Chap. 20
Symmetric Encryption

연세대학교

Cryptography

- 암호화의 세 가지 차원의 분류

1) 평문을 암호문으로 변환하는 데 사용되는 연산 유형
   - 치환(substitution) – 평문의 각 원소가 다른 원소로 매핑됨
   - 전치(transposition) – 평문 원소들이 재배열됨

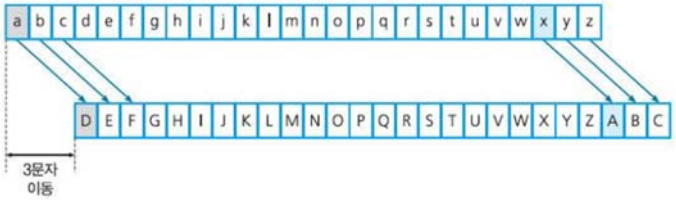
2) 사용된 키 개수
   - 대칭키 – sender와 receiver가 같은 키 사용 (1개)
   - 비대칭키 – sender와 receiver가 서로 다른 키 사용

3) 평문이 처리되는 방식
   - 블록 암호화 – 한 번에 한 블록의 원소들을 처리
   - 스트림 암호화 – 입력 원소들을 연속적으로 처리 (원소단위)

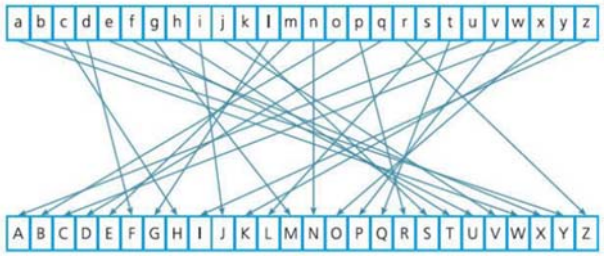
연세대학교

간단한 암호화 - Substitition

- 시저 암호화 – 시프트 치환



- 단일치환 암호화



연세대학교

간단한 암호화 - Transposition

- Transposition – 위치 재배열

  (예) columnar transposition
  - 메시지 WE ARE DISCOVERED. FLEE AT ONCE
  - 키 ZEBRAS → 열순서 6 3 2 4 1 5

6	3	2	4	1	5
W	E	A	R	E	D
I	S	C	O	V	E
R	E	D	F	L	E
E	A	T	O	N	C
E	Q	K	J	E	U

암호문

EVLNE ACDTK ESEAQ ROFOJ DEECU WIREE

연세대학교

Cryptanalysis

- 암호분석가에게 알려진 것
  1) 암호화 알고리즘
  2) 해독될 암호문
  3) 하나 이상의 평문-암호문 쌍
  4) 선택한 평문에 대한 평문-암호문 쌍
  5) 선택한 암호문에 대한 평문-암호문 쌍
- 암호문에 대한 공격 유형

  암호분석가에게 알려진 것
  - ciphertext only : 1, 2
  - known plaintext : 1, 2, 3 * → 표준 헤더의 암호문
  - chosen plaintext : 1, 2, 4 * → 분석용 평문을 선택
  - chosen ciphertext : 1, 2, 5
  - chosen text : 1, 2, 4, 5

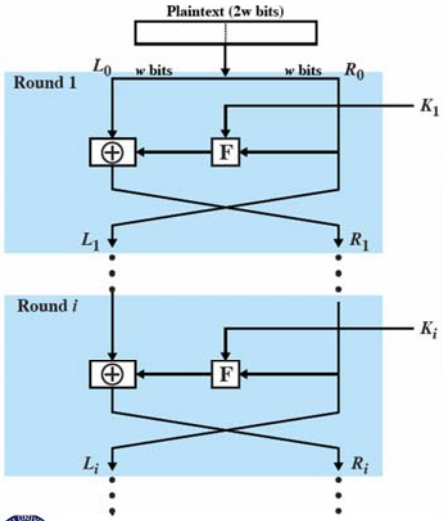
연세대학교

Computationally Secure Algorithms

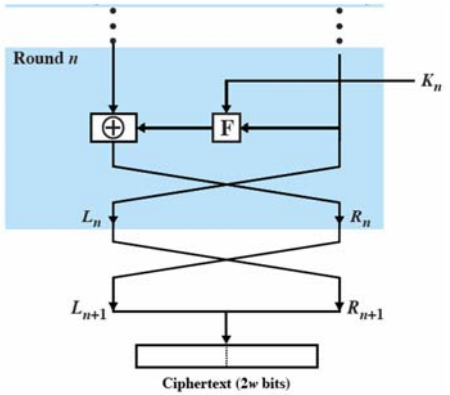
- weak algorithm
  - ciphertext-only 공격을 방어하지 못함
- 암호화 알고리즘은 known-plaintext attack을 막을 수 있게 설계
- 계산적으로 안전한(secure) 알고리즘
  - 암호 해독 비용이 정보 가치를 초과함
  - 암호 해독에 필요한 시간이 정보의 유효 수명을 초과함
- 암호해독에 소요되는 시간
  - 정확하게 측정하기 어려움
  - 알고리즘에 수학적인 약점이 없다면 전수공격 방법에 소요되는 비용과 시간으로 추정함

연세대학교

Feistel Cipher Structure



1973년 IBM Horst Feistel이 설계한 구조



연세대학교

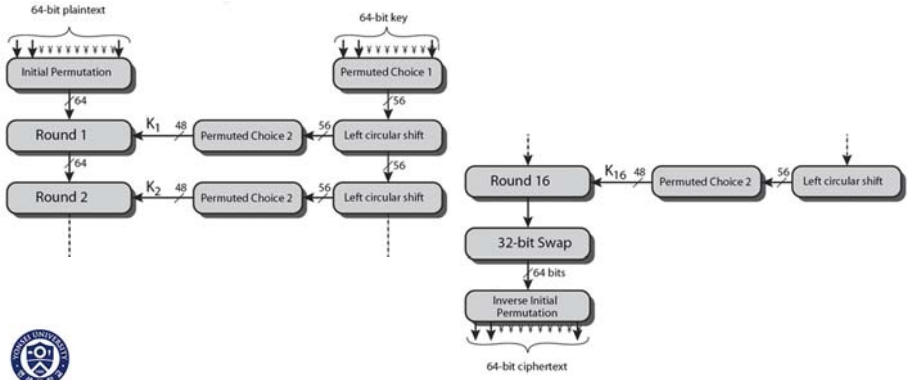
Block Cipher Structure

- 대칭 블록 암호화 구성
  - 여러 라운드의 시퀀스
  - key에 의해 제어되는 substitution과 permutation의 반복
- 암호화 알고리즘의 설계 매개변수
  - 블록 크기
  - 키 크기
  - 라운드 수
  - subkey 생성 알고리즘
  - 라운드 함수(F)
- 암호화 알고리즘의 추가적인 고려사항
  - 빠른 소프트웨어 암호화/복호화
  - 알고리즘 분석의 용이함 – 안정성 있는 알고리즘 개발 가능

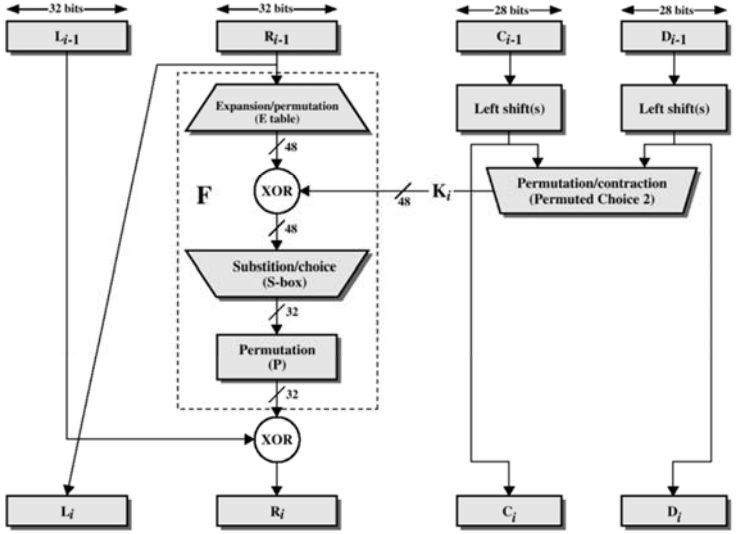
연세대학교

Data Encryption Standard (DES)

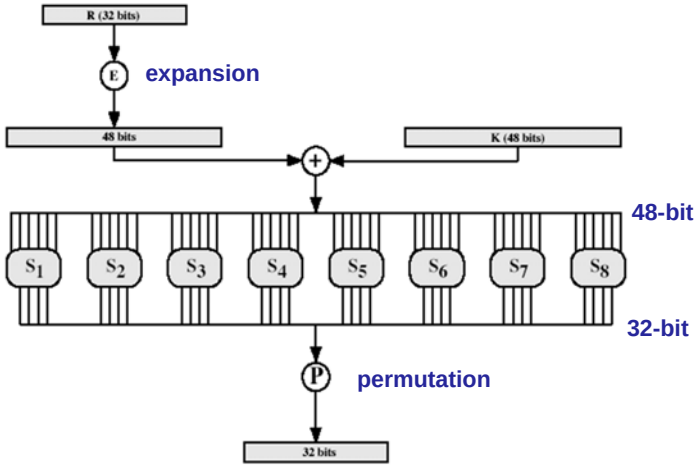
- 가장 널리 사용되는 암호화 방법
- 1977년에 NIST에서 대칭 암호화 표준으로 채택 (FIPS 46)
- 알고리즘은 Data Encryption Algorithm(DEA)라고 부름
- Feistel network의 변형 알고리즘



Single Round of DES Algorithm

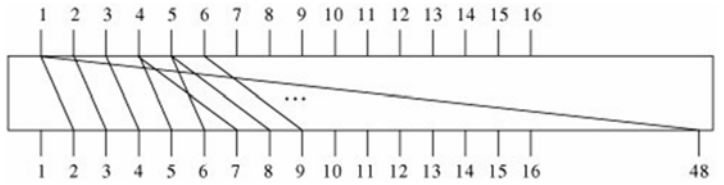


Round function F



Expansion Permutation

Bit	1	2	3	4	5	6	7	8
Moves to Position	2,48	3	4	5,7	6,8	9	10	11,13
Bit	9	10	11	12	13	14	15	16
Moves to Position	12,14	15	16	17,19	18,20	21	22	23,25
Bit	17	18	19	20	21	22	23	24
Moves to Position	24,26	27	28	29,31	30,32	33	34	35,37
Bit	25	26	27	28	29	30	31	32
Moves to Position	36,38	39	40	41,43	42,44	45	46	47,1



Key Transformation

- 64-bit key = 매 8번째 비트를 제거하여 56-bit key로 사용
- 56-bit key를 두 개의 28-bit로 분할 (C_i, D_i)
- shift : 분할된 두 반쪽을 매 round마다 지정된 자리 수만큼 shift left

Cycle Number	Bits Shifted		
1	1	9	1
2	1	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	2
8	2	16	1

- permuted choice : 56-bit에서 48-bit를 선택 (다음 slide)

연세대학교

- Choice Permutation to Select 48 bit Key

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

연세대학교

S-Boxes

- S-box
  - 대치(substitution)은 8개의 S-box를 사용하여 수행
  - 6비트에서 4비트를 선택하는 permuted choice 연산
  - 48비트 입력 = 8개의 6비트 block들로 분할
  - 각 block에 대해서 S-box 연산 수행 ($B_j = b_1b_2b_3b_4b_5b_6$)

		Column $b_2b_3b_4b_5$															
Box	Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1																	
b_1b_6	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2																	
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

B_2 in binary is **0**10011**1** → r = 01(1), c=1001(9) → S = 0 (0000)

연세대학교

P-Boxes, Initial & Final Permutation

- P-Boxes
  - 32 bits 입력의 단순 permutation 수행

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

- Initial and Final Permutations
  - final permutation = inverse initial permutation

pos 4 → pos 16

initial permutation

Bit	Goes to Position							
1-8	40	8	48	16	56	24	64	32
9-16	39	7	47	15	55	23	63	31

final permutation

16 → 4

Bit	Goes to Position							
1-8	58	50	42	34	26	18	10	2
9-16	60	52	44	36	28	20	12	4

연세대학교

Initial & Final Permutation

initial permutation

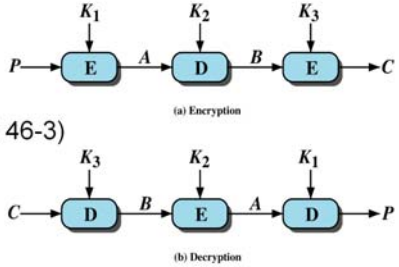
Bit	Goes to Position							
1-8	40	8	48	16	56	24	64	32
916	39	7	47	15	55	23	63	31
1724	38	6	46	14	54	22	62	30
2532	37	5	45	13	53	21	61	29
3340	36	4	44	12	52	20	60	28
4148	35	3	43	11	51	19	59	27
4956	34	2	42	10	50	18	58	26
5764	33	1	41	9	49	17	57	25

final permutation

Bit	Goes to Position							
1-8	58	50	42	34	26	18	10	2
916	60	52	44	36	28	20	12	4
1724	62	54	46	38	30	22	14	6
2532	64	56	48	40	32	24	16	8
3340	57	49	41	33	25	17	9	1
4148	59	51	43	35	27	19	11	3
4956	61	53	45	37	29	21	13	5
5764	63	55	47	39	31	23	15	7

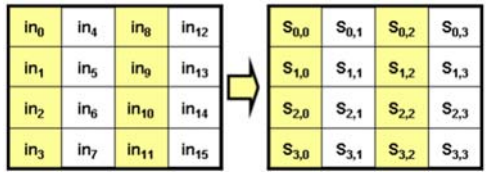
Triple DES (3DES)

- 1999년에 표준으로 지정
- 2개 또는 3개의 키 사용
  - C = E(K_3, D(K_2, E(K_1, P)))
  - K_3=K_1인 방법도 허용 (FIPS 46-3)
- 복호화 방법
  - 암호화 역순
  - P = D(K_1, E(K_2, D(K_3, C)))
- 168-bit 키 크기의 안정성

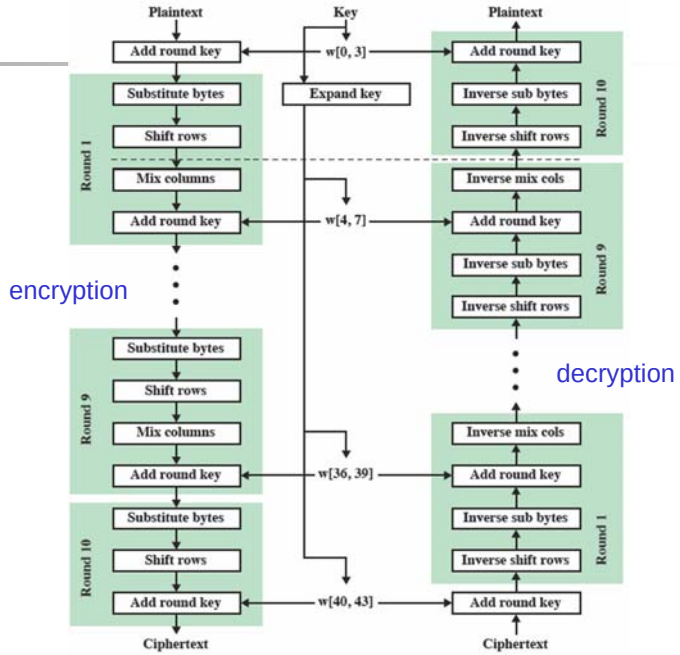


Advanced Encryption Standard (AES)

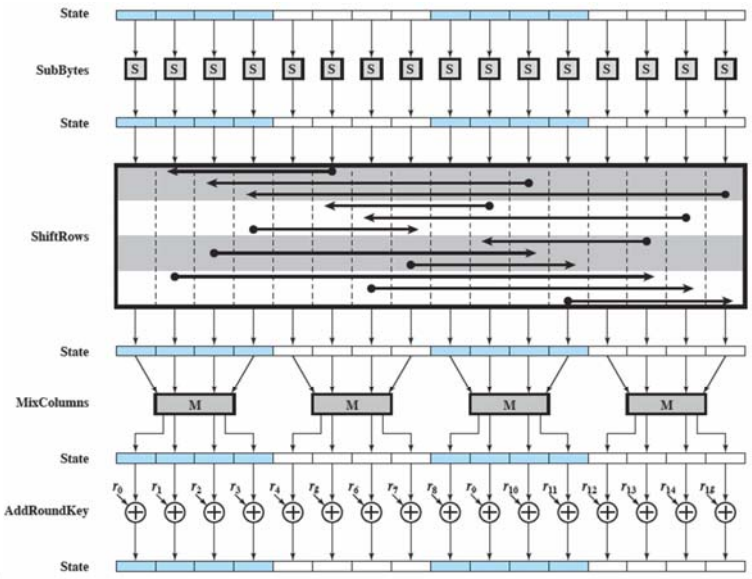
- AES 표준으로 Rijndael 채택 (FIPS 197)
- AES 특징
  - 블록 크기 : 128-bit
  - 키 길이 : 128, 192, 256-bit
  - 여러 라운드로 구성 : 10, 12, 14 round
- AES의 128-bit 블록 (16 bytes)을 4x4 matrix로 표현 → state



- 각 라운드의 구성
  - Substitute bytes
  - Shift Rows
  - Mix Columns
  - Add round key



AES Round Structure



연세대학교

Substitute Bytes

- 두 수학 연산을 연속 수행하여 byte 치환
  - finite field GF(2^8)에서의 inverse 계산: $c = b^{-1}$
  - Affine 변환 :

$$\begin{pmatrix} s_7 \\ s_6 \\ s_5 \\ s_4 \\ s_3 \\ s_2 \\ s_1 \\ s_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_7 \\ c_6 \\ c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

  - 알려진 암호 해독 공격에 대한 저항성
- 실제 치환 연산은 S-box table lookup으로 수행
  - S-box table에 각 바이트 값에 대한 256개의 결과를 미리 저장
- 암호해독은 S-box의 역변환을 사용

연세대학교

S-box, Inverse S-Box

x \ 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	E					
3	04	C7	23	C3	18	96	05	9A	07	12	8					
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D					
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	B					
6	D0	EF	AA	FB	43	4D	33	85	45	F9	0					
7	51	A3	40	8F	92	9D	38	F5	BC	B6	D					
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7					
9	60	81	4F	DC	22	2A	90	88	46	EE	B					
A	E0	32	3A	0A	49	06	24	5C	C2	D3	A					
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F					
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	7					
D	70	3E	B5	66	48	03	F6	0E	61	35	5					
E	E1	F8	98	11	69	D9	8E	94	9B	1E	8					
F	8C	A1	89	0D	BF	E6	42	68	41	99	2					

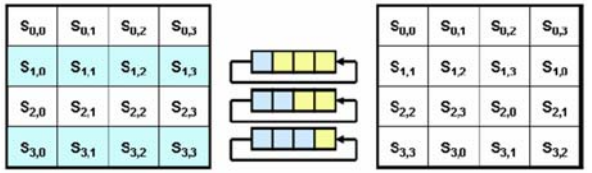
x \	0	1	2	3
0	52	09	6A	D5
1	7C	E3	39	82
2	54	7B	94	32
3	08	2E	A1	66
4	72	F8	F6	64
5	6C	70	48	50
6	90	D8	AB	00
7	D0	2C	1E	8F
8	3A	91	11	41
9	96	AC	74	22
A	47	F1	1A	71
B	FC	56	3E	4B
C	1F	DD	A8	33
D	60	51	7F	A9
E	A0	E0	3B	4D
F	17	2B	04	7E

4x4 matrix

연세대학교

Shift Rows and Mix Columns

- Shift Row - 행 단위로 shift (rotate)



- Mix Column – 열의 원소들을 섞음

$$\begin{bmatrix} s'_{0,i} \\ s'_{1,i} \\ s'_{2,i} \\ s'_{3,i} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} s_{0,i} \\ s_{1,i} \\ s_{2,i} \\ s_{3,i} \end{bmatrix}$$

$$s'_{1,1} = s_{0,1} \oplus 2s_{1,1} \oplus \underbrace{2s_{2,1}}_{3s_{2,1}} \oplus s_{2,1} \oplus s_{3,1}$$

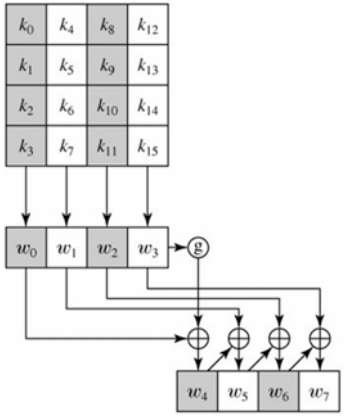
$2s_{2,1}$ is $s_{2,1}$ shifted left one bit

- 두 연산 후에 모든 output bits가 모든 input bits의 영향을 받게 됨

연세대학교

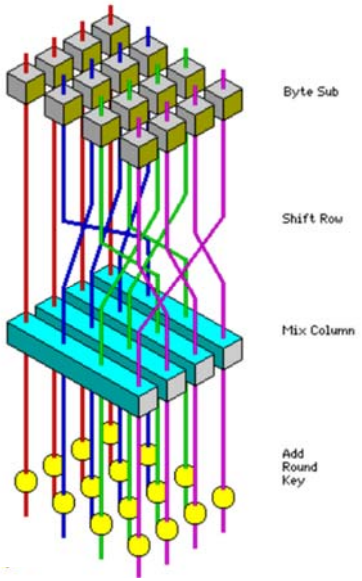
Add Subkey

- Add Subkey
  - mix columns 연산 결과에 각 라운드에 대한 expanded key를 XOR 연산
- AES Key Expansion
  - 이전 키로부터 다음 키를 생성
  - g 함수
    - byte rotate left
    - S-box사용한 byte 치환
    - 라운드 상수와 XOR



연세대학교

AES round structure (3-dimension)



연세대학교

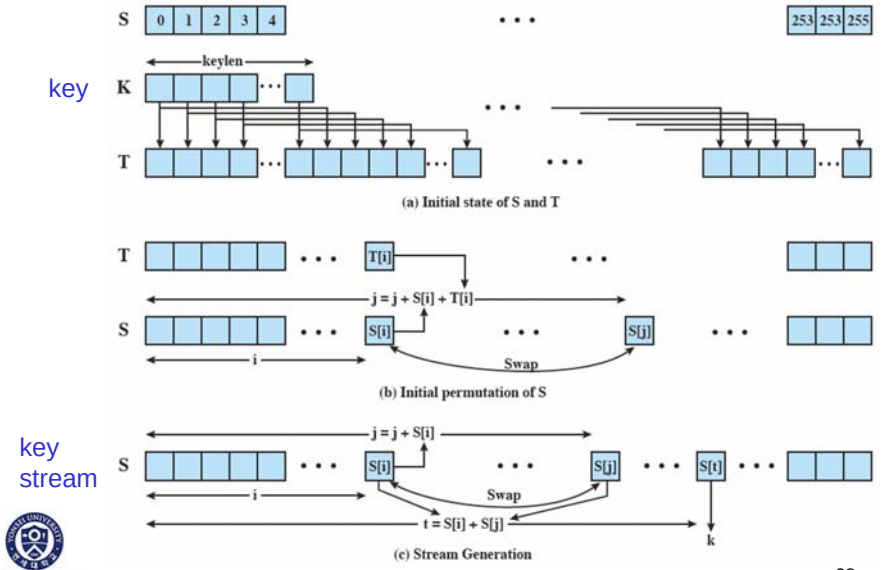
Stream Ciphers

- 입력 원소들을 연속적으로 처리합니다.
- key는 의사 난수 생성기에 대한 입력으로 사용
  - random number stream 생성
  - input key를 모른다면 key stream을 예측할 수 없음
  - 평문과 key stream 출력을 XOR하여 암호문 생성
- 빠르고 적은 크기의 코드
  - RC4 – Ron Rivest가 설계한 stream cipher

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

연세대학교

RC4



연세대학교

- RC4의 용도
  - SSL/TLS (Secure Socket Layer/Transport Layer Security)
    - 웹 브라우저와 서버 간의 통신에 사용
  - WEP (Wired Equivalent Privacy), WPA (WiFi Protected Access)
    - 무선 LAN 표준 프로토콜에서 사용

Block Cipher - Modes of Operation

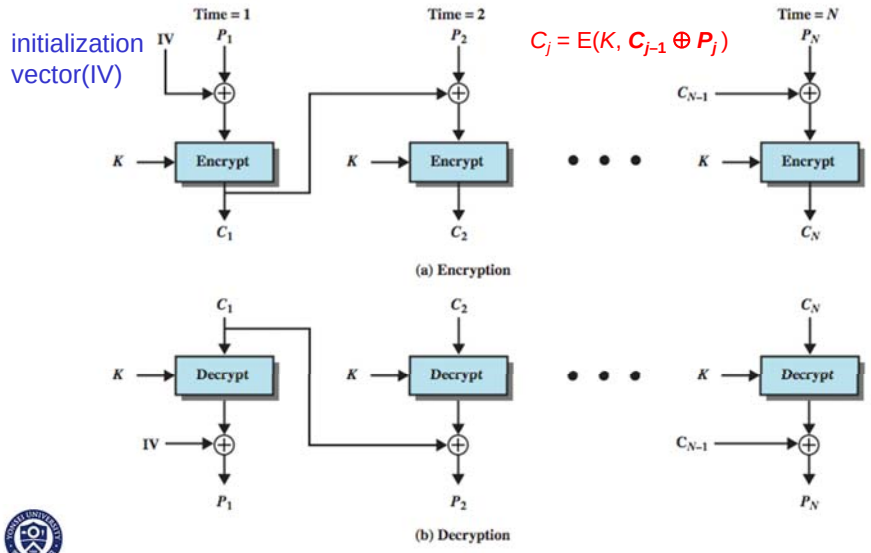
- 블록 암호화 - 블록으로 데이터를 처리
  - 64 비트 (DES, 3DES), 128 비트 (AES)
- 긴 메시지는 여러 블록으로 나누어 처리해야 함
  - 블록 크기의 배수가 되도록 padding도 필요
- 5 가지 동작 모드
  - ECB, CBC, CFB, OFB, CTR (NIST SP 800-38A에 정의)

- 각 동작 모드의 주 용도
  - ECB – 단일 값들의 보안 전송
  - CBC – 범용 블록 보안 전송
  - CFB – 범용 스트림 보안 전송
  - OFB – 잡음이 있는 채널에서 스트림 전송
  - CTR – 범용 블록 보안 전송, 병렬처리 가능 (고속에 유용)

Electronic Codebook (ECB)

- 가장 단순한 모드
- 동일한 키를 사용하여 각 블록을 암호화
- 각 평문 블록에 대해 고유한 암호문 블록 값을 갖고 있기 때문에 "codebook" 이라고 부름
  - 같은 평문에 같은 암호문
  - 평문이 반복되는 경우 안전하지 않으므로 이를 해결하기 위한 기술이 필요함

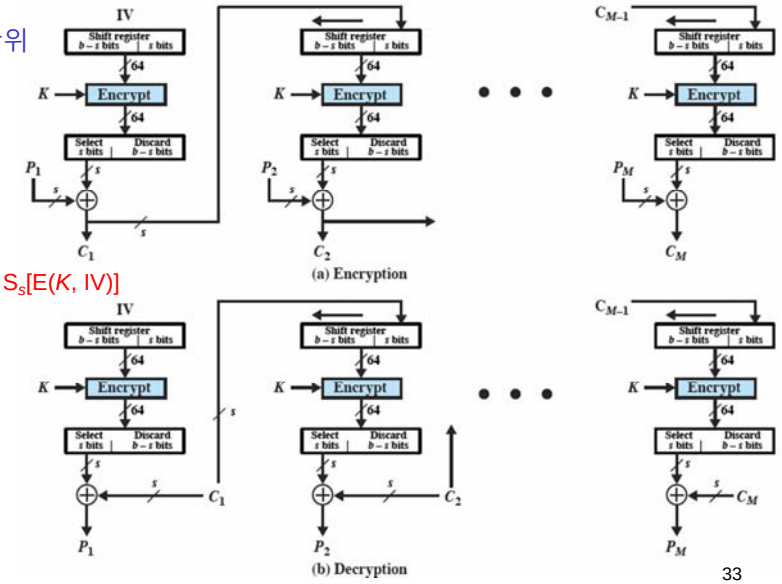
Cipher Block Chaining (CBC)



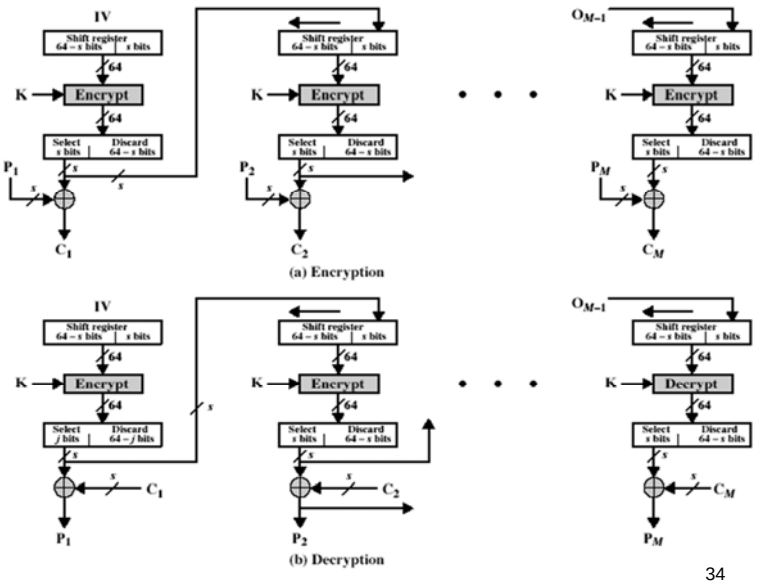
Cipher Feedback (CFB)

s:전송단위

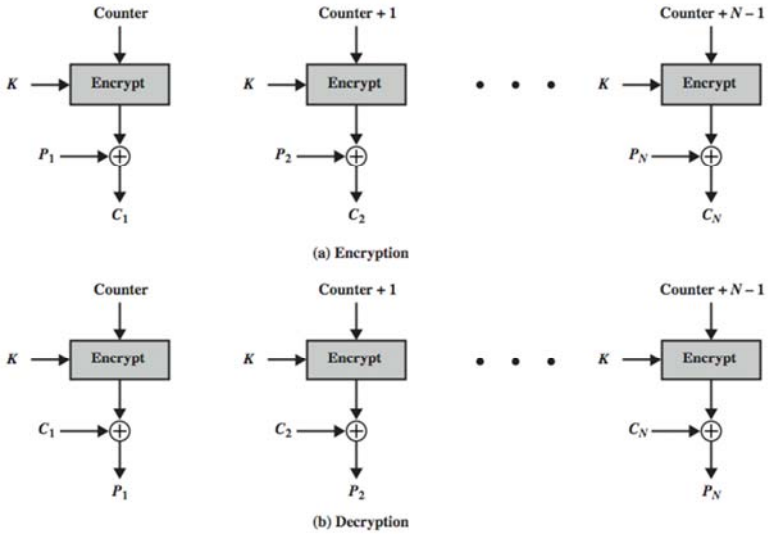
$C_1 = P_1 \oplus S_s[E(K, IV)]$



Output FeedBack (OFB)

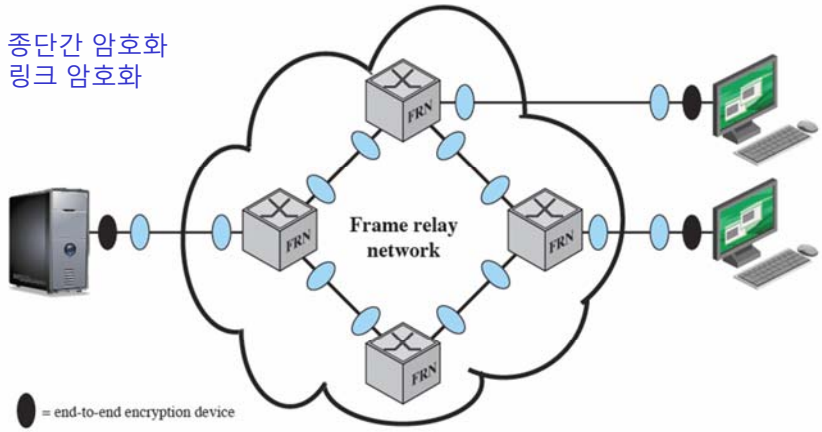


Counter (CTR)



Location of Encryption

- 종단간 암호화
- 링크 암호화



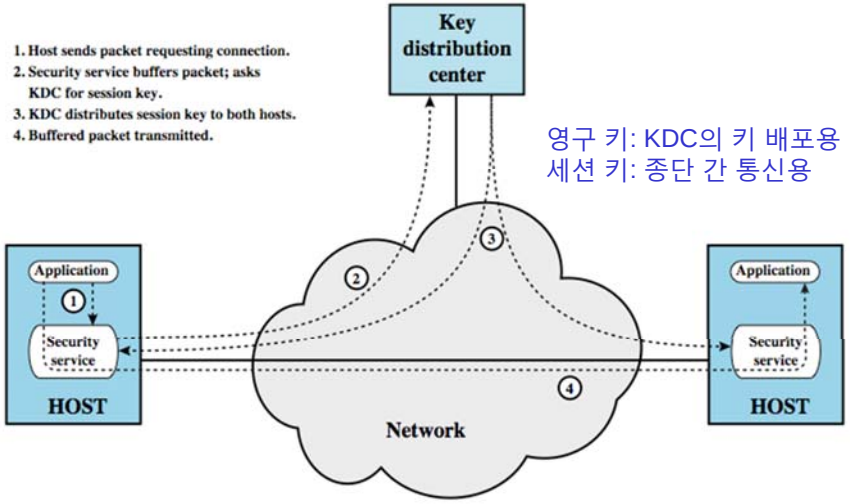
사용자 전송 frame = 헤더 + 데이터

어느 부분을 암호화하는가?

Key Distribution

- 대칭암호화를 위해 두 당사자는 같은 비밀키를 공유해야 함
- 대칭암호화에서의 여러 가지 비밀키 공유 방법
  - A가 키를 선택하고 물리적으로 B에게 전달
  - 제3자가 키를 선택하여, 물리적으로 A, B에게 전달
    - 링크 암호화에서는 reasonable
    - 종단 간 암호화에는 좋은 선택이 아님
  - A, B가 키를 공유한 적이 있으면, A가 새 키를 선택하고, 이전 키를 사용하여 새 키를 암호화하여 B에게 전송
  - 제3자 C와 A,B간에 암호화 연결이 있으면, C가 키를 선택하고 A와 B에게 각각 암호화 전송
    - 종단 간 암호화에 최적
    - C : Key Distribution Center (KDC)

Key Distribution



영구 키: KDC의 키 배포용
세션 키: 종단 간 통신용