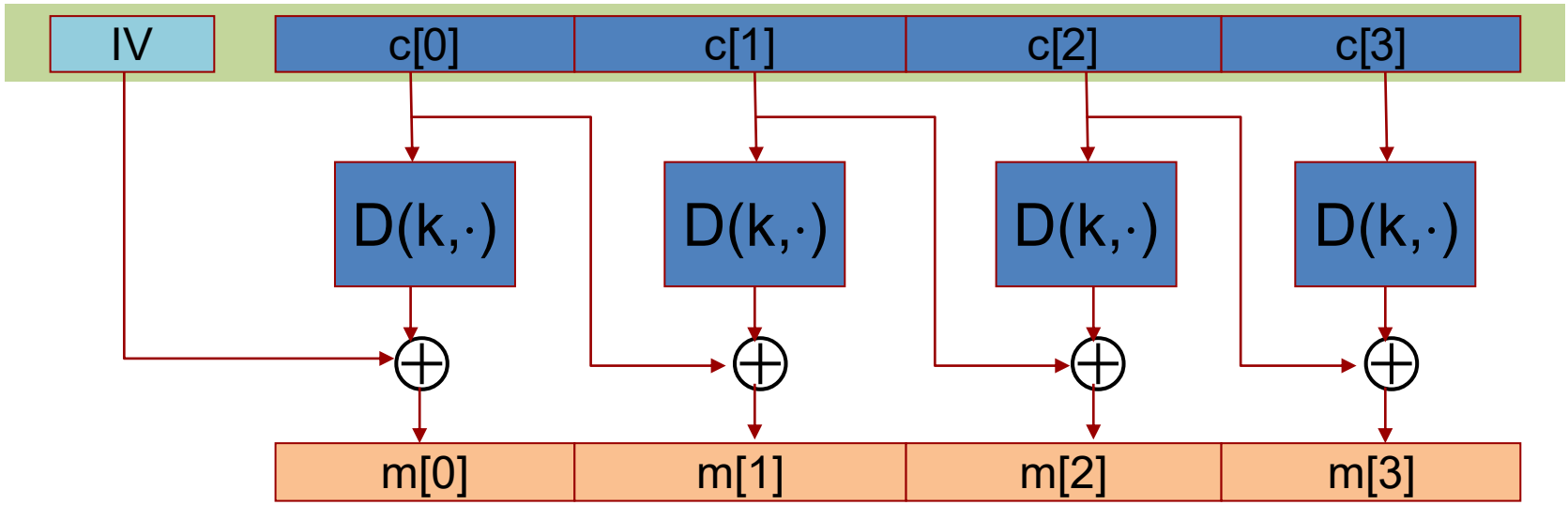
- Message is broken into blocks linked together in encryption operation
- Each previous cipher blocks is chained with current plaintext block (dependent on all blocks before it)
- Use random Initialization Vector (IV) to start the process
  - Captures the randomness used in encryption
- Uses: bulk data encryption, authentication

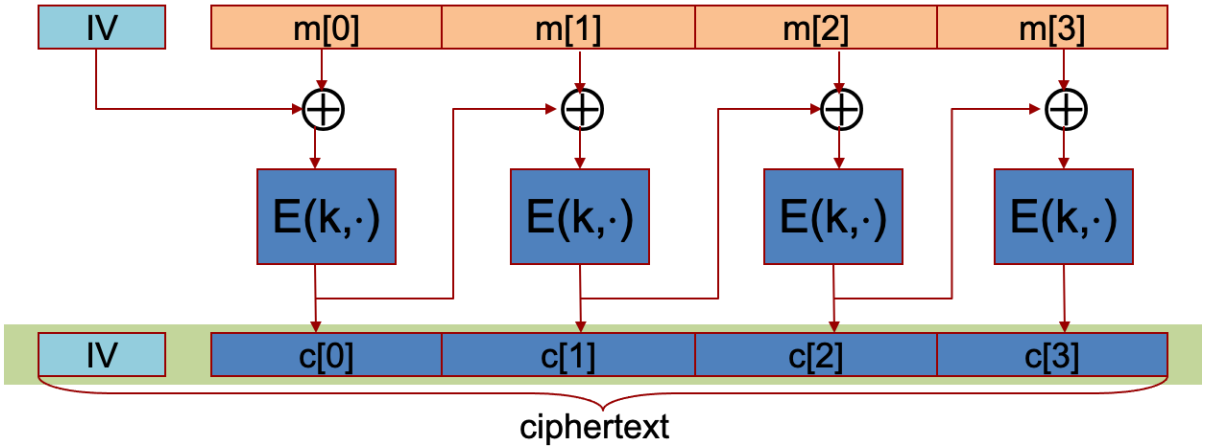
Construction 1: CBC with random IV

Let (E,D) be a PRP (block cipher)
$E_{CBC}(k,m)$: choose random IV ∈ X (the size of IV is the size of the block) and do:



CBC Decryption Circuit





CBC: CPA Analysis

CBC Theorem: For any L > 0, L is message length in blocks

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

E_{CBC} is a semantically secure under CPA over (K, X^L, X^{L+1})

For a q-query adversary A attacking E_{CBC} there exists a PRP adversary B:

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

negligible, since PRP is secure

error term

q is the number of times that we have used the key K to encrypt messages

CBC is only secure as long as $q^2L^2 << |X|$

Details about the formula and the error term: "A Graduate Course in Applied Cryptography", D. Boneh and V. Shoup Version 0.3, December 2016

Practical Application of CPA Analysis

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

q = number of messages encrypted with k, L = length of max message

Suppose we want $Adv_{CPA}[A, E_{CBC}] \leq 1/2^{32} \Leftarrow 2q^2 L^2 / |X| < 1/2^{32}$

then $(2q^2 L^2 // |X|) < 1/2^{32}$

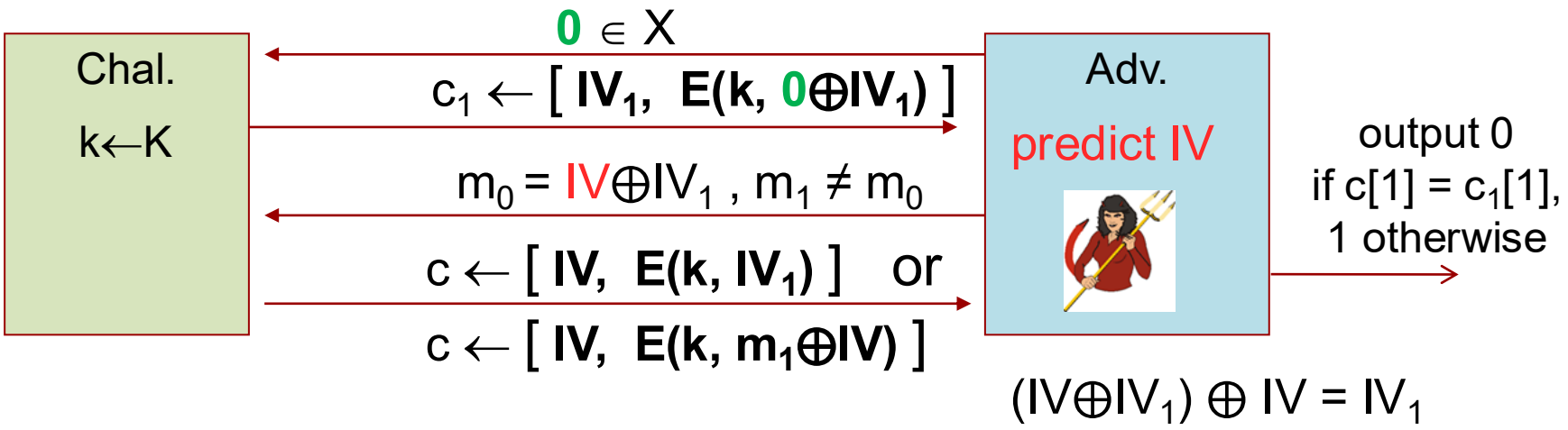
- AES: $|X| = 2^{128} \Rightarrow 2q^2 L^2 < 2^{128-32}$, $q L < 2^{47}$
- 3DES: $|X| = 2^{64} \Rightarrow 2q^2 L^2 < 2^{64-32}$, $q L < 2^{15}$

Practical application: the security theorem tells exactly how frequently one needs to replace the key when using CBC

Attack on CBC with Non-Random IV

CBC where attacker can predict the IV is not CPA-secure!

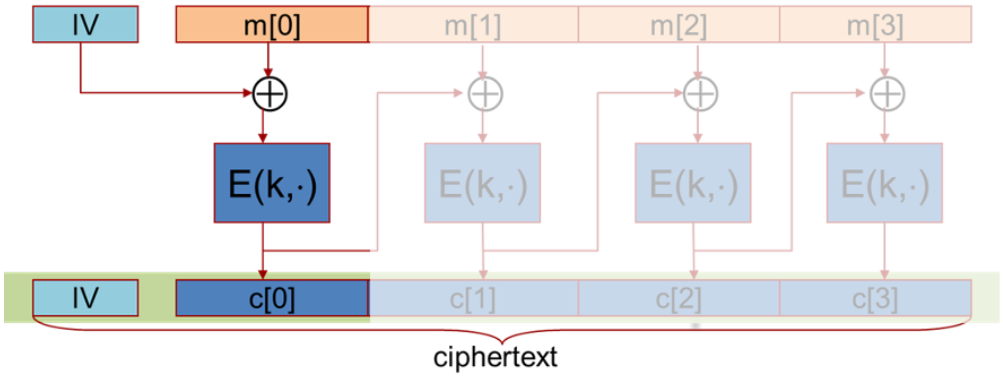
Suppose given $c \leftarrow E_{CBC}(k,m)$ can predict IV for next message



When you use CBC encryption, the IV must be random!

Nonrandom and Predictable IVs

- Nonrandom and predictable IVs are more likely to correlate with the plaintext and leak information
- Simple example:
  - Counter, starting from zero, used as an IV
  - The first one-block message is 0
  - The second one-block message 1
  - Both ciphertexts will be the same!
- IV for CBC is predictable: chained IV
  - Example: bug in SSL/TLS 1.0
  - IV for next record is last ciphertext block of current record
  - Not CPA secure (a practical exploit: BEAST attack)
  - https://niiconsulting.com/checkmate/2013/12/ssltls-attacks-part-1-beast-attack/

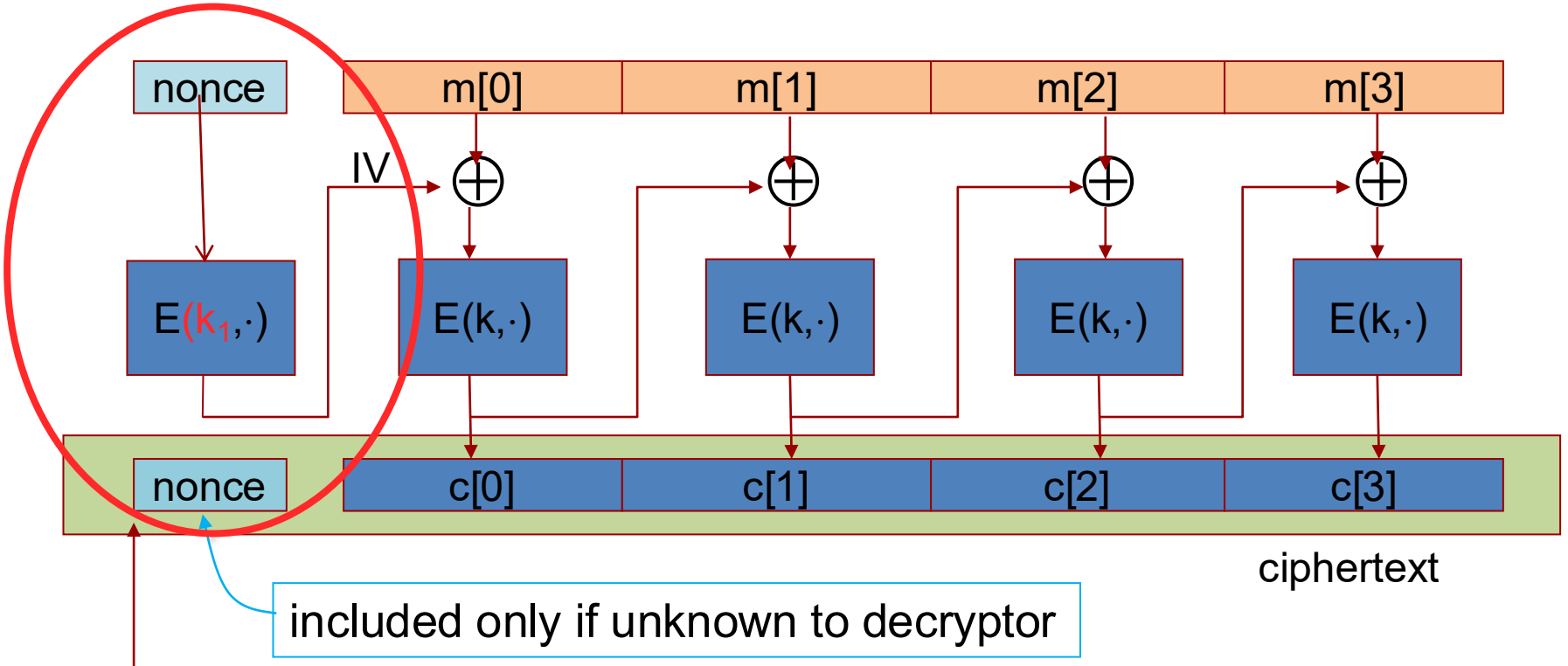


Construction 1': Nonce-based CBC

two independent keys, k and k1

- Cipher block chaining with unique nonce: key = (k, k_1)

unique nonce means: (key, n) pair is used for only one message



This extra encryption step with independent key is required for CPA security!

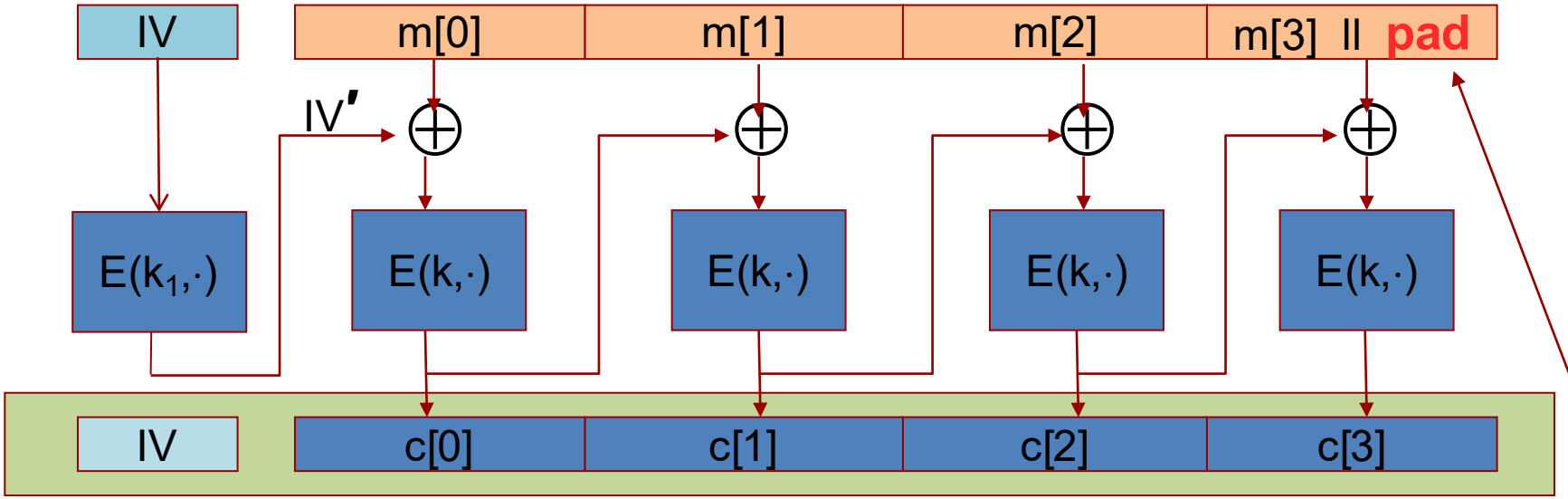
Example Crypto API (OpenSSL)

```
void AES_cbc_encrypt(
   const unsigned char *in,
   unsigned char *out,
   size_t length,
   const AES_KEY *key,
   unsigned char *ivec,      ← user supplies IV
   AES_ENCRYPT or AES_DECRYPT);
```

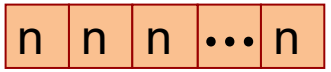
When IV is not random you need to encrypt it using AES before use, otherwise no CPA security!

CBC Padding

- What should we do when the message is not a multiple of the block cipher block length?



TLS padding: for n > 0, n byte pad is



if no pad needed, add a dummy block, n = 16

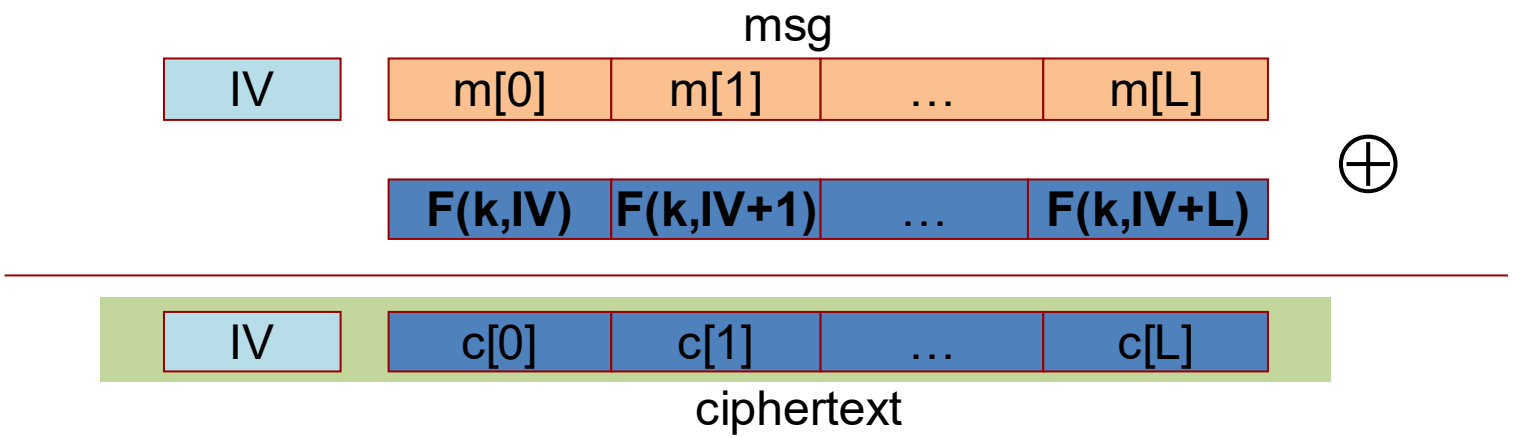
removed during decryption

USC Viterbi School of Engineering

Construction 2: Randomized CTR-mode

Let F: K × $\{0,1\}^n \longrightarrow \{0,1\}^n$ be a secure PRF (do not need block cipher because we do not need to invert)

E(k,m): choose a random IV $\in \{0,1\}^n$ and do:



Parallelizable (unlike CBC)

Encryption algorithm chooses (at random) a new IV for every message

Randomized CTR-mode (rand. IV): CPA Analysis

- **Counter-mode Theorem**: For any L > 0

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

  E_{CTR} is a semantically secure under CPA over (K,X^L,X^{L+1})

  In particular, for a q-query adversary A attacking E_{CTR} there exists a PRF adversary B:

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

negligible, since PRF is secure

error term

CTR-mode only secure as long as $2q^2L << |X|$

Details about the formula and the error term: "A Graduate Course in Applied Cryptography", D. Boneh and V. Shoup Version 0.3, December 2016

Example

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

q = number of messages encrypted with k, L = length of max message

Suppose we want $Adv_{CPA}[A, E_{CTR}] \leq 1/2^{32} \Leftarrow 2q^2 L/|X| < 1/2^{32}$

- AES: $|X| = 2^{128} \Rightarrow q^2 L < 2^{128-32}$, $qL^{1/2} < 2^{47}$
- Better than CBC!

Comparison: CTR vs. CBC

	CBC	CTR mode
uses	PRP	PRF
parallel processing	No	Yes
Security of rand. enc.	$q^2 L^2 << \|X\|$	$q^2 L << \|X\|$
dummy padding block	Yes*	No
1 byte msgs (nonce-based)	16x expansion	no expansion

*for CBC, dummy padding block can be solved using ciphertext stealing, modification to CBC

Summary

- PRPs and PRFs: a useful abstraction of block ciphers
- We examined two security notions (security against eavesdropping only):
  1. Semantic security against one-time CPA
  2. Semantic security against many-time CPA

  Note: neither mode ensures data integrity

Goal \ Power	One-time key	Many-time key (CPA)	CPA and integrity
Semantic Security	steam-ciphers deterministic CTR-mode	randomized CBC randomized CTR-mode	Coming next

Lecture Outline

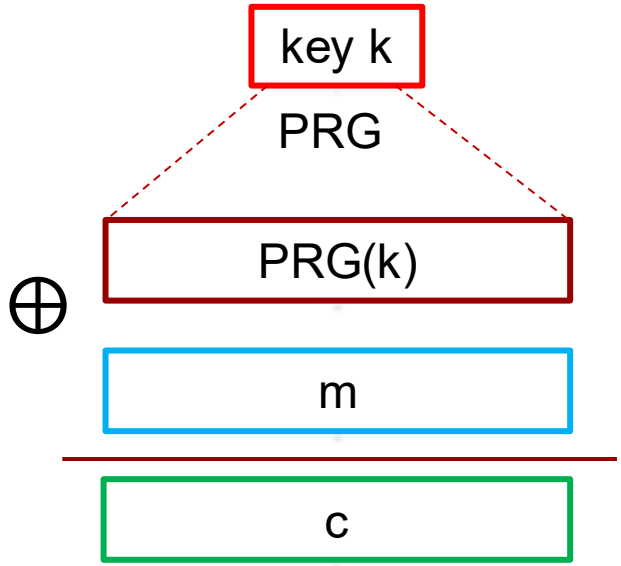
- Review
- Using block ciphers
  - Security for many-time key
  - Modes of operation: many time key (CBC mode)
  - Modes of operation: many time key (CTR mode)
- Message Integrity
  - MAC security

Message Integrity

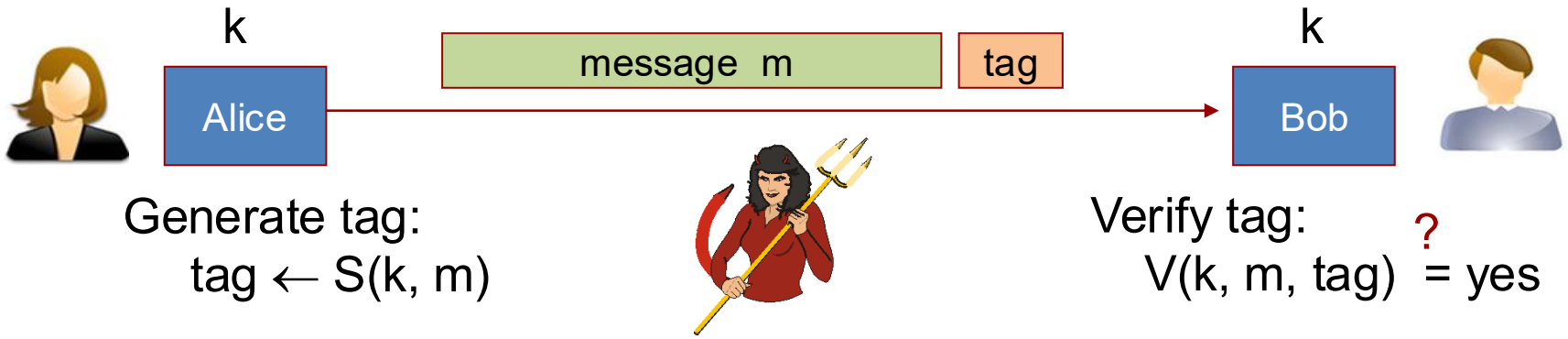
- Confidentiality: semantic security against a CPA attack
  - Encryption secure against eavesdropping only
- Goal: integrity, no confidentiality

Message Integrity

- Bob receives message from Alice, wants to ensure:
  - Message originally came from Alice
  - Message not changed since sent by Alice
- In general, encryption schemes do not ensure message integrity
- Example:



Message Integrity: Message Authentication Code (MAC)



Definition: MAC I = (S,V) defined over $(\mathcal{K},\mathcal{M},\mathcal{T})$ is a pair of algorithms:

1. Signing algorithm S(k,m) outputs t in $\mathcal{T}$
2. Verification algorithm: V(k,m,t) outputs yes or no

Consistency requirement:

$\forall m \in \mathcal{M}, \forall k \in \mathcal{K}$: V(k, m, S(k,m)) = yes

Secure MAC

Attacker's power: chosen message attack

- for $m_1, m_2, \ldots, m_q$ attacker is given $t_i \leftarrow S(k, m_i)$

Attacker's goal: existential forgery

- produce some new valid message/tag pair (m,t)

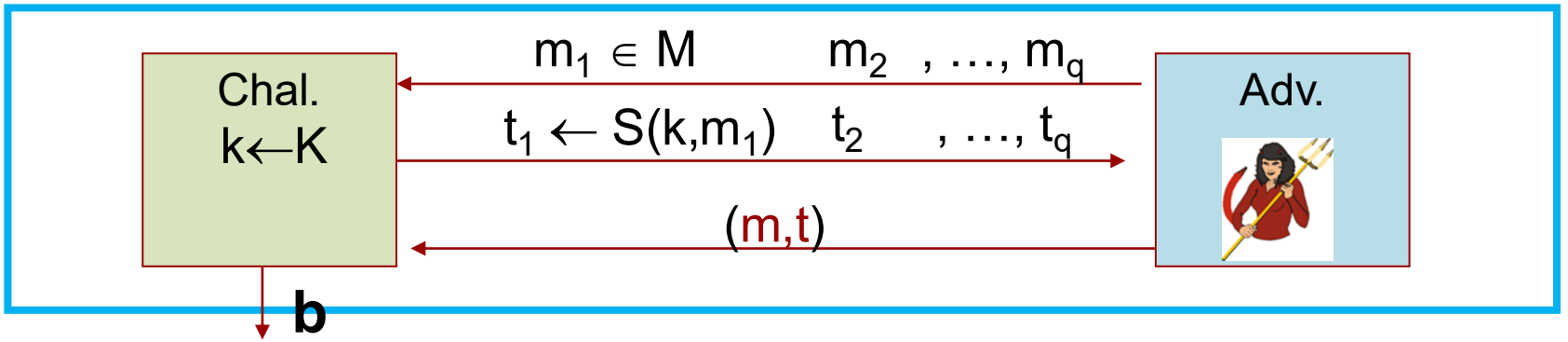
$$(m,t) \notin \{ (m_1,t_1) , \ldots , (m_q,t_q) \}$$

⇒ attacker cannot produce a valid tag for a new message

⇒ given (m,t) attacker cannot produce new tag for the same message (m,t') for t' ≠ t

Secure MAC Game

- For a MAC I = (S,V) and adversary A define a MAC game as:



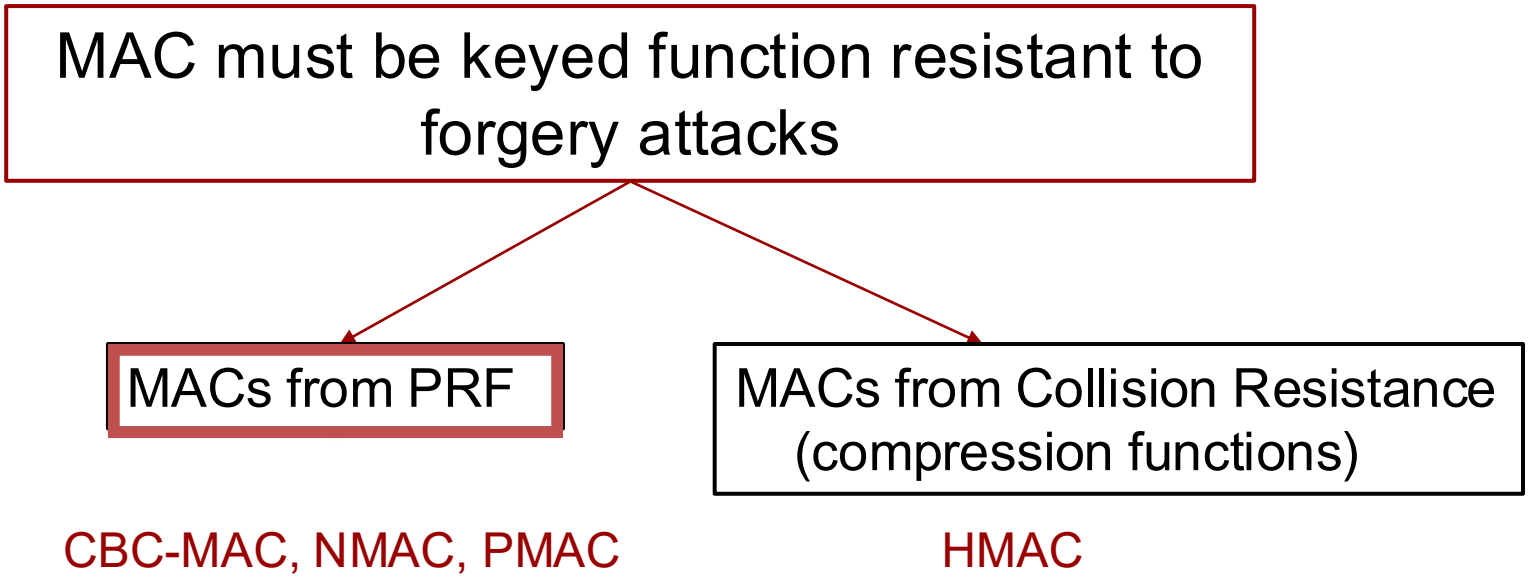
b = 1 if V(k,m,t) = yes and (m,t) $\notin$ { (m_1,t_1) , … , (m_q,t_q) }

b = 0 otherwise

Definition: I = (S,V) is a secure MAC if for all "efficient" A:

$Adv_{MAC}[A,I]$ = Pr[Challenger outputs 1 (yes)] is "negligible"

MAC: Security Goals vs. Specific Algorithms



How do we build secure MACs?

Secure PRF ⇒ Secure MAC

For a PRF $F: K \times X \longrightarrow Y$ define a MAC $I_F = (S,V)$ as:

- S(k,m) = F(k,m) value of the function at point m
- V(k,m,t): output yes if t = F(k,m) and no otherwise

Bad MAC Example

- Suppose F: K × X $\longrightarrow$ Y is a secure PRF with Y = $\{0,1\}^{10}$ - outputs only 10 bits
- Is the derived MAC I_F a secure MAC system?
  - No tags are too short: Eve can guess the tag for any message

$$Adv[A,I_F] = 1/2^{10} = 1/1024$$

The output of PRF should be large!

MAC Security

Theorem: If **F: K×X⟶Y** is a secure PRF and 1/|Y| is negligible (i.e. |Y| is large) then I_F is a secure MAC

In particular, for every efficient MAC adversary A attacking I_F there exists an efficient PRF adversary B attacking F:

$$Adv_{MAC}[A, I_F] \leq Adv_{PRF}[B, F] + 1/|Y|$$

$\Rightarrow$ I_F is secure as long as |Y| is large, say $|Y| = 2^{80}$

Details about the formula and the error term: "A Graduate Course in Applied Cryptography", D. Boneh and V. Shoup Version 0.3, December 2016

MAC Examples

- Any secure PRF is also a secure MAC
  - AES: a MAC for 16-byte messages
- Main question: given a MAC for **small** messages can we build a MAC for **large** messages?
- Two main constructions used in practice:
  1. CBC-MAC (banking – ANSI X9.9, X9.19, FIPS 186-3)
  2. HMAC (Internet protocols: SSL, IPsec, SSH, …)
- Both convert a small-PRF into a big-PRF

Truncating MACs based on PRFs

Lemma: suppose $F: K \times X \longrightarrow \{0,1\}^n$ is a secure PRF

Then so is $F_t(k,m) = F(k,m)[1\ldots t]$ for all $1 \le t \le n$

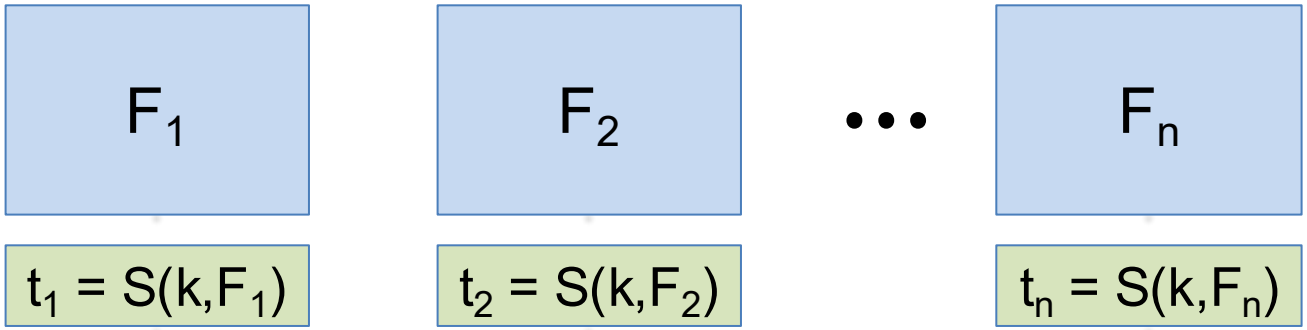
first t-bit of output

$\Rightarrow$ if (S,V) is a MAC based on a secure PRF outputting n-bit tags, the truncated MAC outputting t bits is secure

… as long as $1/2^t$ is still negligible (e.g., $t \ge 64$)

Example: Protecting System Files

- Suppose at install time the system computes:



Later a virus infects system and modifies system files

User reboots into clean OS and supplies the password

– Then: secure MAC ⇒ all modified files will be detected