CS 4770: Cryptography

CS 6750: Cryptography and Communication Security

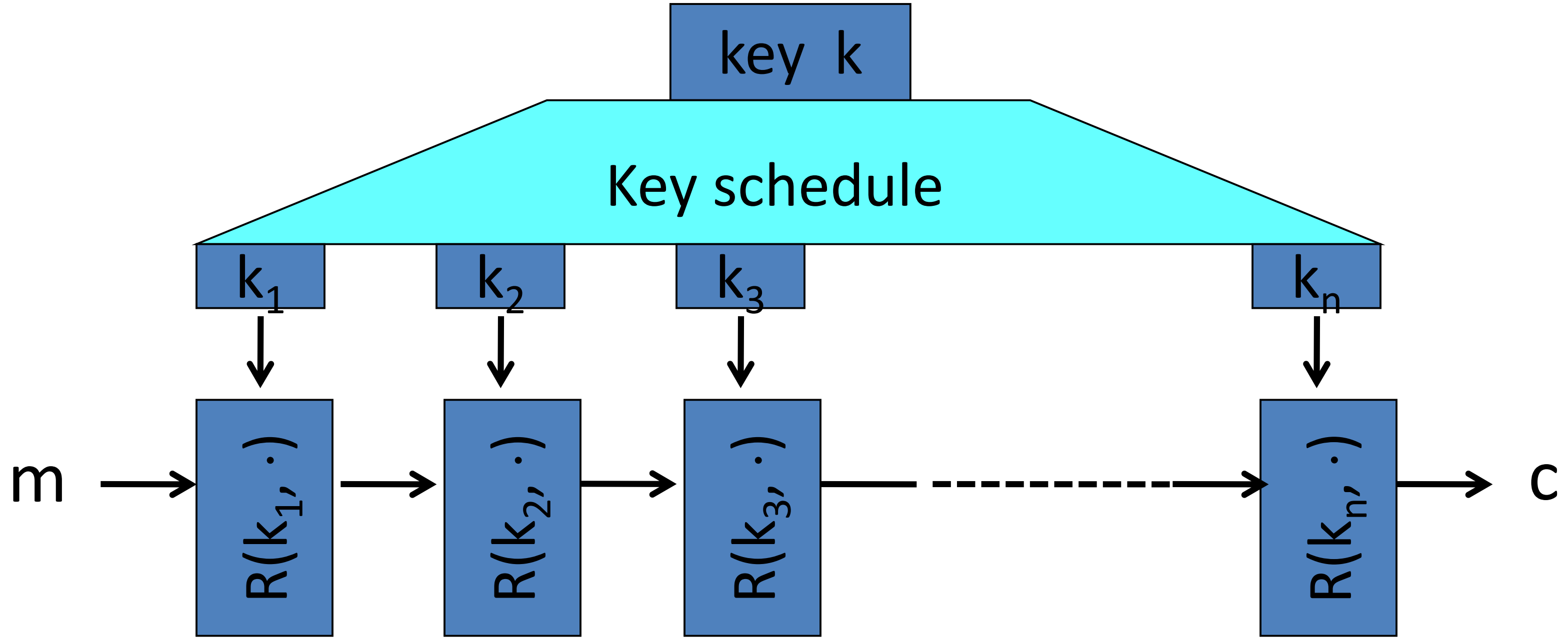
Alina Oprea
Associate Professor, CCIS
Northeastern University

February 8 2018

Review

- CPA-secure construction
  - Security proof by reduction to PRF
  - Randomized
- How to design block ciphers
  - Substitution Permutation Networks
  - Feistel Networks
  - Multiple rounds
- DES
  - Feistel Network
- AES
  - Substitution Permutation Network

Block Ciphers Built by Iteration



R(k,m) is called a *round function*

for DES (n=16), for AES-128 (n=10)

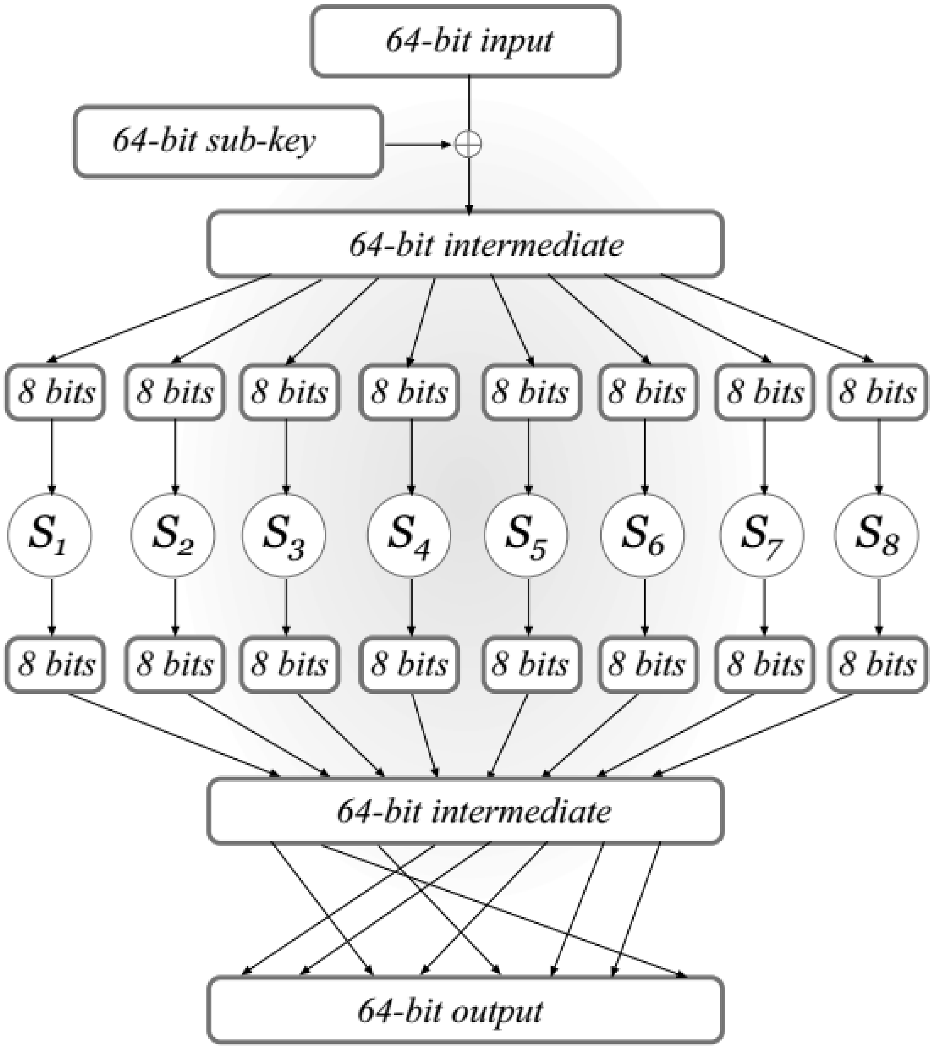
Substitution-Permutation Network

Round key

S-box

Fixed permutation

Invertible



Key mixing

Substitution

Permutation

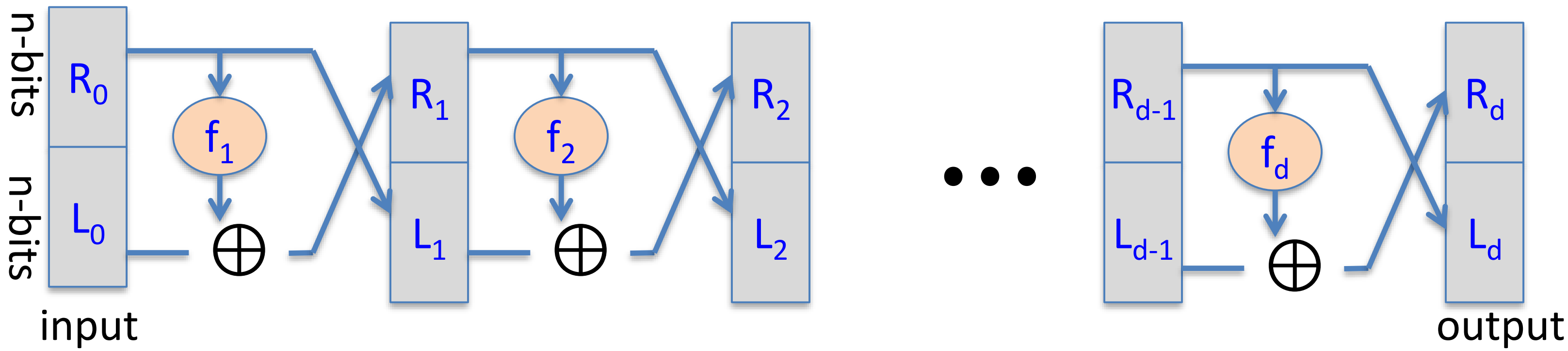
S boxes and mixing permutation are public

Feistel Networks

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

Often $f_i(x) = F_{k_i}(x)$, for k_i secret keys and F a PRF

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

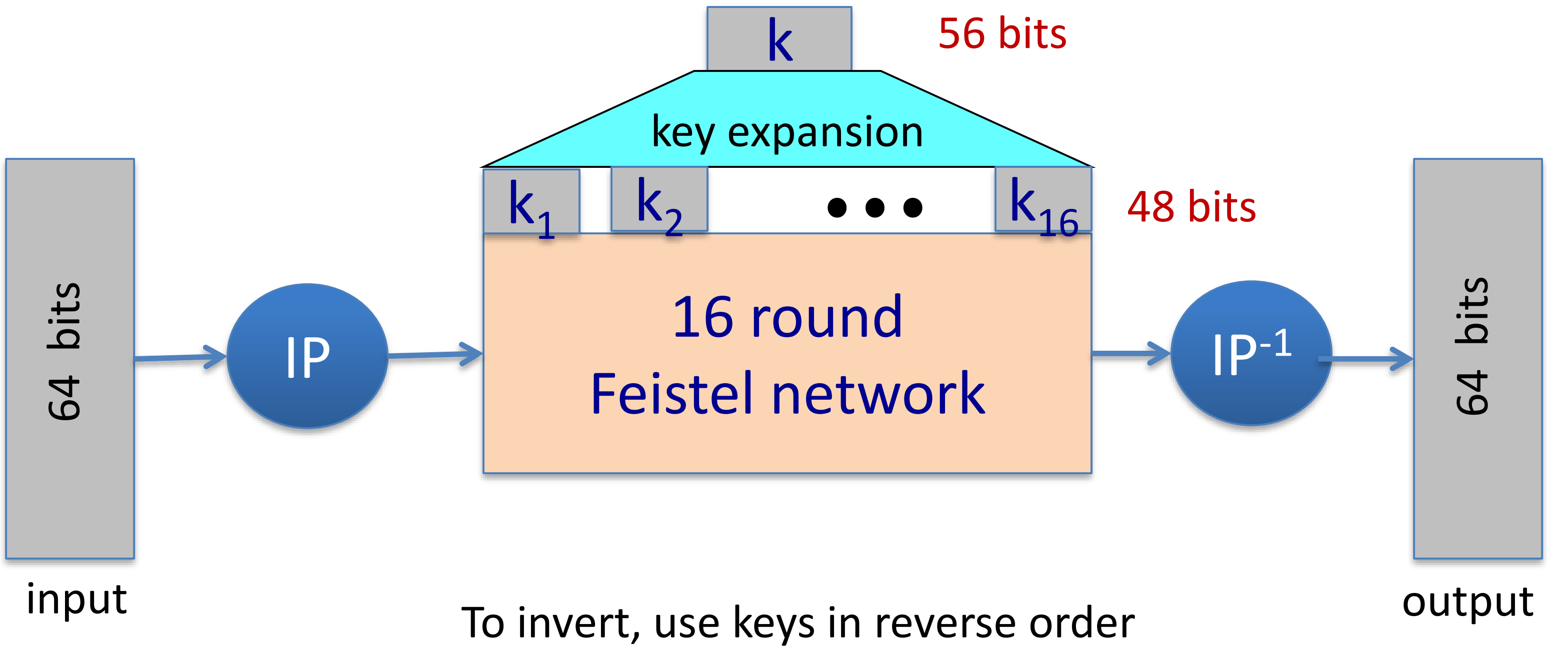


$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f_i(R_{i-1})$$

- Functions f_i are public
- Round key is derived from main key and secret
- Advantage: f_i not invertible!

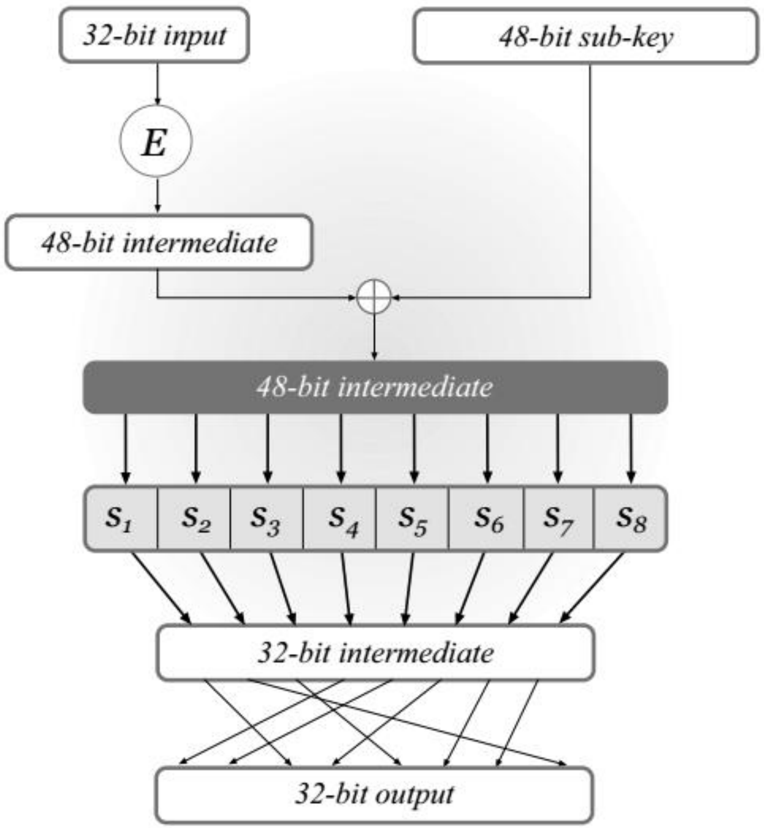
DES: 16 round Feistel network

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



The function $F(k_i, x)$

Substitution-Permutation Network



Key mixing

Substitution

Permutation

S-box: function $\{0,1\}^6 \longrightarrow \{0,1\}^4$, implemented as look-up table.

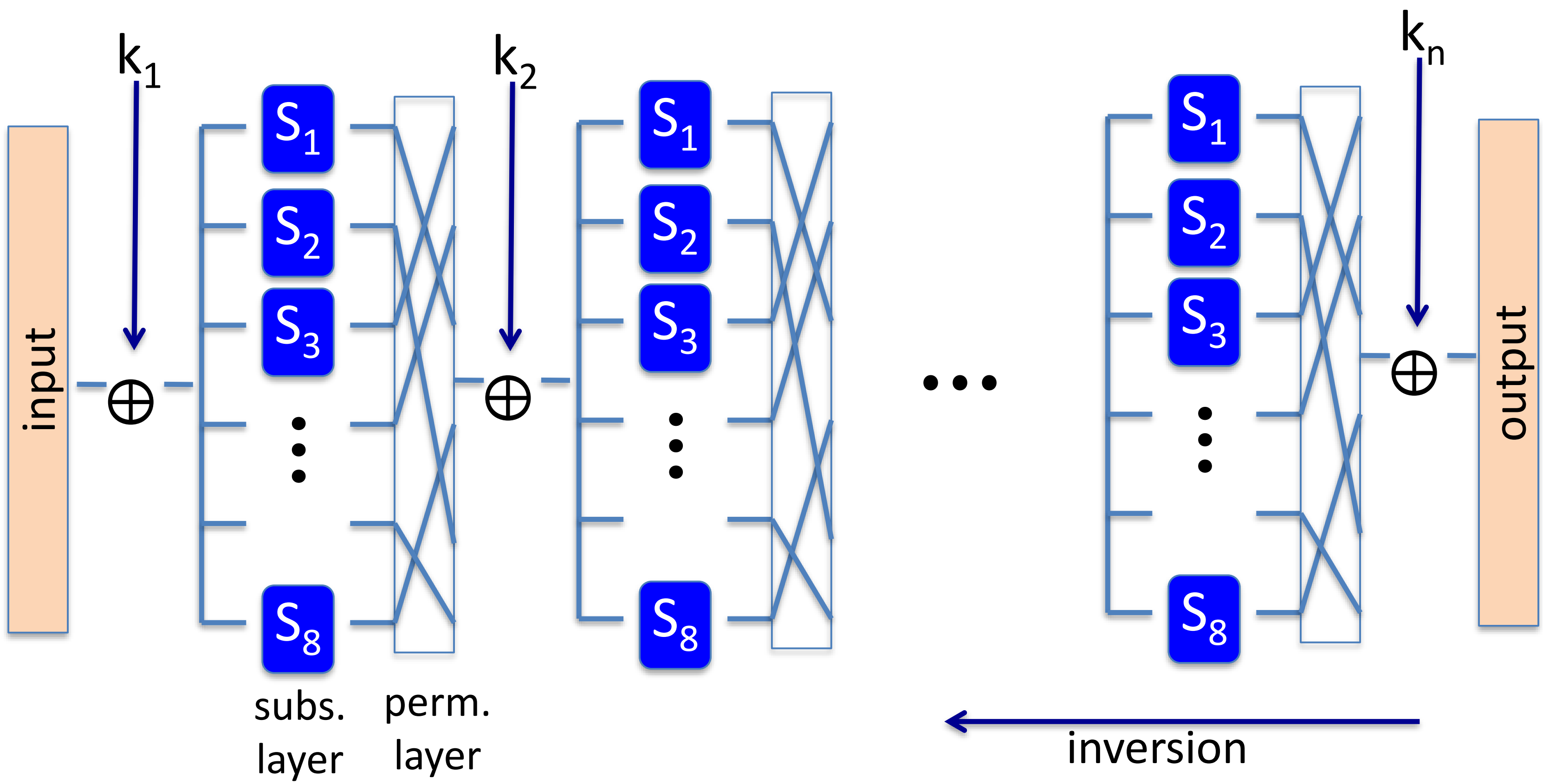
The AES process

- 1997: NIST publishes request for proposal
- 1998: 15 submissions. Five claimed attacks.
- 1999: NIST chooses 5 finalists
- 2000: NIST chooses Rijndael as AES (designed in Belgium)

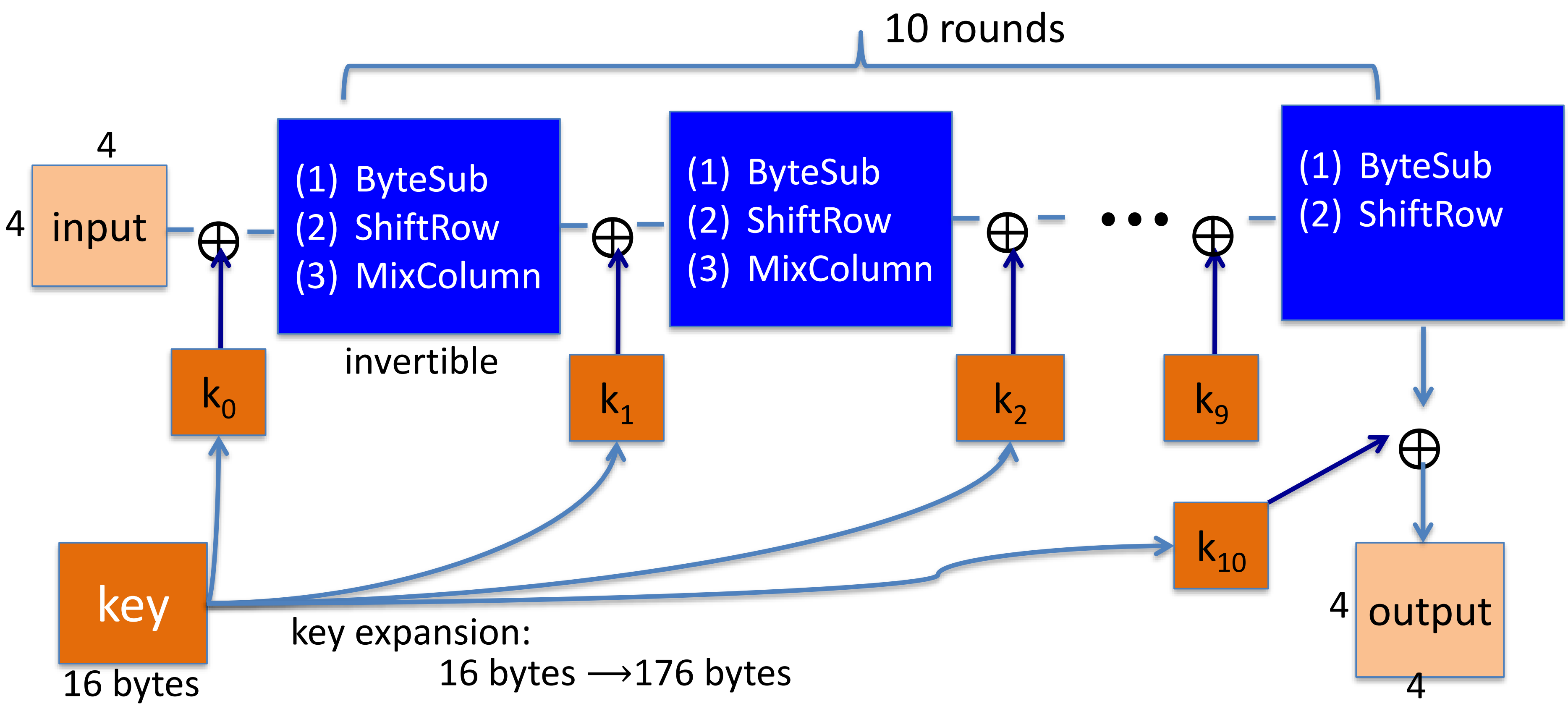
Key sizes: 128, 192, 256 bits.

Block size: 128 bits

AES is a Subs-Perm network (not Feistel)



AES-128 schematic

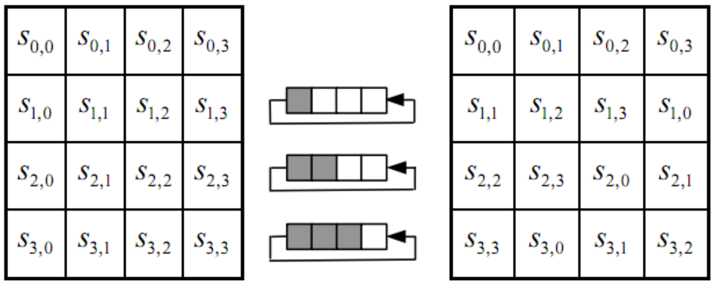


The round function

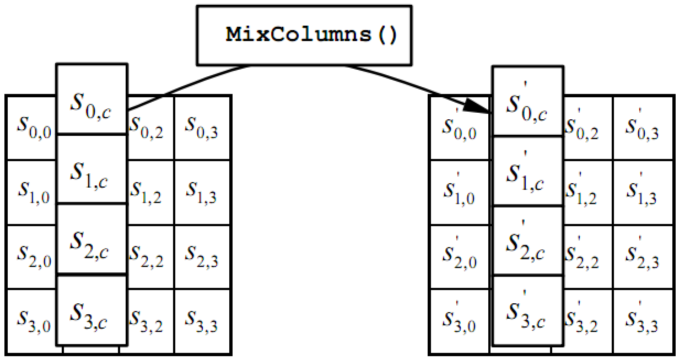
- **ByteSub**: a 1 byte S-box. 256 byte table (non- linear, but easily computable

$$A[i,j] \leftarrow S\left[A[i,j]\right], \forall i,j$$

- **ShiftRows**:



- **MixColumns**:



Code size/performance tradeoff

	Code size	Performance
Pre-compute round functions (24KB or 4KB)	largest	fastest: table lookups and xors
Pre-compute S-box only (256 bytes)	smaller	slower
No pre-computation	smallest	slowest

AES in hardware

AES instructions in Intel Westmere:

- **aesenc, aesenclast**: do one round of AES

128-bit registers: xmm1=state, xmm2=round key

aesenc xmm1, xmm2 ; puts result in xmm1

- **aeskeygenassist**: performs AES key expansion
- Claim 14 x speed-up over OpenSSL on same hardware

Similar instructions on AMD Bulldozer

Attacks

Best key recovery attack:

four times better than ex. search [BKR'11]

Related key attack on AES-256: [BK'09]

Given 2^{99} inp/out pairs from **four related keys** in AES-256

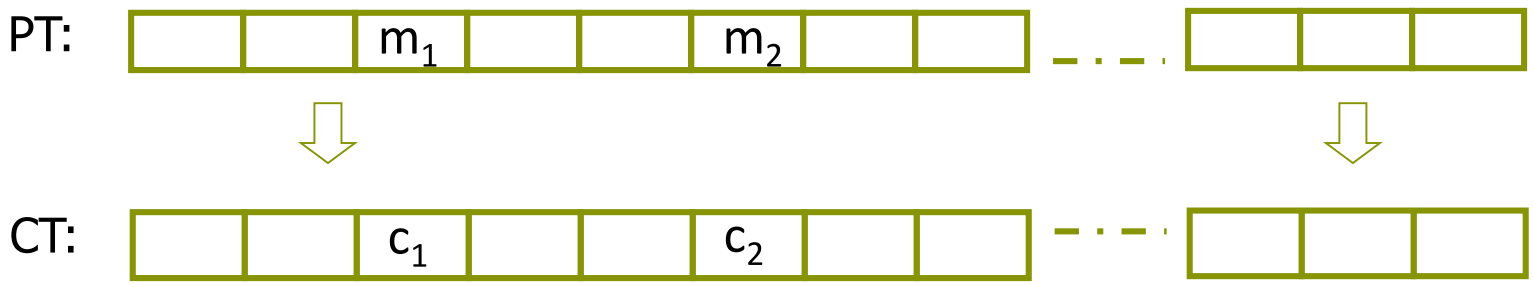
can recover keys in time $\approx 2^{99}$

Block ciphers

- Suggestions:
  - Don’t think about the inner-workings of AES and 3DES.
  - Don’t implement them yourselves
- We assume both are secure PRPs and will see how to use them

Incorrect use of block cipher

Electronic Code Book (ECB):

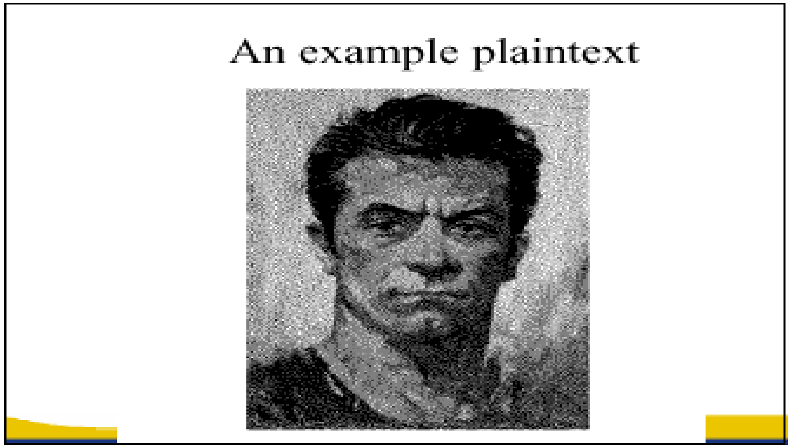


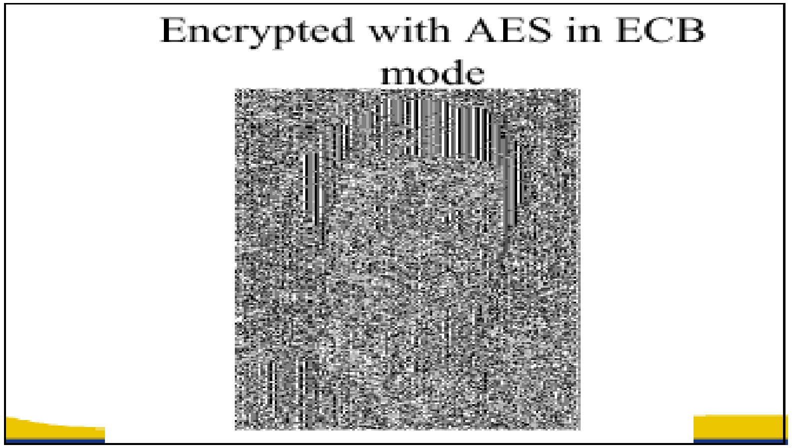
<u>Problem</u>:

– if $m_1 = m_2$ then $c_1 = c_2$

Not EAV-secure!

In pictures



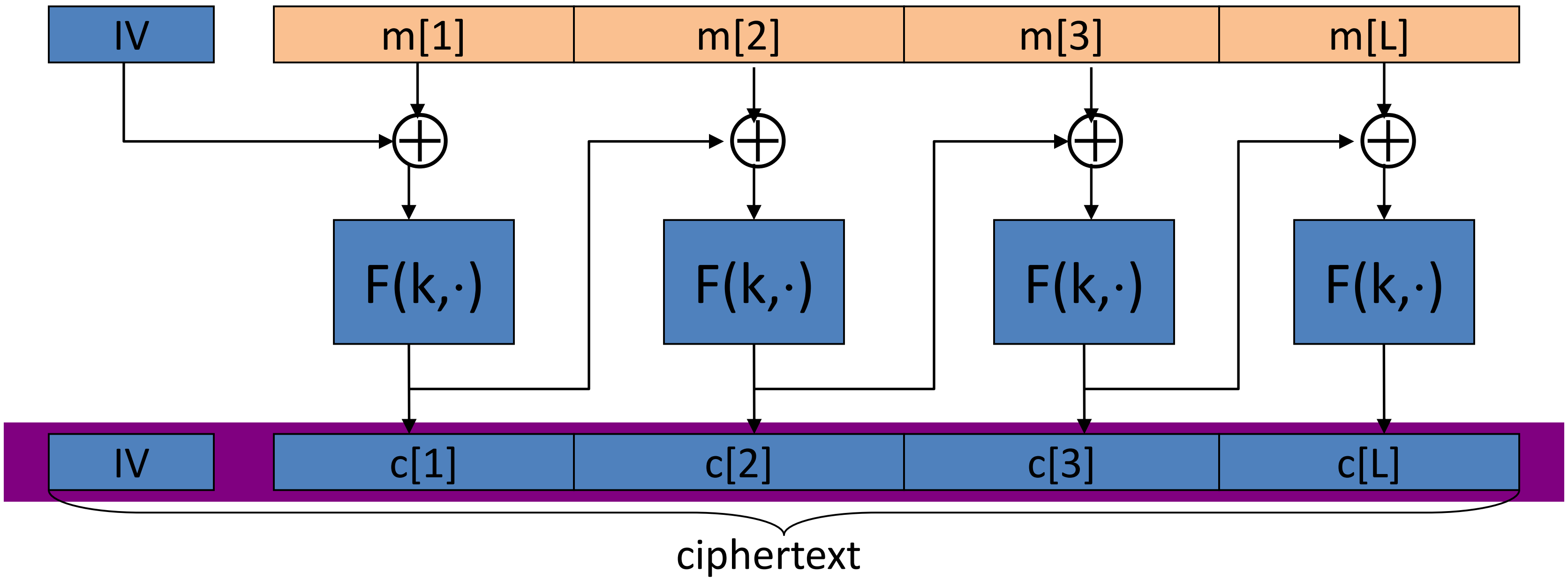


(courtesy B. Preneel)

CBC encryption

Let F be a PRP; F: K × $\{0,1\}^n \longrightarrow \{0,1\}^n$

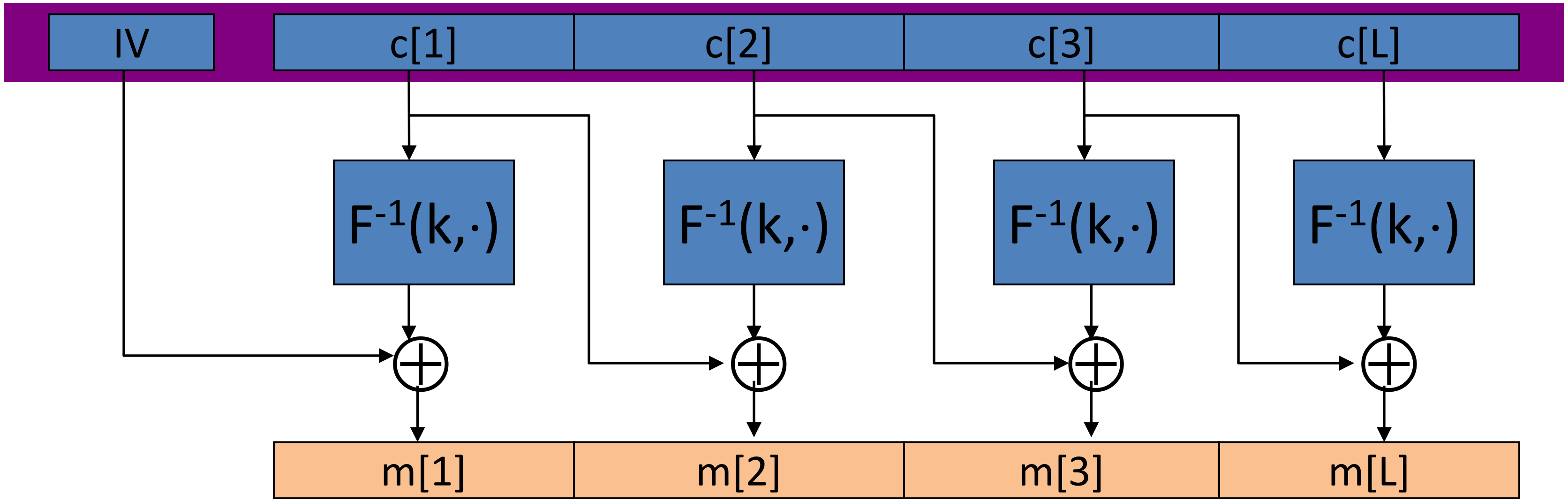
Enc_{CBC}(k,m): choose **random** IV∈ $\{0,1\}^n$ and do:



$$c_i = F_k(c_{i-1} \oplus m_i)$$

Decryption circuit

In symbols: $c[1] = F_k(\, IV \oplus m[1] \,)$ $\Rightarrow$ $m[1] =$



$$m_i = F^{-1}{}_k(c_i) \oplus c_{i-1}$$

CBC: CPA Analysis

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

If F is a secure PRP over (K, $\{0,1\}^n$) then

Enc_{CBC} is CPA-secure over (K, $\{0,1\}^{nL}$, $\{0,1\}^{n(L+1)}$).

In particular, for a q-query adversary A attacking Enc_{CBC} there exists a PRP adversary B s.t.:

$$\mathbf{Pr}[\mathrm{Exp}^{\mathrm{CPA}}_{\mathrm{Enc}_{\mathrm{CBC}},A}(n) = 1] \le 1/2 + 2\mathrm{Adv}^{\mathrm{PRP}}_{\mathrm{F},B} + 2\, q^2\, L^2\, /2^n$$

$$\mathrm{Adv}^{\mathrm{PRP}}_{\mathrm{E},B} = \left|\boldsymbol{Pr}\left[\boldsymbol{B}^{F_k(\cdot),F_k^{-1}(\cdot)}(\boldsymbol{n}) = 1\right] - \boldsymbol{Pr}[\boldsymbol{B}^{f(\cdot),f^{-1}(\cdot)}(\boldsymbol{n})\right|$$

Note: CBC is only secure as long as $\mathbf{q^2 L^2 << 2^n}$

An example

$$\Pr[\mathrm{Exp}^{\mathrm{CPA}}_{\mathrm{E}_{\mathrm{CBC}},A}(n) = 1] \leq 1/2 + \mathrm{Adv}^{\mathrm{PRP}}_{\mathrm{E},B} + 2\, q^2\, L^2 / 2^n$$

q = # messages encrypted with k

L = length of max message

Suppose we want $\Pr[\mathrm{Exp}^{\mathrm{CPA}}_{\mathrm{Enc}_{\mathrm{CBC}},A}(n) = 1] \leq 1/2 + 1/2^{32}$

$q^2\, L^2 / 2^n < 1/\, 2^{32}$

- AES: $2^n = 2^{128}$ ⇒ $q\,L < 2^{48}$

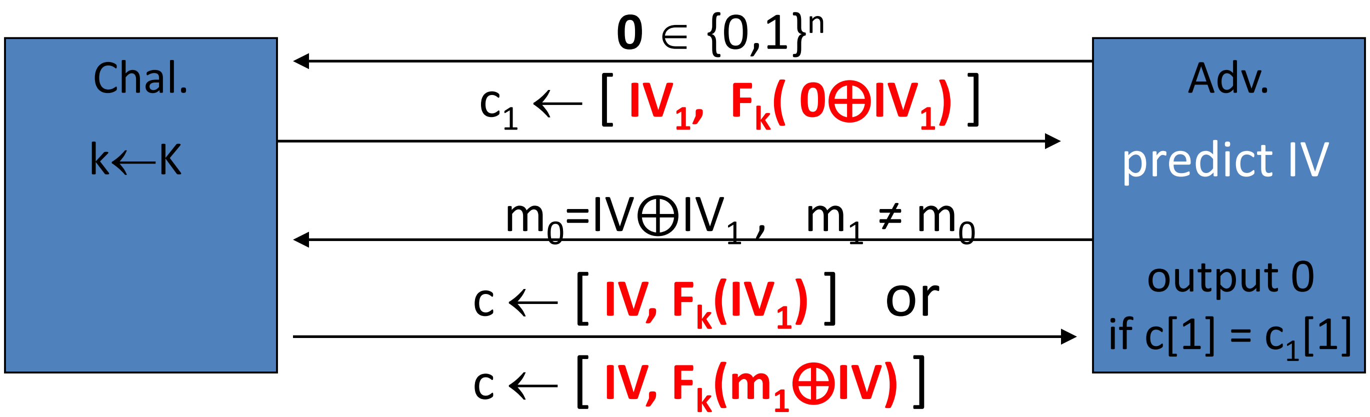
  So, after 2^{48} AES blocks, must change key

- 3DES: $2^n = 2^{64}$ ⇒ $q\,L < 2^{16}$

Attack on CBC with predictable IV

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

Suppose given $c \longleftarrow \text{Enc}_{CBC}(k,m)$ can predict next IV

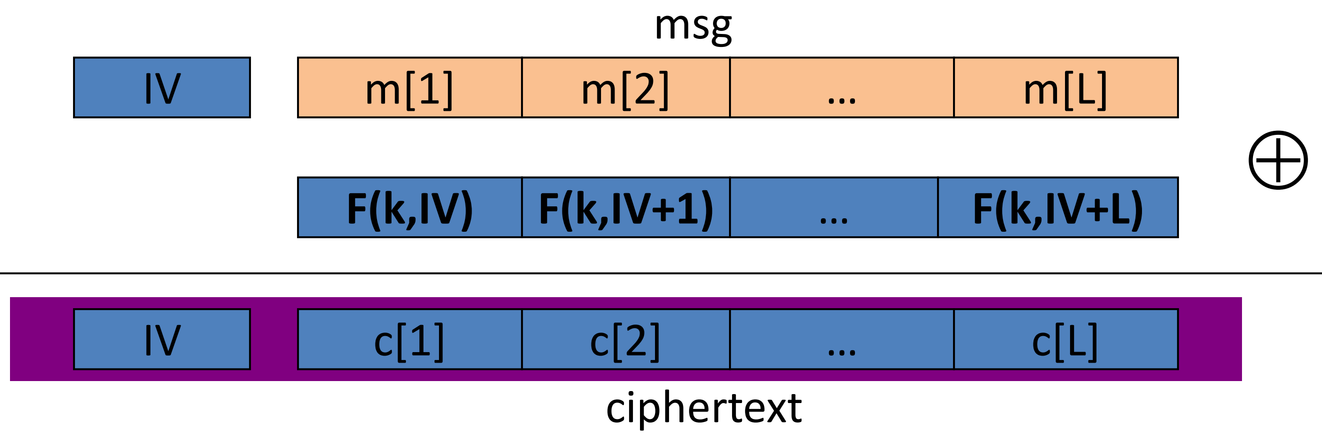


Bug in SSL/TLS 1.0: IV for record #i is last CT block of record #(i-1)

CTR-mode encryption

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

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



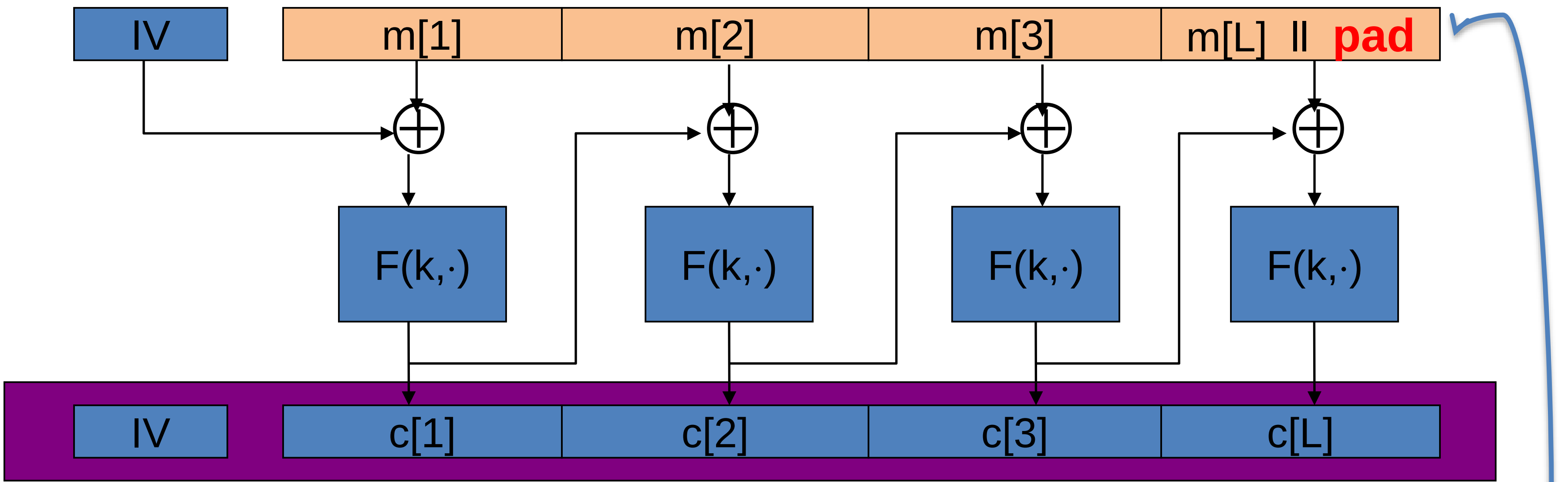
note: parallelizable (unlike CBC)

$$c_i = F_k(IV + i) \oplus m_i$$

Comparison: CTR vs. CBC

	CBC	CTR mode
Uses	PRP	PRF
Parallel processing	No	Yes
Security	q^2 L^2 << 2^n	q^2 L << 2^n
Dummy padding block	Yes	No

A CBC technicality: padding



removed during decryption

TLS: for n>0, n byte pad is | n | n | n | ••• | n |

if no pad needed, add a dummy block

TLS bugs in older versions

IV for CBC is predictable: (chained IV)

- IV for next record is last ciphertext block of current record.
- Not CPA secure.

Padding oracle: during decryption

- If pad is invalid send decryption failed alert
- If mac is invalid send bad_record_mac alert

⇒ attacker learns information about plaintext

Lesson: when decryption fails, do not explain why

Recap

- To encrypt longer messages, use *CBC or CTR mode*
  - CPA security
- CTR mode has some advantages
  - *Parallelizable*
  - *Better security*
- CBC encryption with padding is *vulnerable to padding oracle attack*
- Authenticated encryption schemes are CCA secure

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/