Crypto concepts

Background

Guest lecturer: Mario Frank
Slide credits: Dan Boneh

Cryptography

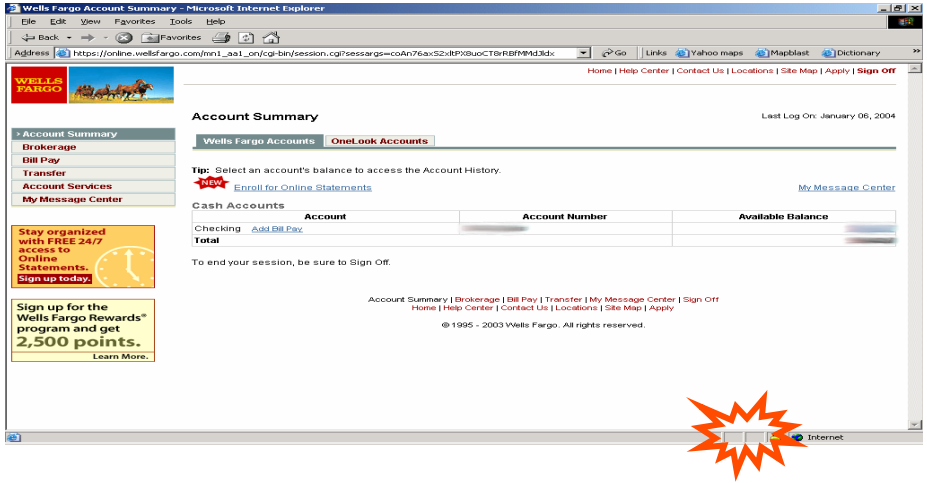
Is:

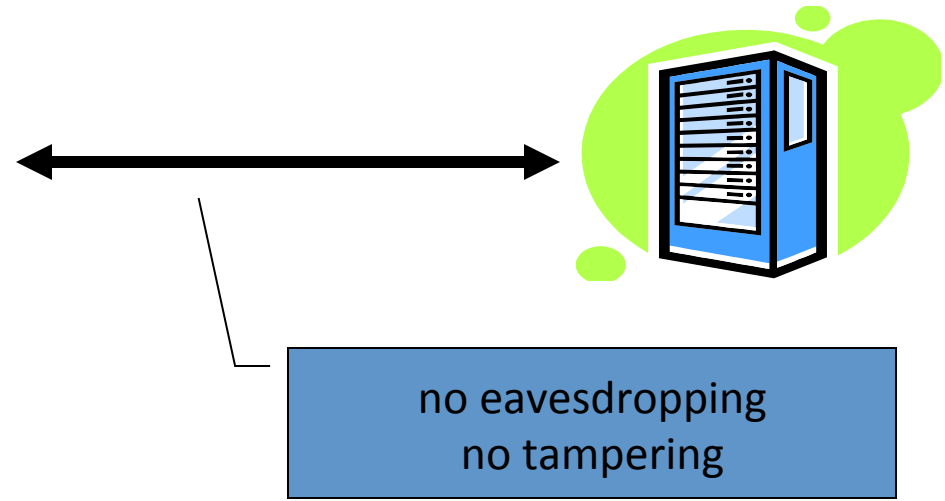
- A tremendous tool
- The basis for many security mechanisms

Is not:

- The solution to all security problems
- Reliable unless implemented and used properly
- Something you should try to invent yourself
    - Need to subject your designs to outside review
    - Need considerable experience

Goal 1: Secure communication





Secure Sockets Layer / TLS

Standard for Internet security

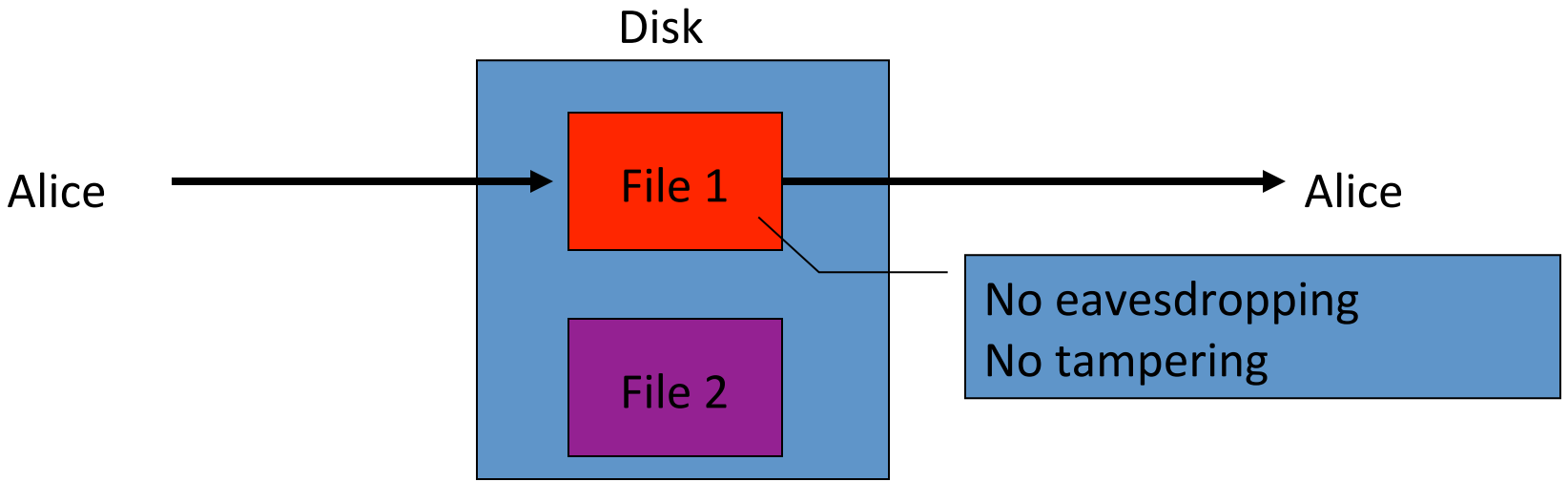
- Goal: "... provide privacy and reliability between two communicating applications"

Two main parts

1. Handshake Protocol: **Establish shared secret key using public-key cryptography**
2. Record Layer: **Transmit data using negotiated key**

   This module: Using a key for encryption and integrity

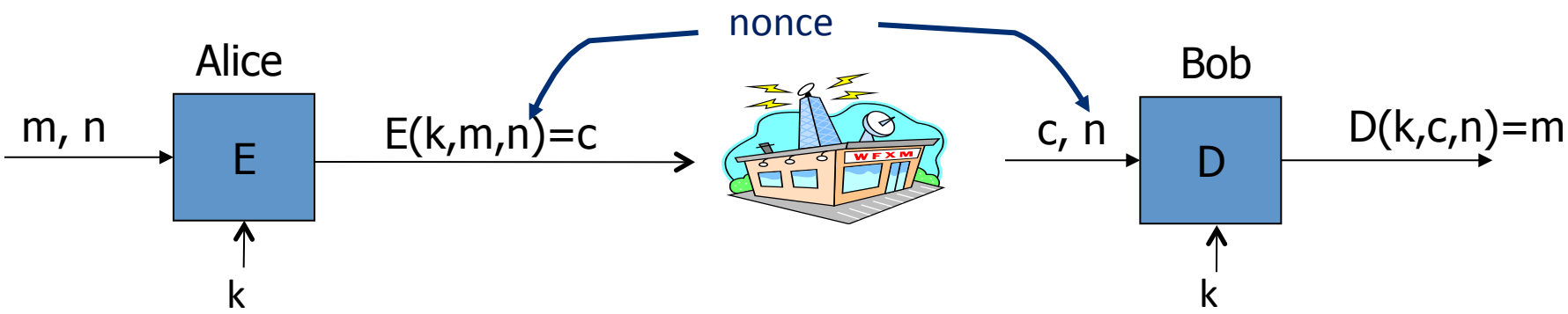
Goal 2: protected files



Analogous to secure communication:

Alice today sends a message to Alice tomorrow

Building block: sym. encryption



E, D: cipher k: secret key (e.g. 128 bits)

m, c: plaintext, ciphertext n: nonce (aka IV)

Encryption algorithm is publicly known

- Never use a proprietary cipher

Use Cases

Single use key: (one time key)

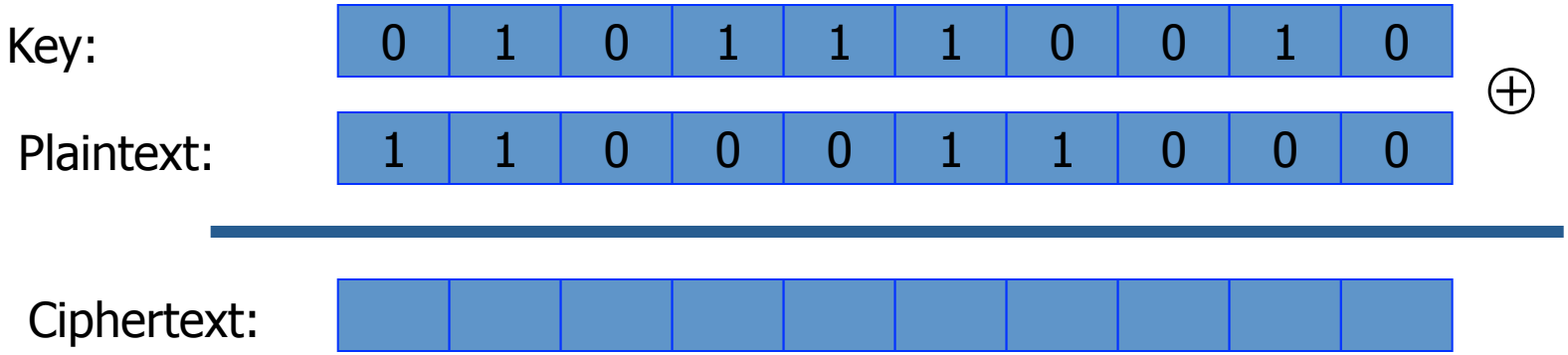
- Key is only used to encrypt one message
    - encrypted email: new key generated for every email
- No need for nonce (set to 0)

Multi use key: (many time key)

- Key used to encrypt multiple messages
    - SSL: same key used to encrypt many packets
- Need either *unique* nonce or *random* nonce

First example: One Time Pad (single use key)

Vernam (1917)



Shannon ‘49:

- OTP is “secure” against one-time eavesdropping

The OTP encryption formula is $c = E(k, m) = m \oplus k$

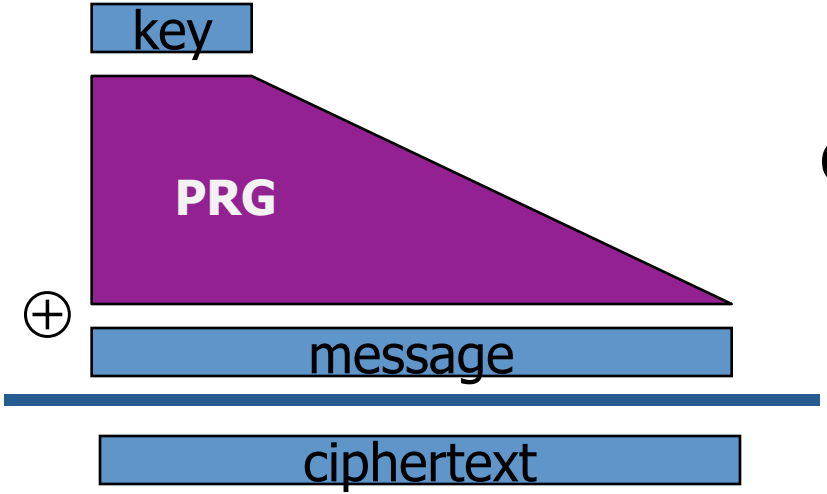
What is the decryption formula?

- $D(k, c) = k + c$
- $D(k, c) = k \times c$
- $D(k, c) = k \oplus c$
- $D(k, c) = k \div c$

Stream ciphers (single use key)

Problem: OTP key is as long the message

Solution: Pseudo random key -- stream ciphers



$c \leftarrow \mathbf{PRG}(k) \oplus m$

Examples: **Salsa20/12** (643MB/s) , **Sosemanuk** (727MB/s), **RC4** (126MB/s)

Dangers in using stream ciphers

One time key !! "Two time pad" is insecure:

$C_1 \leftarrow m_1 \oplus PRG(k)$

$C_2 \leftarrow m_2 \oplus PRG(k)$

Eavesdropper does:

$C_1 \oplus C_2 \rightarrow$ 

Enough redundant information in English that:

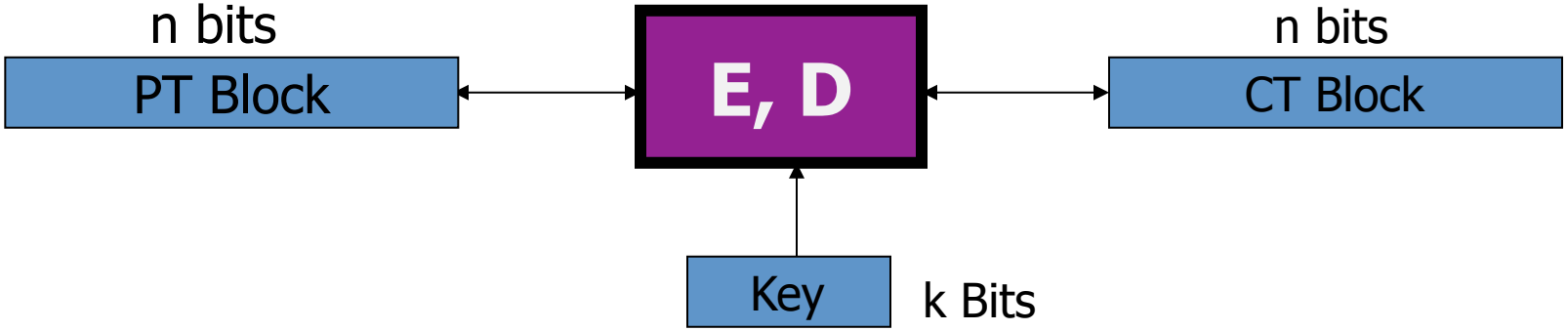
$m_1 \oplus m_2 \rightarrow m_1, m_2$

End of Segment

Crypto concepts

Block ciphers

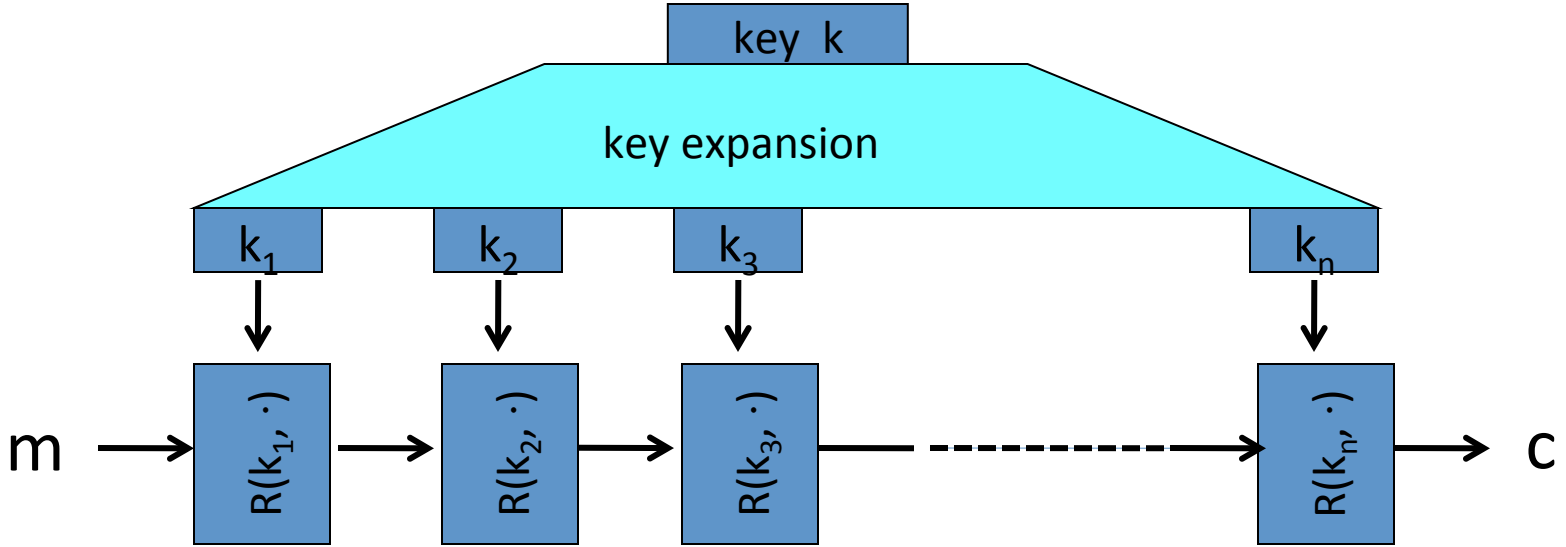
Block ciphers: crypto work horse



Canonical examples:

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

Block Ciphers Built by Iteration



R(k,m): round function

for 3DES (n=48), for AES-128 (n=10)

Standard Block Ciphers

Input: (m, k)

Repeat simple mixing operation several times

- 3**DES**: Repeat 48 times:

$$\begin{cases} m_L \leftarrow m_R \\ m_R \leftarrow m_L \oplus F(k_i, m_R) \end{cases}$$

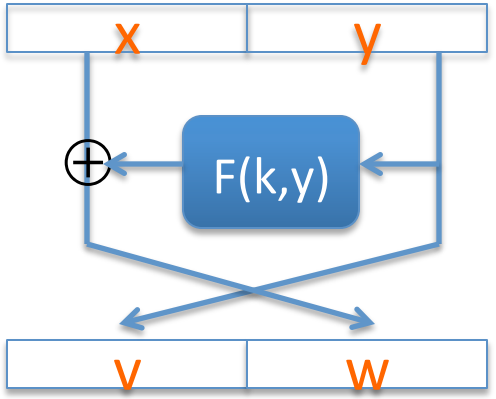
- **AES-128**: Mixing step repeated 10 times

Difficult to design: must resist subtle attacks

- differential attacks, linear attacks, brute-force, …

What is the inverse of the DES round function?

- ○ $(x, y) = (v \oplus F(k,w) \ , \ w)$
- ○ $(x, y) = (w \oplus F(k,v) \ , \ v)$
- ○ $(x, y) = (v \ , \ w \oplus F(k,v))$
- ○ $(x, y) = (w \oplus F(k,w) \ , \ v)$



$(v, w) = (y \ , \ x \oplus F(k,y))$

Abstract Block Ciphers: PRPs and PRFs

PRF: F: $K \times X \rightarrow Y$ such that:

exists "efficient" algorithm to eval. F(k,x)

PRP: E: $K \times X \rightarrow X$ such that:

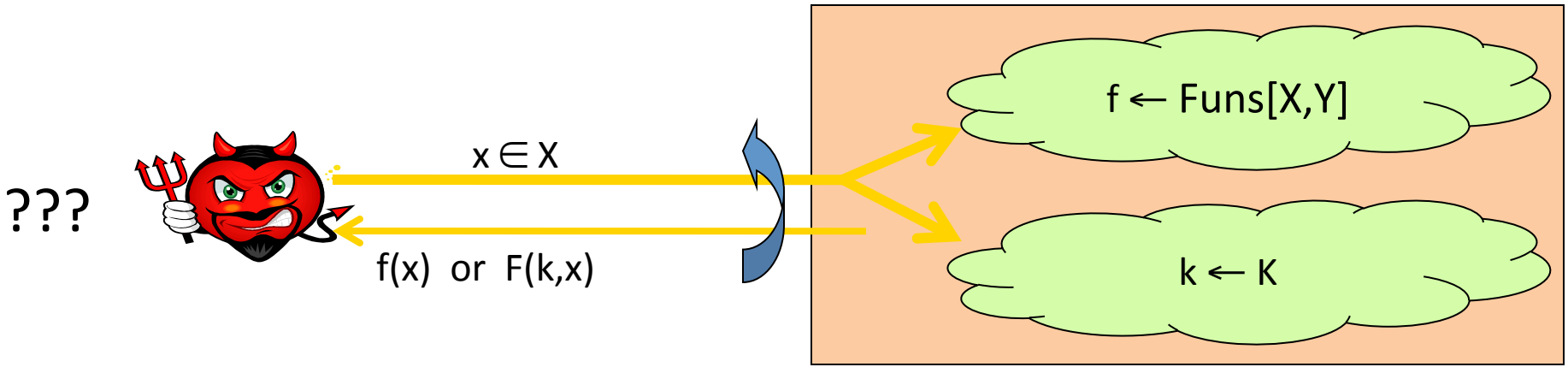
1. Exists "efficient" algorithm to eval. E(k,x)
2. The func $E(k, \cdot)$ is one-to-one
3. Exists "efficient" algorithm for inverse D(k,x)

A block cipher is a PRP

Secure PRF and Secure PRP

- A **PRF** $F: K \times X \rightarrow Y$ is secure if
  $F(k, \cdot)$ is indistinguishable from a random func. $f: X \rightarrow Y$

- A **PRP** $E: K \times X \rightarrow X$ is secure if
  $E(k, \cdot)$ is indisting. from a random perm. $\pi: X \rightarrow X$



What means indistinguishable?

- Secure PRF/PRP $\rightarrow$ indistinguishable from random function/ permutation
- (Efficient) statistical tests

- Advantage

PRF Switching Lemma

PRF Switching lemma:

A secure PRP is also a secure PRF

$\Rightarrow$ AES and 3DES are secure PRFs

Suppose F(k,x) is a secure PRF.

Is the following G a secure PRF?

$$G(k, x) = \begin{cases} 0 & \text{if } x=0 \\ F(k,x) & \text{otherwise} \end{cases}$$

- No, it is easy to distinguish G from a random function
- Yes, an attack on G would also break F
- It depends on F

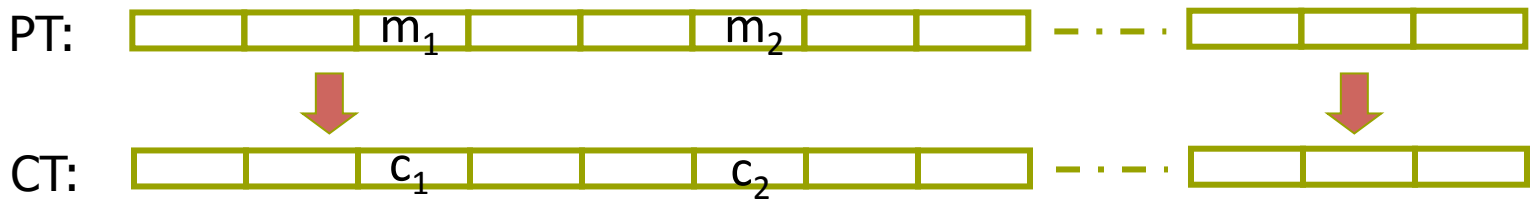
End of Segment

Crypto concepts

Using block ciphers

Incorrect use of block ciphers

Electronic Code Book (ECB):



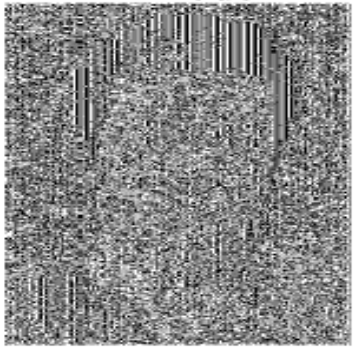
Problem:

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

In pictures

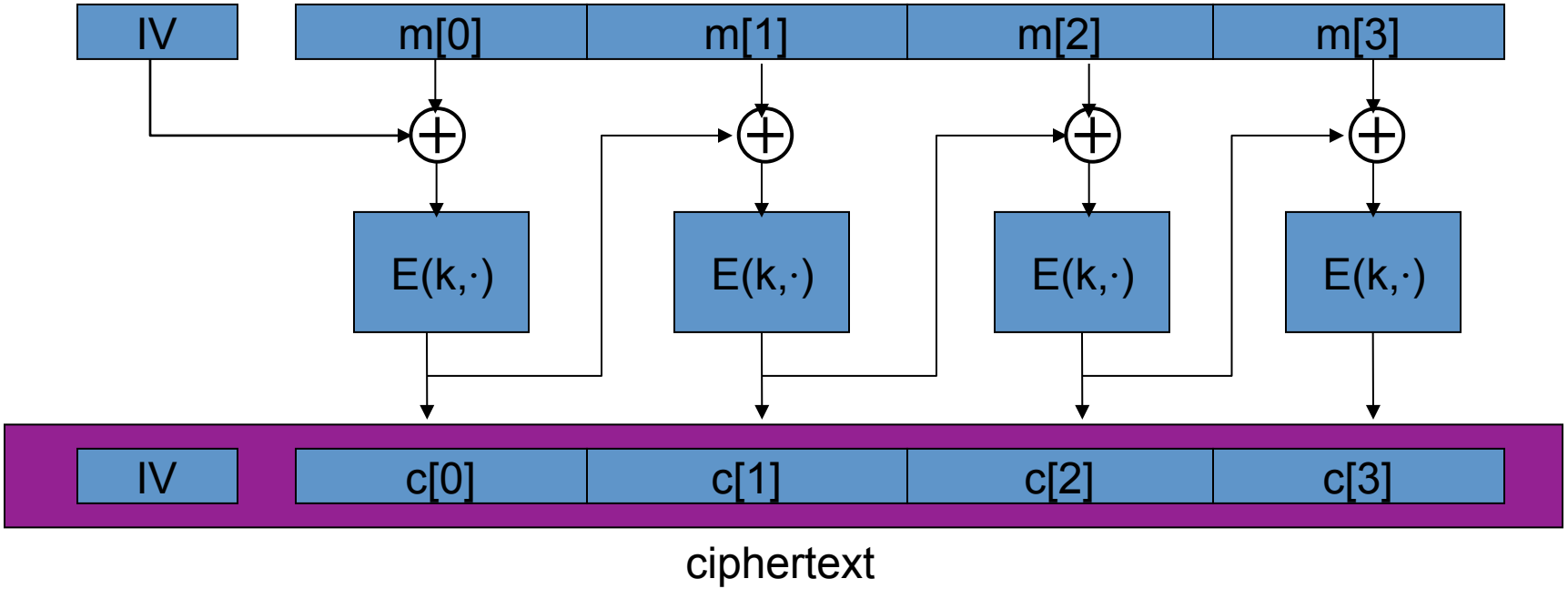
An example plaintext

Encrypted with AES in ECB mode



Eavesdropping security 1: CBC mode

E a secure PRP. Cipher Block Chaining with IV:



Use cases: how to choose an IV

Single use key: no IV needed (IV=0)

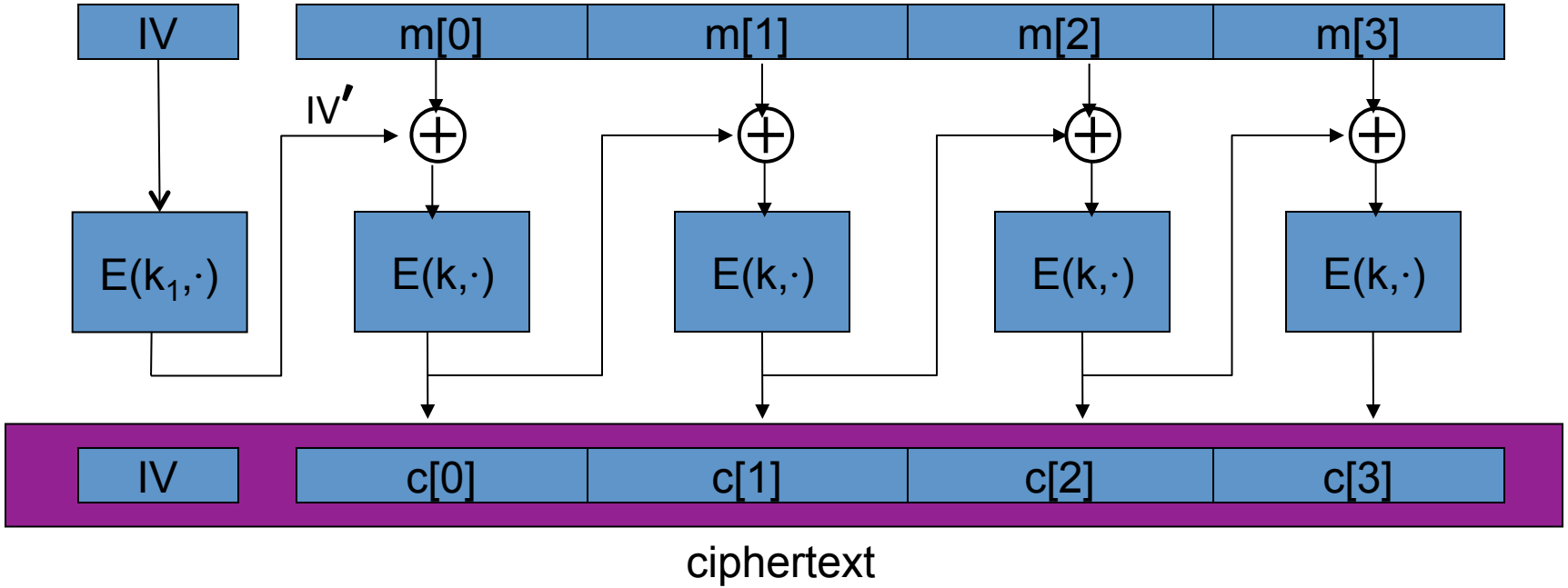
Multi use key: (CPA Security)

- Best: use a fresh *random* IV for every message (IV ← X)
- Can use *unique* IV (e.g counter)
  - but then first step in CBC must be IV' ← E(k,IV)
  - benefit: may save transmitting IV with ciphertext

CBC with Unique IVs (nonce-based encryption)

Cipher Block Chaining with unique IV: key = (k,k_1)

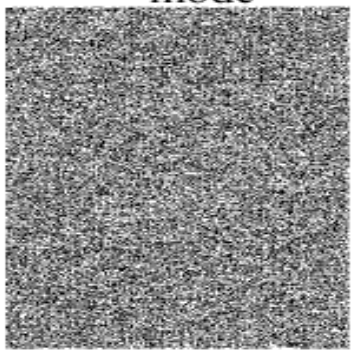
unique IV means: (key,IV) pair is used for only one message



In pictures

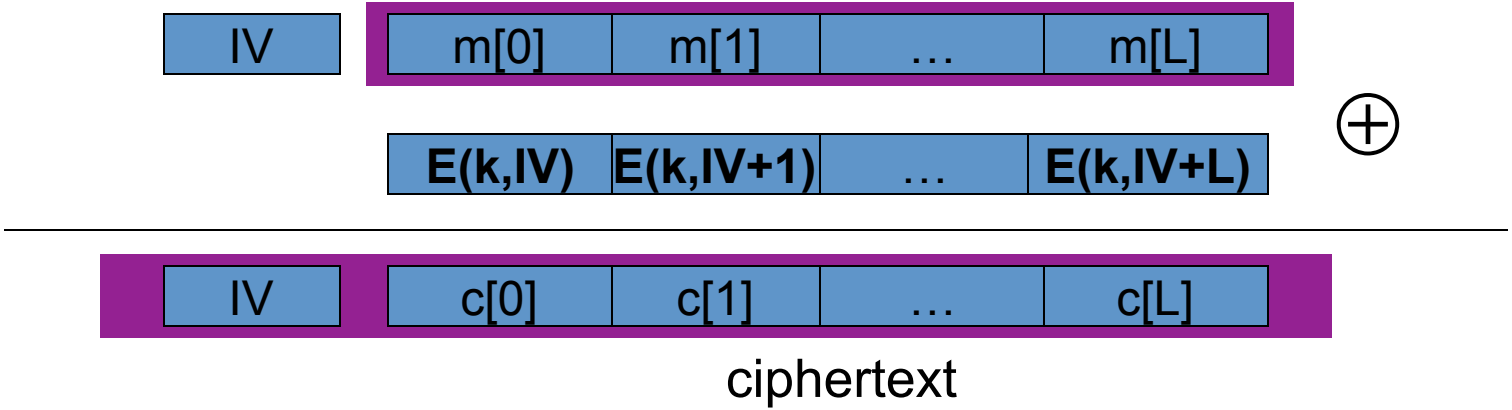
An example plaintext

Encrypted with AES in CBC mode



Eavesdropping security 2: CTR mode

Counter mode with a random IV: (parallel encryption)



Why are these modes secure? See the crypto course.

Performance:

Crypto++ 5.6.0 [Wei Dai]

AMD Opteron, 2.2 GHz (Linux)

	Cipher	Block/key size	Speed (MB/sec)
stream	Salsa20/12		643
	Sosemanuk		727
block	3DES	64/168	13
	AES	128/128	109

A Warning

eavesdropping security is insufficient for most applications

Need also to defend against active attacks.

CBC and CTR modes are insecure against active attacks

Next: methods to ensure message integrity

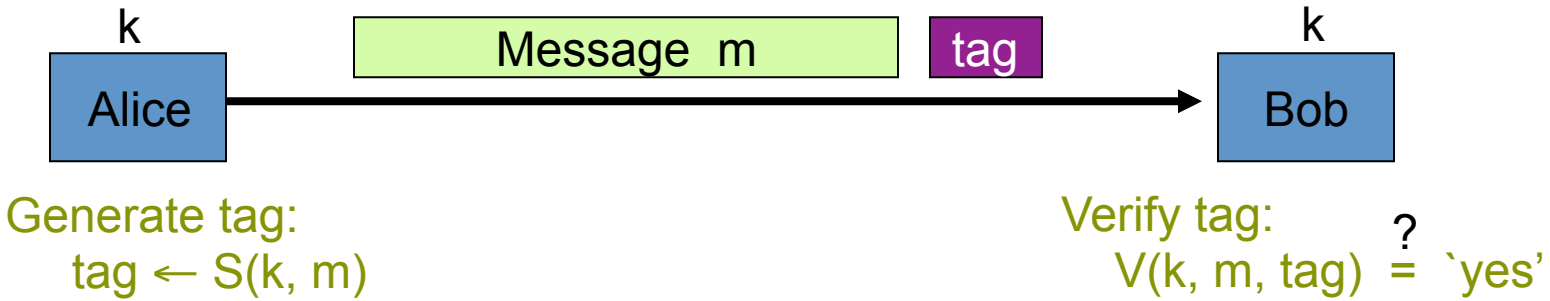
End of Segment

Crypto concepts

Message Integrity

Message Integrity: MACs

- Goal: provide message integrity. No confidentiality.
  - ex: Protecting public binaries on disk.



note: non-keyed checksum (CRC) is an insecure MAC !!

Secure MACs

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.

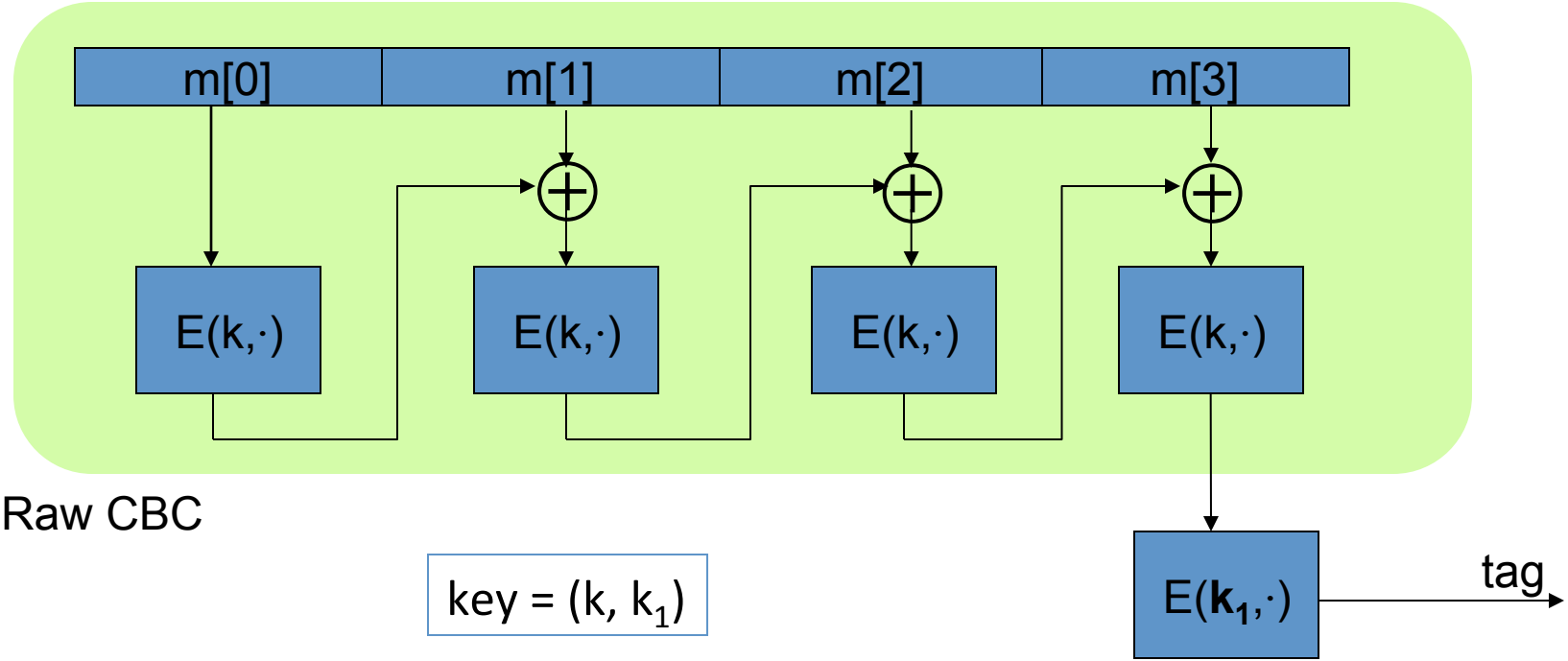
- produce some **<u>new</u>** valid message/tag pair (m,t).

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

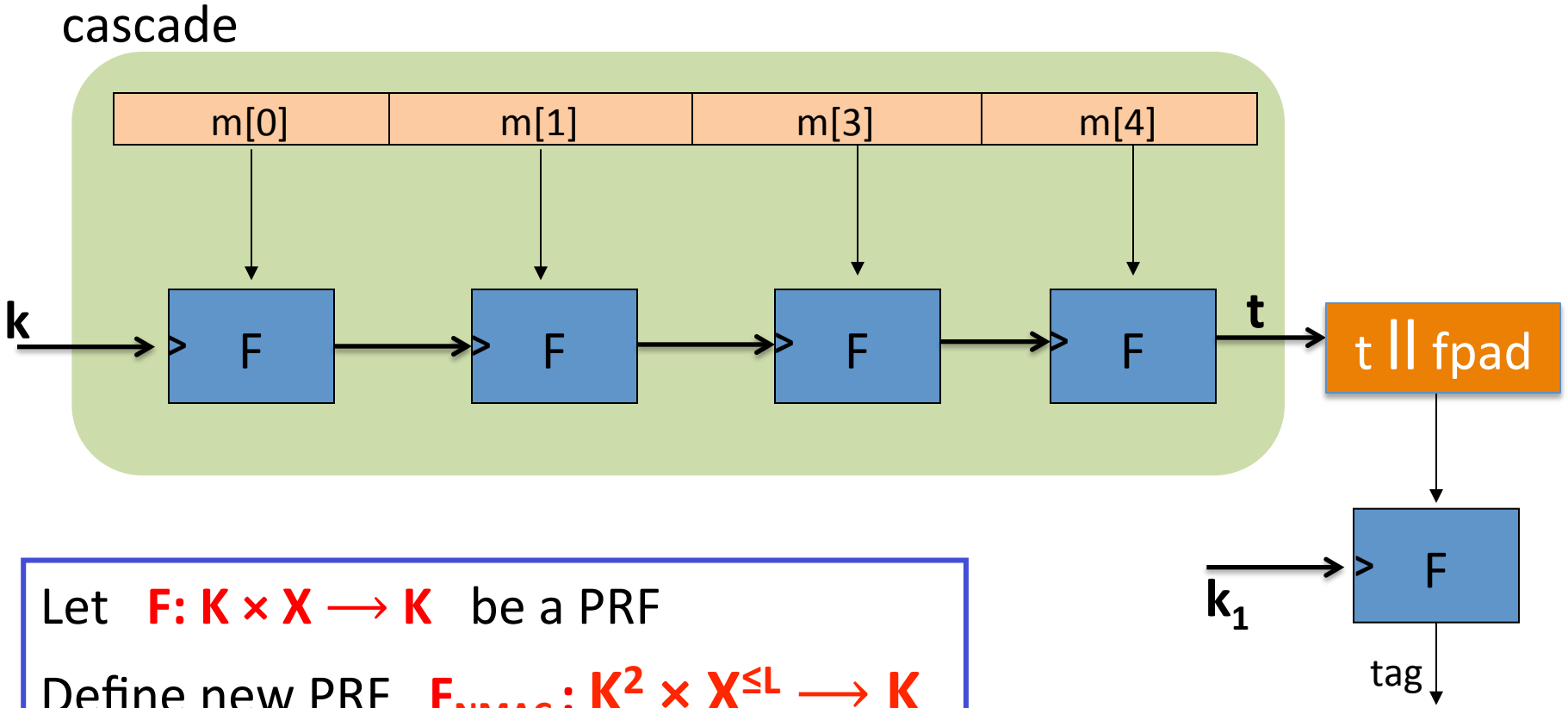
A secure PRF gives a secure MAC:

- $S(k,m) = F(k,m)$
- $V(k,m,t)$: output `yes' if $t = F(k,m)$ and `no' otherwise.

Construction 1: ECBC (encrypted MAC)



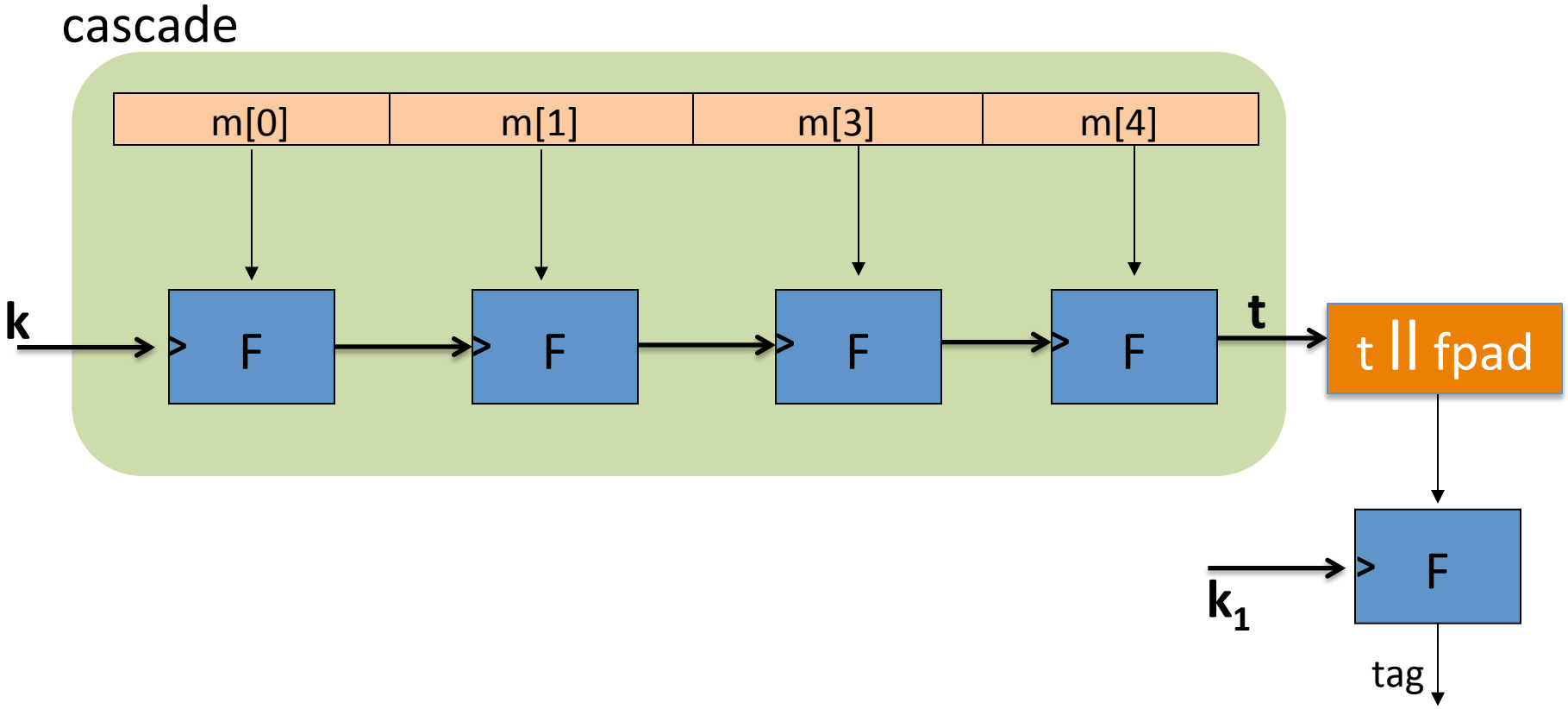
Construction 2: NMAC (nested MAC)



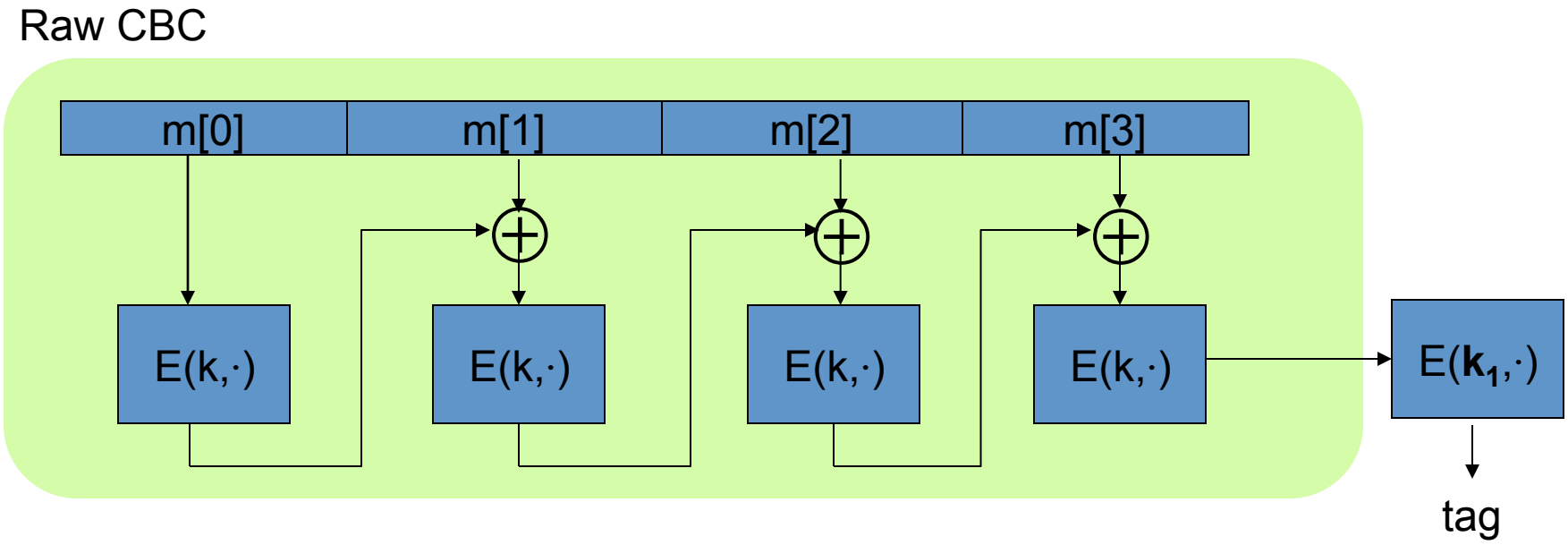
Let $F: K \times X \longrightarrow K$ be a PRF

Define new PRF $F_{NMAC}: K^2 \times X^{\leq L} \longrightarrow K$

Importance of last step (NMAC)



Importance of last step (ECBC)



Construction 3: HMAC (Hash-MAC)

Most widely used MAC on the Internet.

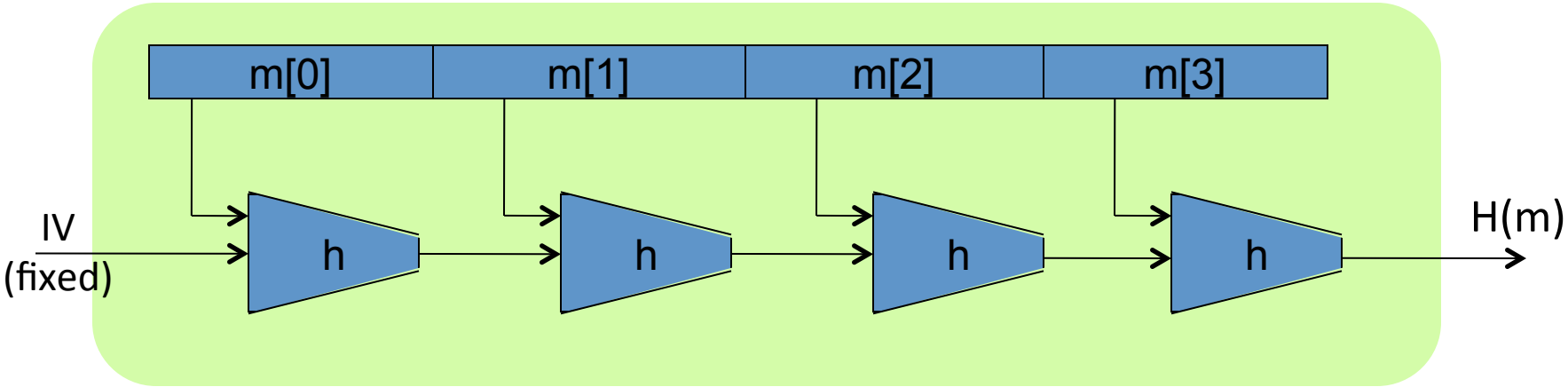
H: hash function.
example: SHA-256 ; output is 256 bits

Building a MAC out of a hash function:

– Standardized method: HMAC

$$S(k, m) = H(k \oplus opad \ , \ \mathbf{H(k \oplus ipad \ , \ m)})$$

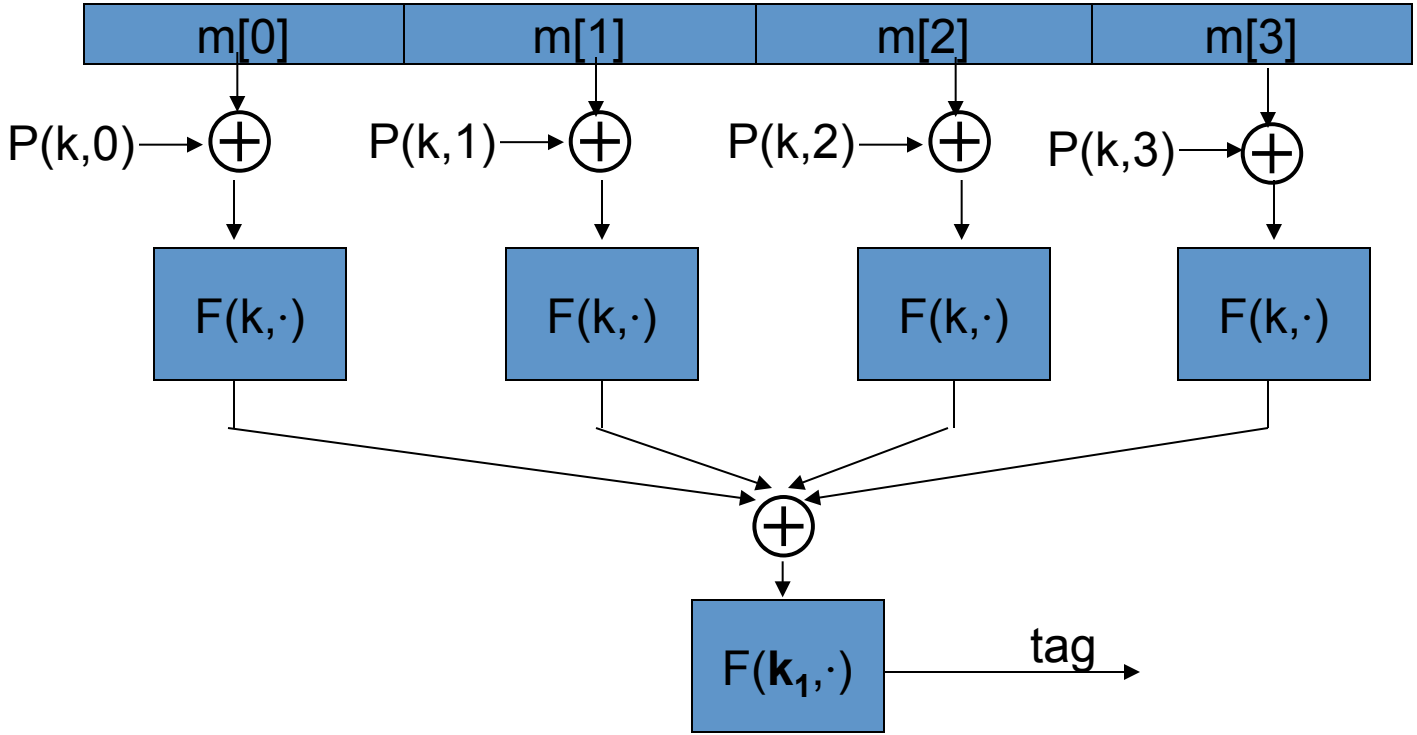
SHA-256: Merkle-Damgard



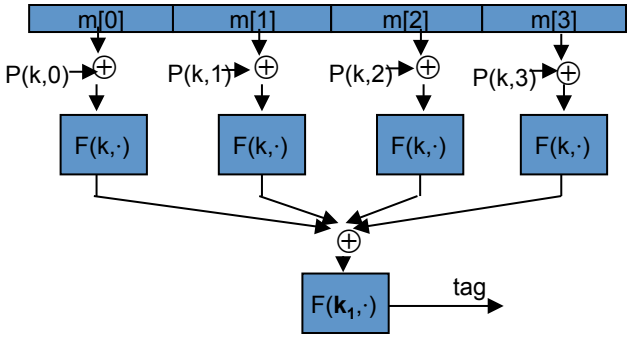
- h(t, m[i]): compression function
- Thm 1: if h is collision resistant then so is H
- "Thm 2": if h is a PRF then HMAC is a PRF

Construction 4: PMAC -- a parallel MAC

ECBC and HMAC are sequential. PMAC:



PMAC



Suppose the P(k,·) function was not used.

[i.e. P(k, ·) = 0]

Would PMAC be a secure MAC?

- No. Given tag on (m[0],m[1]) attacker obtains tag on (m[1],m[0])
- No. Without P() an attacker could obtain the secret key k
- It depends on what F is used

End of Segment

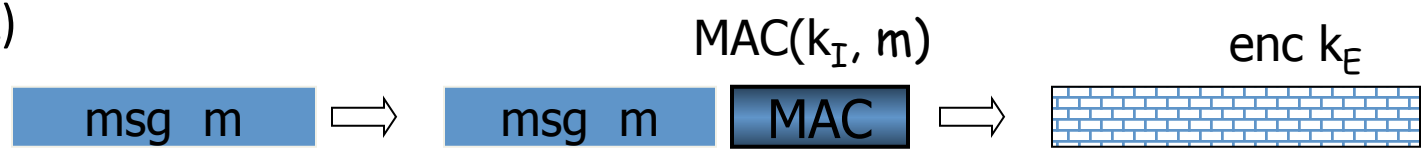
Crypto concepts

Authenticated Encryption

Combining MAC and ENC (CCA)

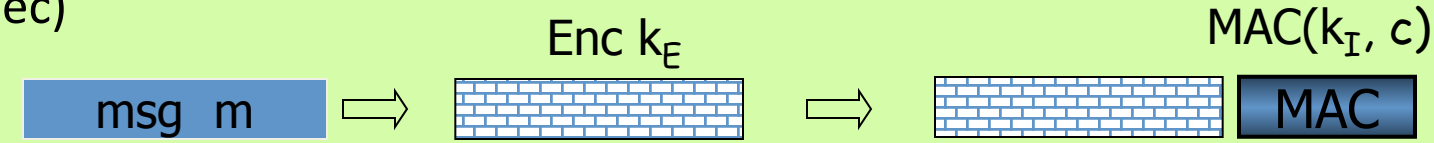
Encryption key k_E. MAC key = k_I

Option 1: (SSL)

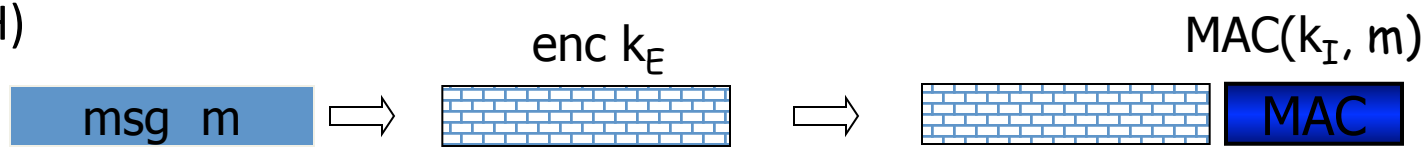


Option 2: (IPsec)

always correct



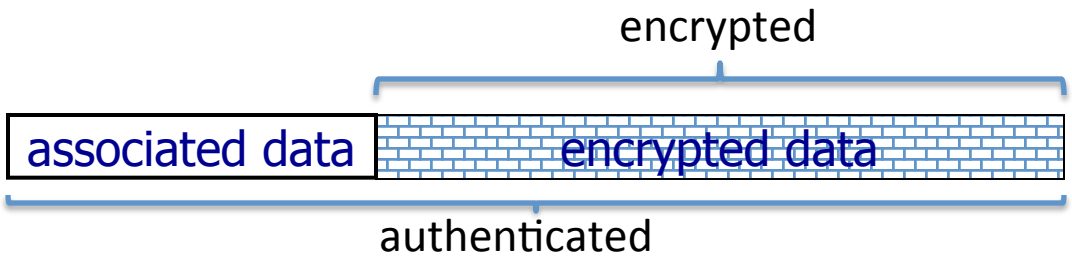
Option 3: (SSH)



Standards (at a high level)

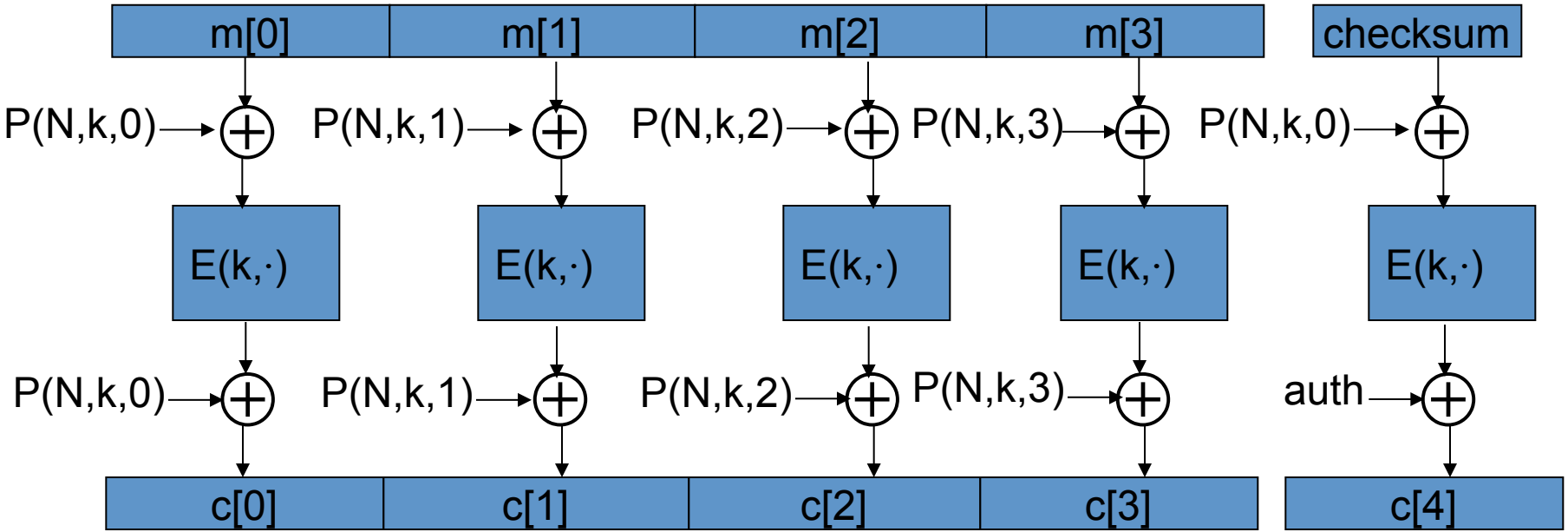
- CCM: CBC-MAC then CTR mode encryption
- GCM: CTR mode encryption then MAC
- EAX: CTR mode encryption then OMAC

All support AEAD: (auth. enc. with associated data)



OCB

More efficient authenticated encryption

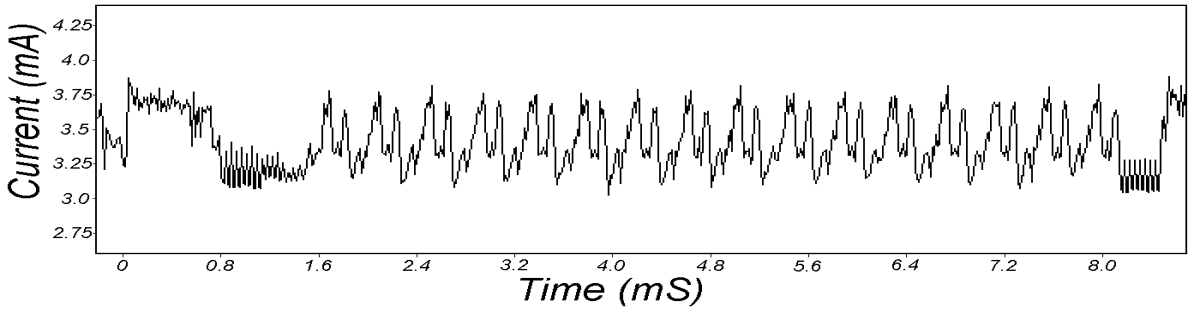


Final words

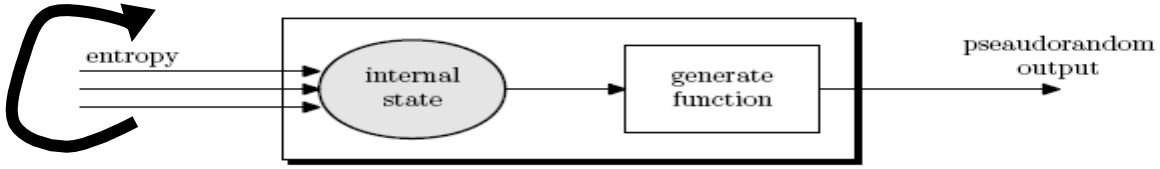
Implementation attacks

Power cryptanalysis. (Kocher-Jaffe-Jun 99)

- Power consumption depends on instruction and data
- Measure power consumption during block cipher operation
- About 1000 ciphertexts suffice to expose secret key.



Generating Randomness (e.g. keys, IV)



Pseudo random generators in practice: (e.g. /dev/random)

- Continuously add entropy to internal state
- Entropy sources:
  - Hardware RNG: Intel **RdRand** inst. (Ivy Bridge). 3Gb/sec.
  - Timing: hardware interrupts (keyboard, mouse)

NIST SP 800-90: NIST approved generators

Summary

Shared secret key:

- Used for secure communication and document encryption

Encryption: (CPA security) **[should not be used standalone]**

- One-time key: stream ciphers, CBC or CTR with fixed IV
- Many-time key: CBC or CTR with random IV

Integrity: ECBC or HMAC or PMAC

Authenticated encryption: encrypt-then-MAC

End of Segment