02. Symmetric Encryption

Hyung Tae Lee

Spring 2025

Symmetric Encryption

- Assume that a sender and a receiver have the same key
  ⇒ Symmetric/Private key

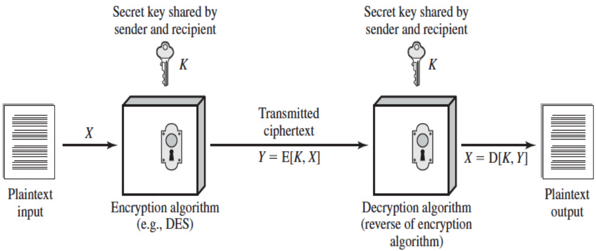
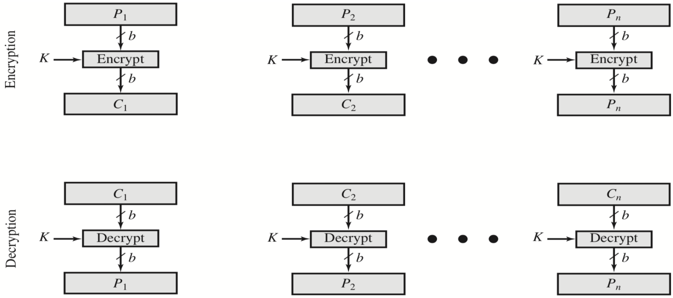


Figure 2.1 **Simplified Model of Symmetric Encryption**

- Classical encryption, block cipher (DES, AES, ARIA, SEED), stream cipher (RC4, ChaCha)
- Pros: Faster than asymmetric (public key) encryption
- Cons: Key share problem, large number of keys

Picture from [SB15]

Block Cipher

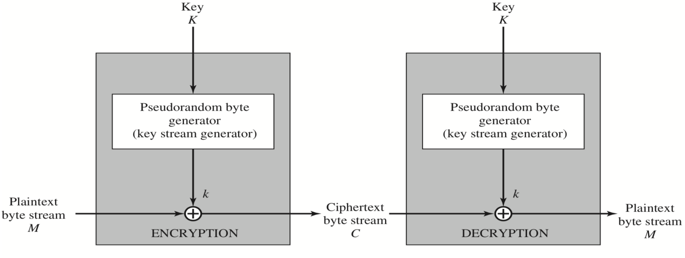


(a) Block cipher encryption (electronic codebook mode)

- Input: One block of elements at a time
- Output: A block for each input block
- Can reuse keys
- More common
- e.g., Substitution cipher, Permutation cipher, DES, AES, ARIA, SEED

Picture from [SB15]

Stream Cipher



(b) Stream encryption

- Input: Elements continuously
- Output: One element at a time
- Faster than block cipher
- Use a key only once
  - Insecure against known plaintext attack: $M \oplus C = M \oplus (M \oplus k) = k$
- e.g., LFSR cipher, RC4, ChaCha

Picture from [SB15]

Block Cipher

Design of Block Cipher

- A symmetric block cipher consists of a sequence of **rounds** which are composed of **substitutions** and **permutations** controlled by a **key**
- Parameters and design features for a symmetric block cipher
  - Block size
  - Key size
  - Number of rounds
  - Subkey generation algorithm
  - Round function
  - Fast software/hardware encryption/decryption
  - Ease of (security) analysis

Computationally Secure

- An encryption scheme is computationally secure if the ciphertext generated by the scheme meets one or both the following criteria:
  - The cost of breaking the cipher exceeds the value of the encrypted information
  - The time required to break the cipher exceeds the useful lifetime of the information
- Example: Cryptographic key sizes (`https://www.keylength.com`)

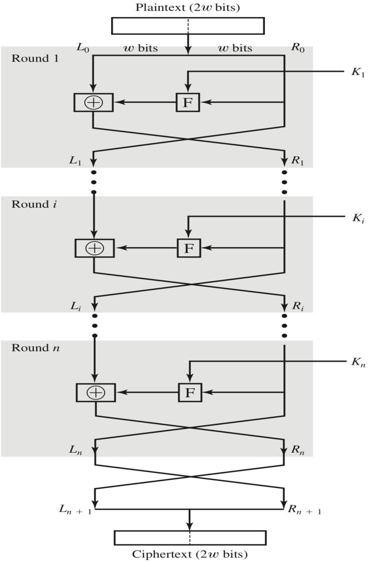
Table: Keys Length Recommendations by NIST (2020)

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Size	Elliptic Curve	Hash (A)	Hash (B)
(Legacy)	80	2TDEA	1024	160	1024	160	SHA-1	
2019-2030	112	3TDEA AES-128	2048	224	2048	224	SHA-224 SHA-512/224 SHA3-224	
2019-2030 & beyond	128	AES-128	3072	256	3072	256	SHA256 SHA-512/256 SHA3-256	SHA-1 KMAC128
2019-2030 & beyond	192	AES-192	7680	384	7680	384	SHA-384 SHA3-384	SHA-224 SHA-512/224
2019-2030 & beyond	256	AES-256	15360	512	15360	512	SHA-512 SHA3-512	SHA-256, 384, 512 SHA-512/256 SHA3-512

Data Encryption Standard (DES)

Feistel Cipher Structure

- First designed by Horst Feistel in 1973
- Input: A plaintext block of length $2w$ bits and a key K
- Divide an input plaintext into two halves, L_0 and R_0
- They pass n rounds and then are combined to produce the ciphertext block
- All rounds have the same structure

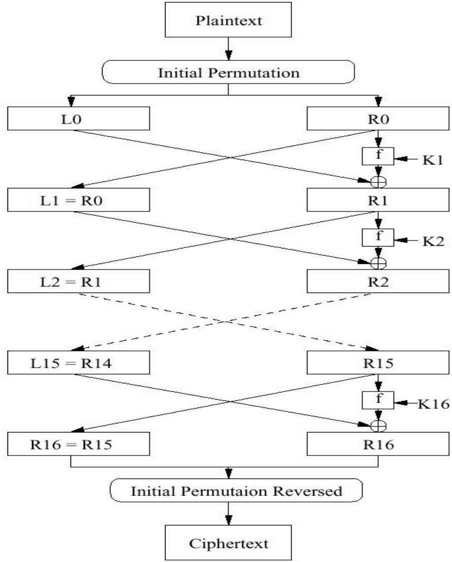


Picture from [SB15]

Data Encryption Standard (DES)

- Adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46)
- Plaintext: 64 bits in length
- Key: 56 bits in length
- 16 rounds which were designed based on a variant of Feistel cipher structure
- 16 subkeys are generated from a key, one of which is used for each round.
- Key schedule for encryption/decryption (K_i: i-th subkey)
  - Encryption: $K_1 \to K_2 \to \cdots \to K_{16}$
    $\Rightarrow$ Decryption: $K_{16} \to K_{15} \to \cdots \to K_1$

DES Overview



Picture from http://www.sm.luth.se/csee/courses/smd/102/lek3/lek3.html

DES: Initial Permutation

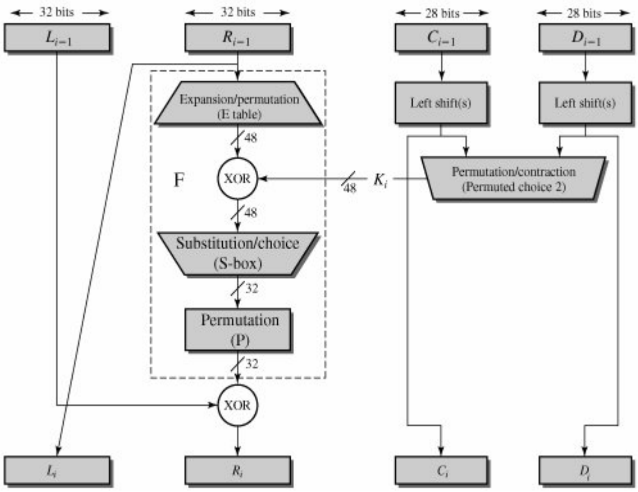
Table: Initial Permutation

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Table: Inverse of Initial Permutation

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

DES: Single Round



Picture from [Sta05]

DES: Permutation Tables E, P

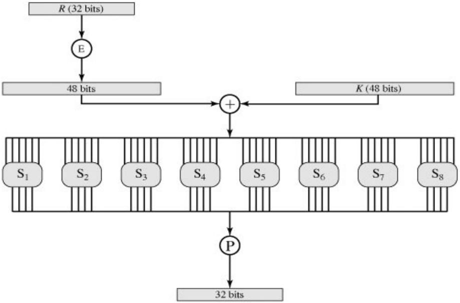
Table: Expansion Permutation (E)

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Table: Permutation Function (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

DES: S Boxes



- S-box $S_i : \{0,1\}^6 \to \{0,1\}^4$ for $1 \leq i \leq 8$
- $S_i(b_0 b_1 b_2 b_3 b_4 b_5)$: $(b_0 b_5)$-row $(b_1 b_2 b_3 b_4)$-column in S_i table where $b_i \in \{0,1\}$

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Picture from [Sta05]

DES: Key Schedule I

- Permuted choice 1: Key $K \mapsto (C_0, D_0)$ where C_0 and D_0 are 28-bit.

Table: Permuted choice 1 (PC-1)

Left							Right						
57	49	41	33	25	17	9	63	55	47	39	31	23	15
1	58	50	42	34	26	18	7	62	54	46	38	30	22
10	2	59	51	43	35	27	14	6	61	53	45	37	29
19	11	3	60	52	44	36	21	13	5	28	20	12	4

DES: Key Schedule II

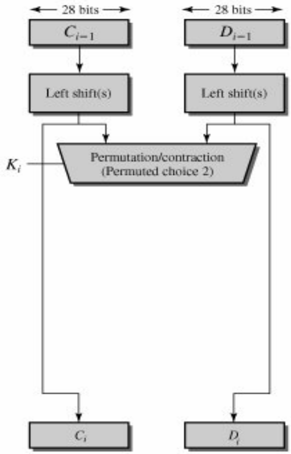


Table: Permuted choice 2 (PC-2)

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Table: Rotations in the key-schedule

Rounds	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Num of left shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Picture from [Sta05]

DES: Security Analysis

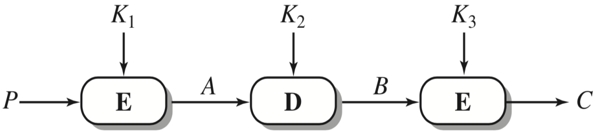
- All parts are linear, except S-box
- Brute-force attack: 2^{56} candidates for a key
  - At CRYPTO '93 Rump Session, Michael Wiener gave a very detailed design of a DES key search machine: Expected record - 1.5 days using 1993 technology for $100,000
  - In 1998, Electronic Frontier Foundation built a key search machine costing $250,000 (Record: 56 hours in July 1998)
  - In 1999, RSA Laboratory found a DES key in 22 hours and 15 minutes.
- Linear cryptanalysis (LC) proposed by Mitsuru Matsui
  - Generate 2^{43} pairs: 40 days
  - Find a key: 10 days

Enhancing Security: Triple DES

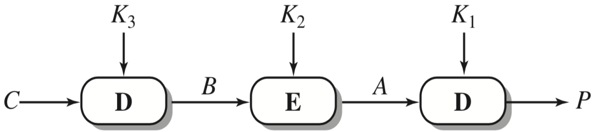
Triple DES

- $\mathsf{Enc}(K = (K_1, K_2, K_3), P) = \mathsf{E}(K_3, \mathsf{D}(K_2, \mathsf{E}(K_1, P)))$
- $\mathsf{Dec}(K = (K_1, K_2, K_3), C) = \mathsf{D}(K_1, \mathsf{E}(K_2, \mathsf{D}(K_3, C)))$

where P: plaintext, C: ciphertext, E: encryption algorithm, and D: decryption algorithm



(a) Encryption



(b) Decryption

Picture from [SB15]

Properties of Triple DES

- Standardized for use in financial applications in ANSI standard X9.17 in 1985
- Incorporated as part of DES in 1999, with the publication of FIPS PUB 46-3
- The encryption algorithm follows an encrypt-decrypt-encrypt (EDE) sequence
- Plaintext size = Ciphertext size: 64 bits in length
- Key size: $3 \times$DES keys = 168 bits in length
  - FIPS 46-3 also allows for the use of two keys, with $K_1 = K_3 \Rightarrow$ 112 bits
- Attractions
  - Secure against brute-force attacks of DES
  - Underlying cryptographic algorithm is DES
- Weak points
  - 64-bit block size (vs 168-bit key size)
  - Algorithm is slow in software

Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES)

- Also known as Rijndal
- Developed by Vincent Rijmen and Joan Daemen
- Selected by the five-year NIST standardization process in 2001
- Specification

Block size (bits)	128		
Key size (bits)	128	192	256
Number of rounds	10	12	14
Underlying structure	Substitution-Permutation Network		

Preliminaries for AES

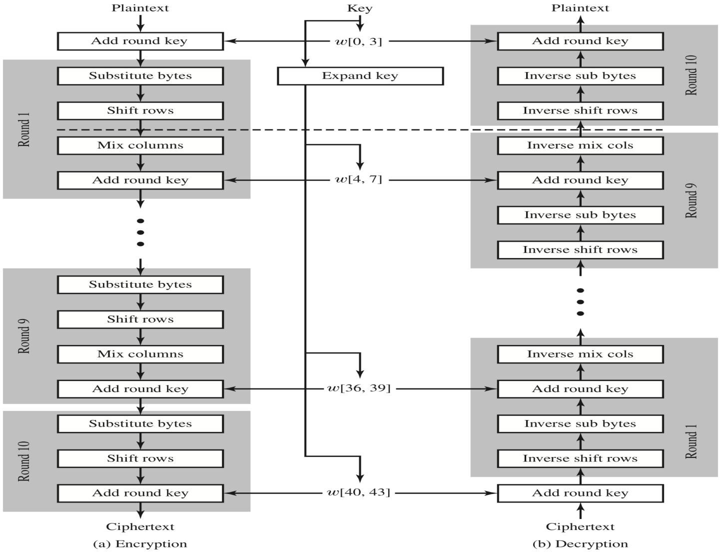
- Hexadecimal notation

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

- State: 128 bits $\rightarrow$ (4×4) matrix of bytes

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

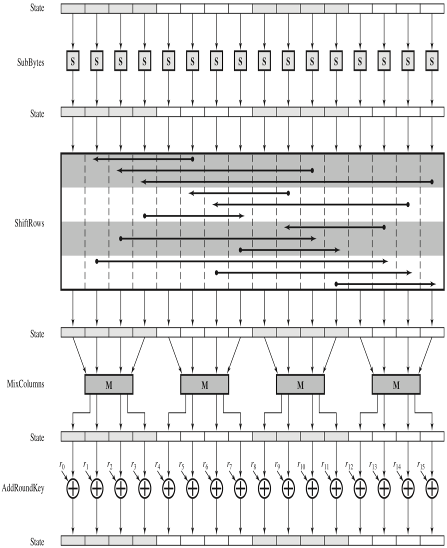
AES: Overview of the Encryption/Decryption Algorithms



Picture from [SB15]

AES: Single Round

- Substitution bytes: Byte-by-byte substitution of the block using S-box (all rounds)
- Shift rows: A simple permutation that is performed row by row (all rounds)
- Mix columns: A substitution that alters each byte in a column as a function of all of the bytes in the column (1-9 rounds)
- Add round keys: A simple bitwie XOR of the current block with a portion of the explanded key (all rounds)



Picture from [SB15]

Mathematical Background: Finite Fields

- Loosely speaking, a field $\mathbb{F}$ is a set that has the operations of addition, multiplication, subtraction, and division by nonzero elements. It is also required that the associative, commutative, and distributive laws hold: For $a, b, c \in \mathbb{F}$,
  - (associative) $(a+b)+c = a+(b+c)$, $a \times (b \times c) = (a \times b) \times c$
  - (commutative) $a+b = b+a$, $a \times b = b \times a$
  - (distributive) $a \times (b+c) = a \times b + a \times c$
- The set of real numbers, the set of complex numbers are fields. But, the set of integers is not a field. (e.g., $1/3$ is not an integer.)
- $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ is a field if p is prime. It is not a field if p is composite.
- $GF(p^n)$: a field with p^n elements ($\mathbb{Z}_p[X] \pmod{P(X)}$ where $P(X)$ is an irreducible polynomial mod p of degree n)

$GF(2^8)$ for AES

- $P(X) = X^8 + X^4 + X^3 + X + 1$ is irreducible in $\mathbb{Z}_2[X]$.
- $GF(2^8)$: the set consisting of

$$b_7X^7 + b_6X^6 + b_5X^5 + b_4X^4 + b_3X^3 + b_2X^2 + b_1X^1 + b_0$$

  where $b_i \in \{0, 1\}$ for $0 \leq i \leq 7$
- $B(X) = b_7X^7 + b_6X^6 + b_5X^5 + b_4X^4 + b_3X^3 + b_2X^2 + b_1X^1 + b_0$ is corresponded to a 8-bit vector $b_7b_6b_5b_4b_3b_2b_1b_0$.
- Addition: $A(X) + B(X) \bmod P(X)$ for $A(X), B(X) \in GF(2^8)$
  - $A(X) = X^7 + X^6 + X^3 + X + 1$
  - $B(X) = X^4 + X^3 + 1$
  - $A(X) + B(X) = (X^7 + X^6 + X^3 + X + 1) + (X^4 + X^3 + 1) = X^7 + X^6 + X^4 + X$
  - Bitwise XOR between corresponded vectors

$$11001011 \oplus 00011001 = 11010010$$

$GF(2^8)$ for AES (Cont.)

- Multiplication: $A(X) \cdot B(X) \pmod{P(X)}$
  - $A(X) = X^7 + X^6 + X^3 + X + 1$
  - $B(X) = X^4 + X^3 + 1$

$$
\begin{aligned}
& A(X) \cdot B(X) \pmod{P(X)} \\
= \quad & (X^7 + X^6 + X^3 + X + 1)(X^4 + X^3 + 1) \\
= \quad & (X^7 + X^6 + X^3 + X + 1)X^4 + (X^7 + X^6 + X^3 + X + 1)X^3 \\
& \qquad + (X^7 + X^6 + X^3 + X + 1) \\
= \quad & (X^{11} + X^{10} + X^7 + X^5 + X^4) + (X^{10} + X^9 + X^6 + X^4 + X^3) \\
& \qquad + (X^7 + X^6 + X^3 + X + 1) \\
= \quad & X^{11} + X^9 + X^5 + X + 1 \pmod{X^8 + X^4 + X^3 + X + 1} \\
= \quad & X^7 + X^6 + X^3 + X^2 + 1
\end{aligned}
$$

$GF(2^8)$ for AES (Cont.)

- Inverse: Extended Euclidean Algorithm
  - $A(X) = X^7 + X^6 + X^3 + X + 1 \Rightarrow A(X)^{-1} \bmod P(X)$?
  - $\gcd(X^7 + X^6 + X^3 + X + 1, X^8 + X^4 + X^3 + X + 1) = 1$

$$\begin{aligned} X^8 + X^4 + X^3 + X + 1 &= (X+1)(X^7 + X^6 + X^3 + X + 1) + (X^6 + X^2 + X) \\ X^7 + X^6 + X^3 + X + 1 &= (X+1)(X^6 + X^2 + X) + \underbrace{1}_{\text{GCD}} \end{aligned}$$

Then,

$$\begin{aligned} 1 &= (X^7 + X^6 + X^3 + X + 1) + (X+1)(X^6 + X^2 + X) \\ &= (X^7 + X^6 + X^3 + X + 1) \\ &\quad + (X+1)((X+1)(X^7 + X^6 + X^3 + X + 1) + (X^8 + X^4 + X^3 + X + 1)) \\ &= (1 + (X+1)^2)(X^7 + X^6 + X^3 + X + 1) + (X^8 + X^4 + X^3 + X + 1)(X+1) \\ &= (X^2)(X^7 + X^6 + X^3 + X + 1) + (X^8 + X^4 + X^3 + X + 1)(X+1) \end{aligned}$$

$$\therefore (X^7 + X^6 + X^3 + X + 1)^{-1} = X^2 \pmod{X^8 + X^4 + X^3 + X + 1}$$

Euclidean Algorithm I

Theorem

Let $\mathbb{F}$ be a field and $A(x), P(x) \in \mathbb{F}[x]$. Then, there exist polynomials $S(X)$, $T(X)$ $\in \mathbb{F}[X]$ such that

$$P(X)S(X) + A(X)T(X) = \gcd(P(X), A(X)).$$

(In fact, it holds if $\mathbb{F}$ is an Euclidean domain, e.g., $\mathbb{F} = \mathbb{Z}$ (the set of integers))

- $P(X)$ is irreducible in $\mathbb{F}[X]$

  $\Rightarrow \gcd(P(X), A(X)) = 1$ if $A(X)$ is not a multiple of $P(X)$

  $\Rightarrow \; P(X)S(X) + A(X)T(X) = \gcd(P(X), A(X)) = 1$

  $\Rightarrow \; A(X)T(X) \equiv 1 \pmod{P(X)}$

  $\therefore \; T(X) \equiv A(X)^{-1} \pmod{P(X)}$

Euclidean Algorithm II

- Assume that $\deg(P(X)) \geq \deg(A(X))$.

$$
\begin{array}{rcl}
\underbrace{P(X)}_{:=R_0(X)} & = & Q_0(X) \cdot \underbrace{A(X)}_{:=R_1(X)} + R_2(X) \\
R_1(X) & = & Q_1(X) \cdot R_2(X) + R_3(X) \\
R_2(X) & = & Q_2(X) \cdot R_3(X) + R_4(X) \\
 & \vdots & \\
R_{n-2}(X) & = & Q_{n-2}(X) \cdot R_{n-1}(X) + R_n(X) \\
R_{n-1}(X) & = & Q_{n-1}(X) \cdot R_n(X)
\end{array}
$$

$\Rightarrow \; R_n(X) = \gcd(P(X), A(X))$

Example of Euclidean Algorithm

- $A(X) = X^7 + X^6 + X^3 + X + 1$
- $P(X) = X^8 + X^4 + X^3 + X + 1$
- $\gcd(P(X), A(X)) = 1$

$$
\begin{aligned}
\underbrace{X^8 + X^4 + X^3 + X + 1}_{R_0(X):=P(X)} &= \underbrace{(X+1)}_{Q_0(X)}\underbrace{(X^7 + X^6 + X^3 + X + 1)}_{R_1(X):=A(X)} + \underbrace{(X^6 + X^2 + X)}_{R_2(X)} \\
\underbrace{X^7 + X^6 + X^3 + X + 1}_{R_1(X)} &= \underbrace{(X+1)}_{Q_1(X)}\underbrace{(X^6 + X^2 + X)}_{R_2(X)} + \underbrace{1}_{R_3(X)} \\
\underbrace{(X^6 + X^2 + X)}_{R_2(X)} &= \underbrace{(X^6 + X^2 + X)}_{Q_2(X)} \cdot \underbrace{1}_{R_3(X)} + 0
\end{aligned}
$$

Extended Euclidean Algorithm

Theorem

- $S_0(X) = 1, S_1(X) = 0$
- $T_0(X) = 0$, $T_1(X) = 1$
- $S_{i+1} = S_{i-1} - S_i Q_i$, $T_{i+1} = T_{i-1} - T_i Q_i$

Then, $P(X)S_i(X) + A(X)T_i(X) = R_i(X)$.

Proof. (Use the mathematical induction on i)

1. $i = 0$; $P(X)S_0(X) + A(X)T_0(X) = P(X) = R_0(X)$
2. $i = 1$; $P(X)S_1(X) + A(X)T_1(X) = A(X) = R_1(X)$
3. $$\begin{aligned} PS_i + AT_i &= P(S_{i-2} - S_{i-1}Q_{i-1}) + A(T_{i-2} - T_{i-1}Q_{i-1}) \\ &= \underbrace{(PS_{i-2} + AT_{i-2})}_{R_{i-2}} - \underbrace{(PS_{i-1} + AT_{i-1})}_{R_{i-1}} Q_{i-1} \\ &= R_{i-2} - R_{i-1}Q_{i-1} = R_i \end{aligned}$$

Example of Extended Euclidean Algorithm I

- $S_0(X) = 1, S_1(X) = 0$
- $T_0(X) = 0,\ T_1(X) = 1$
- $S_{i+1} = S_{i-1} - S_i Q_i,\ T_{i+1} = T_{i-1} - T_i Q_i$

1. $S_2 = S_0 - S_1 Q_1 = 1,\ T_2 = T_0 - T_1 Q_1 = -(X+1)$
   $\Rightarrow\ P(X) - (X+1)A(X) = R_2(X)$
   $\Rightarrow\ P(X) = (X+1)A(X) + R_2(X)$
2. $S_3 = S_1 - S_2 Q_2 = -(X+1),\ T_3 = T_1 - T_2 Q_2 = 1 + (X+1)^2 = X^2$
   $\Rightarrow\ P(X)(X+1) + X^2 A(X) = 1$

Example of Extended Euclidean Algorithm II

$$
\begin{aligned}
1 &= (X^7 + X^6 + X^3 + X + 1) + (X + 1)(X^6 + X^2 + X) \\
&= (X^7 + X^6 + X^3 + X + 1) \\
&\quad + (X + 1)((X + 1)(X^7 + X^6 + X^3 + X + 1) + (X^8 + X^4 + X^3 + X + 1)) \\
&= (1 + (X + 1)^2)\underbrace{(X^7 + X^6 + X^3 + X + 1)}_{A(X)} + \underbrace{(X^8 + X^4 + X^3 + X + 1)}_{P(X)}(X + 1) \\
&= (X^2)(X^7 + X^6 + X^3 + X + 1) + (X^8 + X^4 + X^3 + X + 1)(X + 1)
\end{aligned}
$$

$$
\therefore (X^7 + X^6 + X^3 + X + 1)^{-1} = X^2 \pmod{X^8 + X^4 + X^3 + X + 1}
$$

AES: Substitue Bytes Transformation

- S(**xy**) = (**x**, **y**)-component in S-box where **x**, **y** are hexadecimal digits.

Table: S-box for AES

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
x	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

AES: Inverse S-Box

Table: Inverse S-box for AES

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	FA
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

AES: Example of S-Box Evaluation

- Example

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

$\longrightarrow$

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

AES: Design Principle of S-Box

- Regard a two-byte element as an element in $\mathbb{Z}_2[X]/\langle X^8 + X^4 + X^3 + X + 1 \rangle$
- S-Box: Given $x = (x_7x_6x_5x_4x_3x_2x_1x_0)$ with $x_i \in \{0, 1\}$ for $0 \leq i \leq 7$,
  1. Compute $y_7y_6y_5y_4y_3y_2y_1y_0 \longleftrightarrow x^{-1}$ (cf. $00000000 \longleftrightarrow 00000000$)
  2. Compute

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{pmatrix}$$

  3. Output $z_7z_6z_5z_4z_3z_2z_1z_0$

AES: S-Box Computation in $GF(2^8)$

- Bytes: 00 $\longrightarrow$ x=00000000 $\xrightarrow{x^{-1}}$ y = 00000000 $\longrightarrow$ z =01100011 $\longrightarrow$ 63
- Bytes: 01 $\longrightarrow$ x=00000001 $\xrightarrow{x^{-1}}$ y = 00000001 $\longrightarrow$ z = 01111100 $\longrightarrow$ 7C
- Bytes: CB $\longrightarrow$ x=11001011 $\xrightarrow{x^{-1}}$ y = 00000100 $\longrightarrow$ z = 00011111 $\longrightarrow$ 1F

AES: Shift Row Transformation

- Shift row transformation
  - 1st row: No change
  - 2nd row: 1-byte circular left shift
  - 3rd row: 2-byte circular left shift
  - 4th row: 3-byte circular left shift
- Example

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

$\longrightarrow$

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

- Inverse of the shift row transformation: left $\Rightarrow$ right

AES: Mix Column Transformation

- Multiply the current state by a matrix in $GF(2^8)$

$$\underbrace{\begin{pmatrix} 00000010 & 00000011 & 00000001 & 00000001 \\ 00000001 & 00000010 & 00000011 & 00000001 \\ 00000001 & 00000001 & 00000010 & 00000011 \\ 00000011 & 00000001 & 00000001 & 00000010 \end{pmatrix}}_{=:\mathbf{M}} \begin{pmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{pmatrix}$$

$$= \begin{pmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{pmatrix}$$

- **M** is invertible
- Inverse of the mix column transformation: Multiply by the inverse of **M**

AES: Example of the Mix Column Transformation

- Example

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

$\longrightarrow$

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$$\underbrace{\begin{pmatrix} X & X+1 & 1 & 1 \\ 1 & X & X+1 & 1 \\ 1 & 1 & X & X+1 \\ X+1 & 1 & 1 & X \end{pmatrix}}_{\mathbf{M} \text{ in } GF(2^8)} \begin{pmatrix} 87 & (= X^7 + X^2 + X + 1) \\ 6E & (= X^6 + X^5 + X^3 + X^2 + X) \\ 46 & (= X^6 + X^2 + X) \\ A6 & (= X^7 + X^5 + X^2 + X) \end{pmatrix}$$

$$\begin{aligned} &= X(X^7 + X^2 + X + 1) + (X+1)(X^6 + X^5 + X^3 + X^2 + X) \\ &\qquad +1(X^6 + X^2 + X) + 1(X^7 + X^5 + X^2 + X) \\ &= X^8 + X^6 + X^4 + X^3 + X^2 \\ &= X^6 + X^2 + X + 1 \pmod{X^8 + X^4 + X^3 + X + 1} \\ &= 0100\ 0111 = 47 \end{aligned}$$

AES: Add Round Key Transformation

- Add round key transformation: Bitwise XOR of the 128 bits of the current state with the 128 bits of the round key.
- Example

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

Current state

$\oplus$

AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6E

Round key

$=$

EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D2

Output of the round

$$
\begin{array}{ccccccccccc}
 & 4 & 7 & = & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
\oplus & A & C & = & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
\hline
 & E & B & = & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1
\end{array}
$$

- Inverse of the add round key transformation:

$$
(\text{Output of the round}) \oplus (\text{Round key}) = (\text{Current state})
$$

AES: Key Expansion

- Key $K = (W(0), W(1), W(2), W(3))$

$w_{0,0}$	$w_{0,1}$	$w_{0,2}$	$w_{0,3}$
$w_{1,0}$	$w_{1,1}$	$w_{1,2}$	$w_{1,3}$
$w_{2,0}$	$w_{2,1}$	$w_{2,2}$	$w_{2,3}$
$w_{3,0}$	$w_{3,1}$	$w_{3,2}$	$w_{3,3}$
$W(0)$	$W(1)$	$W(2)$	$W(3)$

- $W(i) = \begin{cases} W(i-4) \oplus W(i-1) & \text{if } i \text{ is not a multiple of } 4 \\ W(i-4) \oplus T(W(i-1)) & \text{if } i \text{ is a multiple of } 4 \end{cases}$ where T is a transformation performed as follows.

  Let $W(i-1) = (w_{0,i-1}, w_{1,i-1}, w_{2,i-1}, w_{3,i-1})$. Then,

  1. Shift one byte circular left: $(w_{1,i-1}, w_{2,i-1}, w_{3,i-1}, w_{0,i-1})$
  2. Evaluate the S-box: $(S(w_{1,i-1}), S(w_{2,i-1}), S(w_{3,i-1}), S(w_{0,i-1}))$
  3. Compute $r(i) = X^{(i-4)/4}$ in $GF(2^8)$
  4. $T(W(i-1)) = (S(w_{1,i-1}) \oplus r(i), S(w_{2,i-1}), S(w_{3,i-1}), S(w_{0,i-1}))$

AES: Security

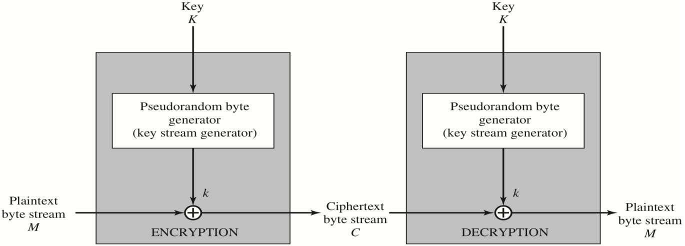
- Only Add Round Key Transformation uses a key
- Brute-force attacks: 2^{128} candidates for a key of AES-128
- The best known attack: Biclique attacks by A. Bogdanov, D. Khovratovich, and C. Rechberger (at ASIACRYPT 2011)
  - $2^{126.1}$ for AES-128
  - $2^{189.7}$ for AES-192
  - $2^{254.4}$ for AES-256

Comparison of Representative Block Ciphers

	DES	Triple DES	AES		
Block size (bit)	64	64	128		
Key size (bit)	56	112 or 168	128	192	256
Number of rounds	16	3×16	10	12	14
Underlying Structure	Feistel		SPN		

Stream Cipher

Stream Cipher



(b) Stream encryption

- Input: Elements continuously
- Output: One element at a time
- Faster than block cipher
- Use a key only once
  - Insecure against known plaintext attack: $M \oplus C = M \oplus (M \oplus k) = k$
- e.g., LFSR cipher, RC4, ChaCha

Picture from [SB15]

Random Numbers

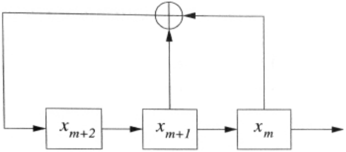
- Wide range of uses in cryptography, e.g.,
  - private keys in public-key encryption
  - keys for stream cipher
  - symmetric key for use as a temporary session key
- Requirements
  - Randomness
    - Uniform distribution: Frequency of occurrence of each number should be identical
    - Independence: No one value can be inferred from the others
  - Unpredictability
    - Each number is statistically independent of other numbers
    - Anyone should not be able to predict future elements of the sequence on the basis of earlier elements

Random vs. Pseudorandom

- True random number generator (TRNG)
  - Expensive to be realized
  - Use a nondeterministic source to produce randomness
  - Operate by measuring unpredictable natural processes such as radiation, gas discharge and leaky capacitors
  - Increasingly provided on recent processors
- Pseudorandom numbers
  - Sequences produced that satisfy statistical randomness tests
  - Use a deterministic source $\Rightarrow$ likely to be predictable

Recall: LFSR Cipher

- A shift register whose input bit is a linear function of its previous state
- Example: a shift register satisfying
  $\underbrace{x_{m+3} = x_{m+1} + x_m}_{\text{linear relation}}$



LFSR Cipher

For a linear function $f(z_1, \ldots z_\ell) = \sum_{i=1}^{\ell} c_i z_i$ with constant c_i's and a key $K = (k_1, \ldots, k_\ell) \in (\mathbb{Z}_2)^\ell$

- $\mathrm{Enc}(K, (x_1, \ldots, x_m)) = (x_1 \oplus k_1, \ldots, x_m \oplus k_m)$
- $\mathrm{Dec}(K, (y_1, \ldots, y_m)) = (y_1 \oplus k_1, \ldots, y_m \oplus k_m)$

where $k_j = f(k_{j-\ell}, k_{j-\ell+1}, \ldots k_{j-1})$ for $\ell + 1 \leq j \leq m$

- Insecure against known plaintext attacks

Picture and example from [TW06]

RC4: Overview

- Designed by Ron Rivest in 1987, but leaked in 1994
- RC = Rivest Cipher (cf. RC2, RC5, RC6: Block cipher)
- Variable-key-size stream cipher with byte-oriented operations
  - Expected to be fast in software

Cipher	Key length	Speed (Mbps)
DES	56	21
3DES	168	10
AES	128	61
RC4	Variable	113

- Based on the use of a random permutation

RC4: Initialization of S

- **S**: Set equal to the values from 0 through 255 in ascending order
  $\Rightarrow$ **S**[0]=0, **S**[1]=1, ..., **S**[255]=255
- **T**: A temporary vector where the first *keylen* elements of **T** are copied from the key K and then K is repeated as many times as necessary to fill out **T** (*keylen* = the byte-size of K)

```
/* Initialization */
for i=0 to 255 do
    S[i] = i;
    T[i] = K[i mod keylen];
```

- Permute **S** using **T**

```
/* Initial Permutation of S */
j=0;
for i=0 to 255 do
    j = (j+ S[i]+T[i]) mod 256;
    Swap(S[i], S[j]);
```

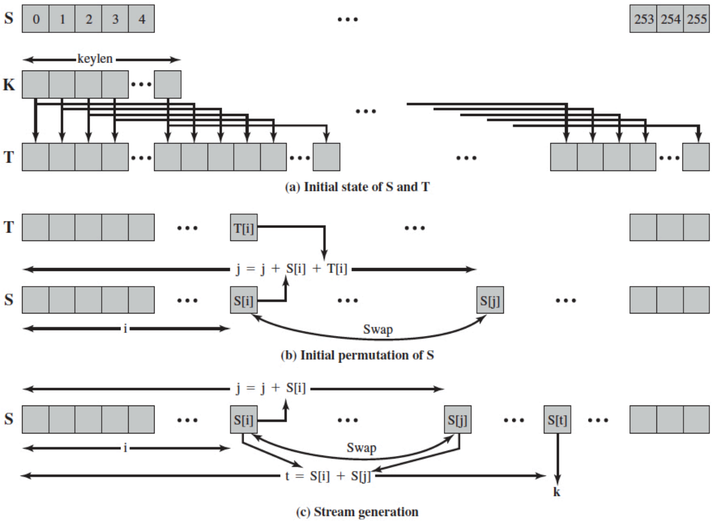
RC4: Stream Generation

- Cycling through all the elements of **S**[i]
- For each **S**[i], swapping **S**[i] with **S**[j] where j is determined by the scheme description
- After **S**[255] is reached, the process continues starting over again at **S**[0]

```
/*Stream Generation*/
i, j =0;
while (true)
    i = (i+1) mod 256;
    j = (j+S[i]) mod 256;
    Swap(S[i], S[j]);
    t = (S[i]+S[j]) mod 256;
    k = S[t];
```

- Encryption: XOR the value k with the next byte of plaintext
- Decryption: XOR the value k with the next byte of ciphertext

RC4: Graphical Explanation



RC4: Security

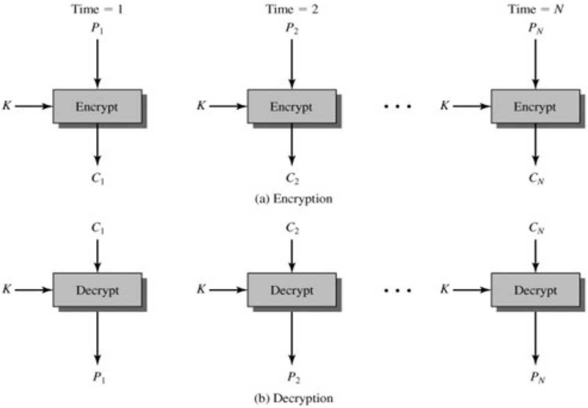
- None of attacks is practical against RC4 itself if a key size is reasonable, e.g., 128 bit
- In 2001, there was reported that the WEP (Wired Equivalent Privacy) protocol with RC4 is vulnerable to a particular attack approach.
- As of 2015, some cryptography agencies may possess the capability to break RC4 when used in the TLS (Transport Layer Security) protocol.
- IETF (Internet Engineering Task Force), Mozilla and Microsoft recommended to prohibit the use of RC4 in TLS protocols.

Modes of Operation

Modes of Operation

- Limitation of block cipher: Fixed length of plaintexts
- Modes of operations
  - To encrypt various-sized plaintexts with block cipher
  - Defined by NIST (Special Publication 800-38A)
    - Electronic Codebook (ECB) Mode
    - Cipher Block Chaining (CBC) Mode
    - Cipher Feedback (CFB) Mode
    - Output Feedback (OFB) Mode
    - Counter (CTR) Mode

Electronic Codebook (ECB) Mode I



Electronic Codebook (ECB) Mode II

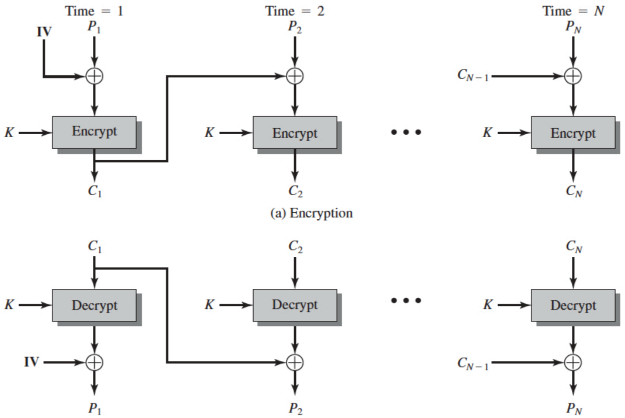
Electronic Codebook (ECB) Mode

- $\mathrm{Enc}(K, P_i) = C_i$
- $\mathrm{Dec}(K, C_i) = P_i$

- Pros
  - No need block synchronization
  - Transmission error affects the corresponding block only
  - Can be parallelized
- Cons
  - Deterministic encryption
  - Insecure against substitution attack: Replace the 4-th block by my account number!

Block #	1	2	3	4	5
	Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount $

Cipher Block Chaining (CBC) Mode I



(a) Encryption

(b) Decryption

Cipher Block Chaining (CBC) Mode II

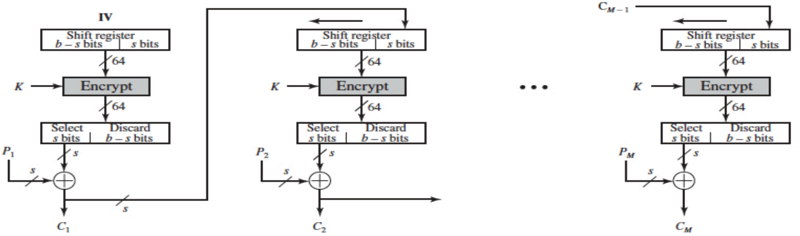
Cipher Block Chaining (CBC) Mode

- $\mathrm{Enc}(K, P_i) = \begin{cases} \mathrm{Enc}_{\mathrm{Block}}(K, IV \oplus P_1) & \text{for the first block} \\ \mathrm{Enc}_{\mathrm{Block}}(K, C_{i-1} \oplus P_i) & \text{for other blocks} \end{cases}$
- $\mathrm{Dec}(K, C_i) = \begin{cases} \mathrm{Dec}_{\mathrm{Block}}(K, C_1) \oplus IV & \text{for the first block} \\ \mathrm{Dec}_{\mathrm{Block}}(K, C_i) \oplus C_{i-1} & \text{for other blocks} \end{cases}$

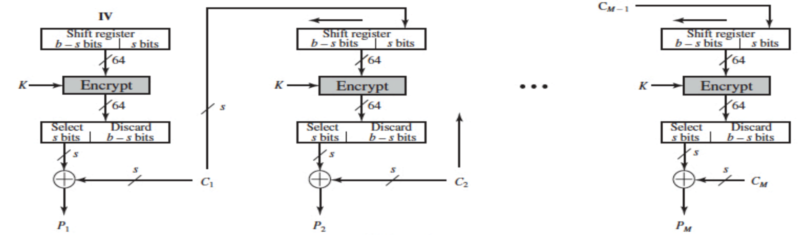
- Chained together
- Randomized by using the initialization vector (IV)
- IV should be non-predictable. If the IV is kept the same for several transfers, the adversary can modify the amount of money being transferred.

Block #	1	2	3	4	5
	Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount $

Cipher Feedback (CFB) Mode I



(a) Encryption



(b) Decryption

Cipher Feedback (CFB) Mode II

Cipher Feedback (CFB) Mode

- $\mathrm{Enc}(K, P_i) = \begin{cases} F_s(\mathrm{Enc}_{\mathrm{Block}}(K, IV)) \oplus P_1 & \text{for the first block} \\ F_s(\mathrm{Enc}_{\mathrm{Block}}(K, C_{i-1})) \oplus P_i & \text{for other blocks} \end{cases}$
- $\mathrm{Dec}(K, C_i) = \begin{cases} F_s(\mathrm{Enc}_{\mathrm{Block}}(K, IV)) \oplus C_i & \text{for the first block} \\ F_s(\mathrm{Enc}_{\mathrm{Block}}(K, C_{i-1})) \oplus C_i & \text{for other blocks} \end{cases}$

where F_s is the function that returns the first s-bit of input.

- Use a block cipher as a building block for a stream cipher
- Randomized by using the initialization vector (IV)

Output Feedback (OFB) Mode I

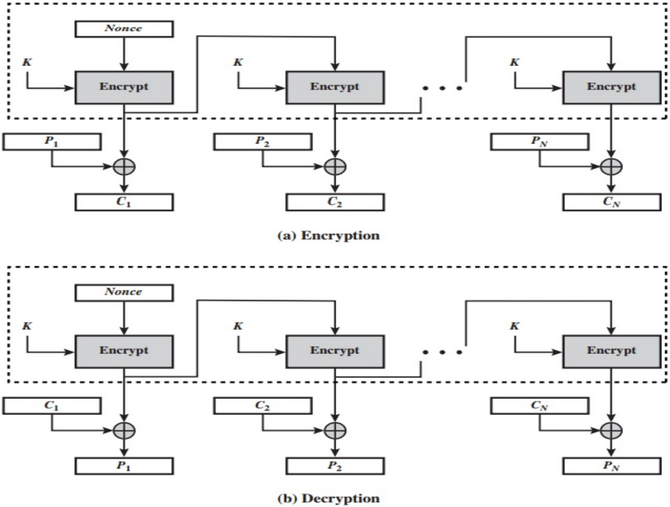


Figure 6.6 Output Feedback (OFB) Mode

Output Feedback (OFB) Mode II

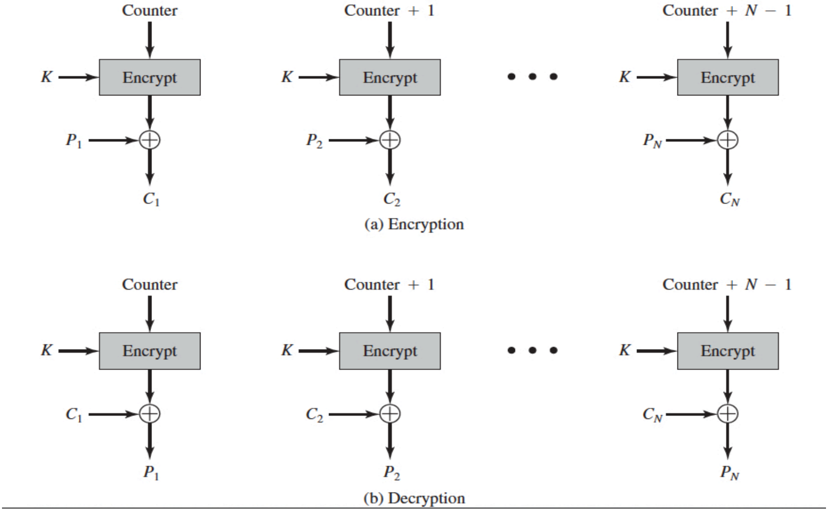
Output Feedback (OFB) Mode

- $\text{Enc}(K, P_i) = X_i \oplus P_i$
- $\text{Dec}(K, C_i) = X_i \oplus C_i$

where $X_i = \begin{cases} \text{Enc}_{\text{Block}}(K, IV) & \text{for the first block} \\ \text{Enc}_{\text{Block}}(K, X_{i-1}) & \text{for other blocks} \end{cases}$

- Use a block cipher as a building block for a stream cipher, like CFB
- Randomized by using the initialization vector (IV)
- The block cipher computations are independent of the plaintext, thus it is possible to precompute X_is

Counter (CTR) Mode I



Counter (CTR) Mode II

Counter (CTR) Mode

- $\text{Enc}(K, P_i) = X_i \oplus P_i$
- $\text{Dec}(K, C_i) = X_i \oplus C_i$

where $X_i = \text{Enc}_{\text{Block}}(K, CTR + i - 1)$

- Use a block cipher as a building block for a stream cipher, like CFB and OFB
- Non-deterministic if CTR is changed.
- Can be parallelized/precomputed

References

PP10 C. Paar and J. Pelzl, Understanding Cryptography, Springer, 2010

SB15 W. Stallings and L. Brown, Computer Security: Principles and Practice, 3rd edition, Pearson Prentice Hall, 2015

Sta05 W. Stallings, Cryptography and Network Security, 4th edition, Pearson Prentice Hall, 2005

TW06 W. Trappe and L. C. Washington, Introduction to Cryptography with Coding Theory, 2nd edition, Pearson Prentice Hall, 2006