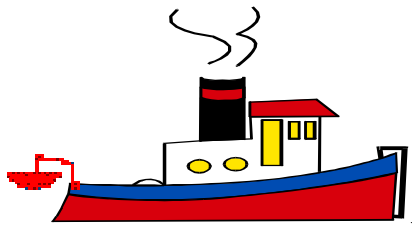
2. 암호의 개념과 대칭키 암호 시스템

담당교수: 차 영욱

ywcha@andong.ac.kr

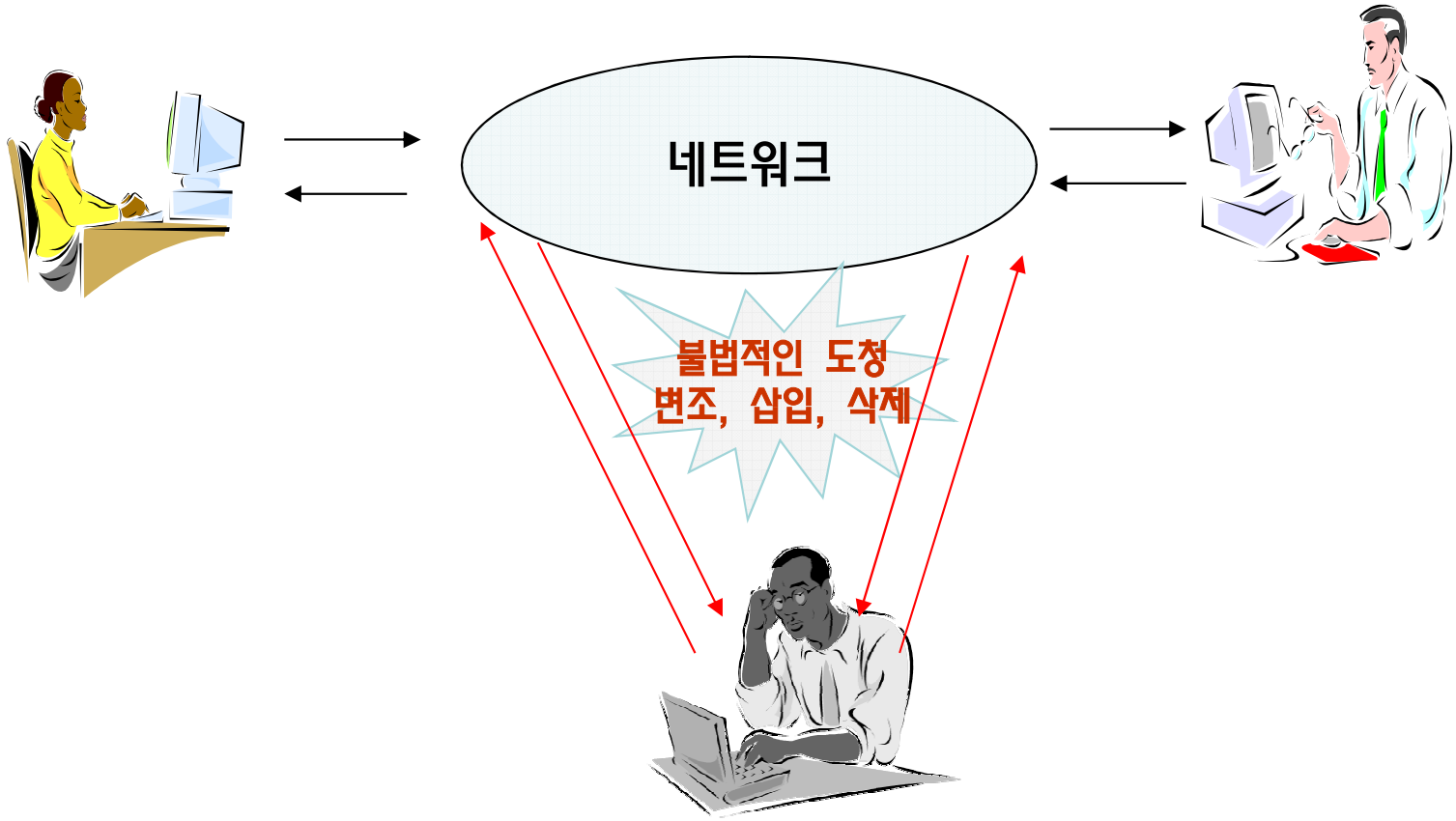
목 차

- 보안 공격 및 보안 서비스
- 암호의 개념 및 역사
- 수동 암호
- 기계 암호
- 현대 암호의 기술
- 데이터 암호화 표준(DES: DATA Encryption Standard)
- 진보된 암호화 표준(AES: Advanced Encryption Standard)
- 삼중 데이터 암호화 표준



보안 공격

- 수동적 공격: 훔쳐보기, 인터넷 전화의 도청
- 능동적 공격: 변조, 삽입, 삭제



보안 서비스

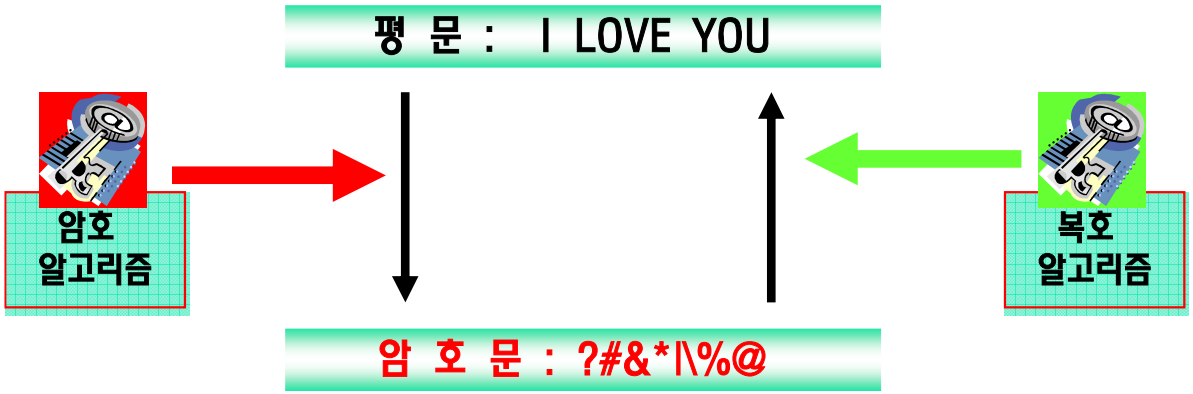
- 안전한 통신을 위한 요구사항
  - 기밀성: 정당한 사용자 만이 데이터의 내용을 파악할 수 있게 함
  - 무결성: 수신된 메시지에 불법적인 삽입이나 변조가 있는지 확인할 수 있게 함

- 암호기술의 도입

보안 서비스의 응용


현실 세계
✓ 주민등록증
✓ 도장, 사인
✓ 화폐
✓ 상거래
✓ 은행
보안기술도입
사이버 세계
✓ 전자주민 카드
✓ 전자서명(인증서)
✓ 전자화폐
✓ 전자상거래
✓ 인터넷 뱅킹

암호의 개념



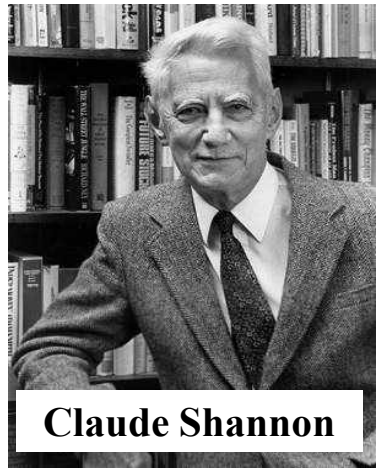
- 암호화: 암호 알고리즘과 암호 키를 이용하여 평문을 암호문으로 변환
- 복호화: 복호 알고리즘과 복호 키를 이용하여 암호문을 평문으로 변환
- 암호해독: 제 3자가 암호문에서 평문을 추정
- 암호학: 암호와 암호 해독을 연구하는 학문

통계적 암호해독

- 통계적 암호해독(statistical cryptanalysis): 통계정보를 이용

- 영어의 각 문자가 문장 중에서 발생할 확률
  - E: 문서에서 나올 확률(0.12)이 가장 높다.
  - D와 L: 0.04의 확률
  - A, H, I, N, O, R, S, T: 0.06~0.09의 확률
  - B, C, F, G, M, P, U, W, Y: 0.015~0.028의 확률
  - J, K, Q, V, 그리고 Z: 0.01보다 약간 작은 확률
  - 이중 문자열이나 삼중 문자열의(각각 2개나 3개 문자로 구성된 문자열) 빈도 → [참조: Stinson, "Cryptography Theory and Practice," CRC Press LLC, 1995].

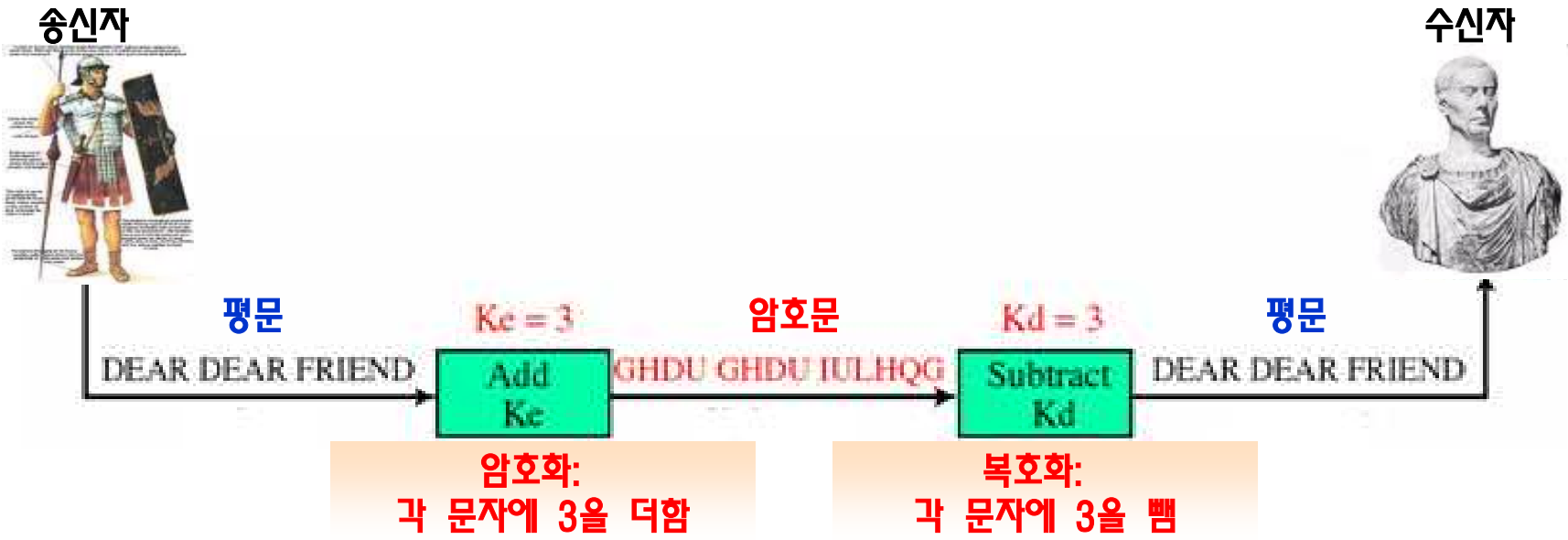
암호의 역사


Claude Shannon

- 수동암호 시대: 고대 ~ 1920
  - 치환 암호, 전치 암호
- 기계암호 시대: 1920 ~1950
  - 복잡한 기계 사용
- 컴퓨터암호 시대: 1950 ~ 현재
  - 근대 암호의 개척자
    - Friedman: 독일군이 사용하던 ENIGMA 암호와 일본군이 사용하던 무라사끼 암호 해독
    - Claude Shannon: 현대 암호학의 아버지로 암호 시스템의 기본 개념인 확산(diffusion)과 혼동(confusion)의 개념 도입(1949년)
      - 확산: 암호문의 통계적 특성이 평문의 통계적 특성과 무관하게 하는 기법
      - 혼동: 암호문의 통계적 특성과 암호키 값과의 관계를 가능한 복잡하게 하는 기법
  - 세계 대전 후 전자 계산기의 출현으로 암호의 실용화 연구 활발
  - 미국과 소련의 냉전 시대에 암호 기술의 연구 가속화

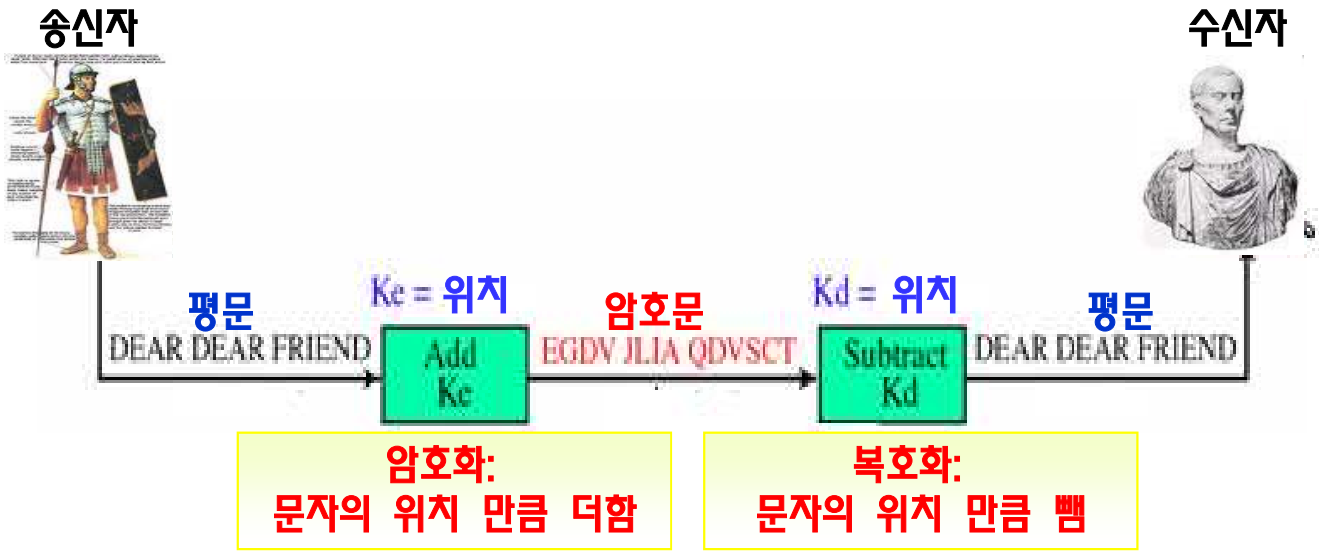
수동 암호-치환 암호(1/2)

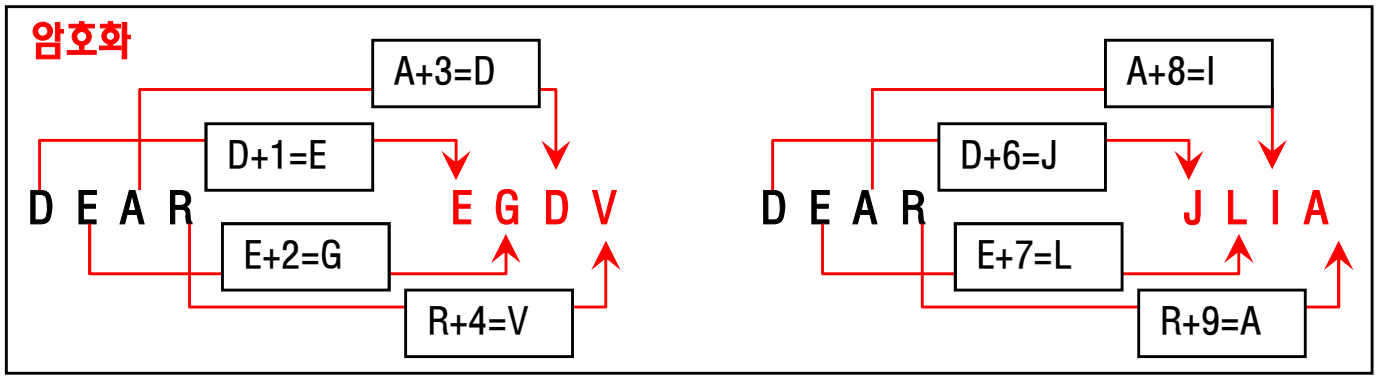
- 치환 암호: 단일문자 치환, 다중문자 치환
- 단일문자 치환: 각 문자를 일정한 길이 만큼 이동
  - 줄리어스 시저의 암호: 문자를 3 만큼 이동하는 치환암호
  - 암호키를 찾기 위한 방법: 읽을 수 있는 평문을 얻을 때까지 모든 가능한 키의 값을 사용하여 복호화



수동 암호-치환 암호(2/2)

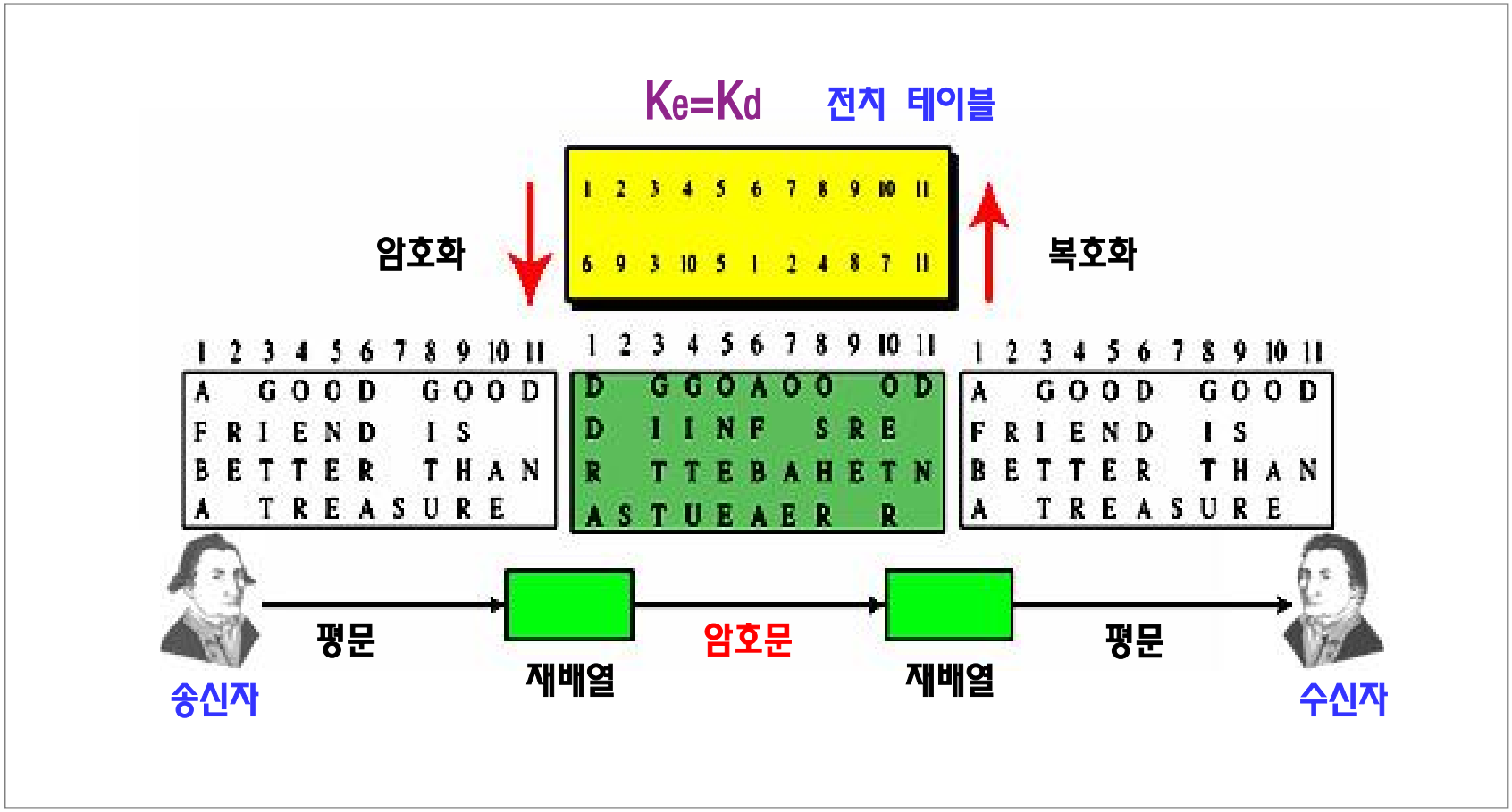
❑ 다중문자 치환: 문장 내에서 문자의 위치 만큼 각 문자를 이동





수동 암호-전치 암호

- **전치암호:** 전치 테이블을 이용하여 문장 내의 문자 위치를 재배열



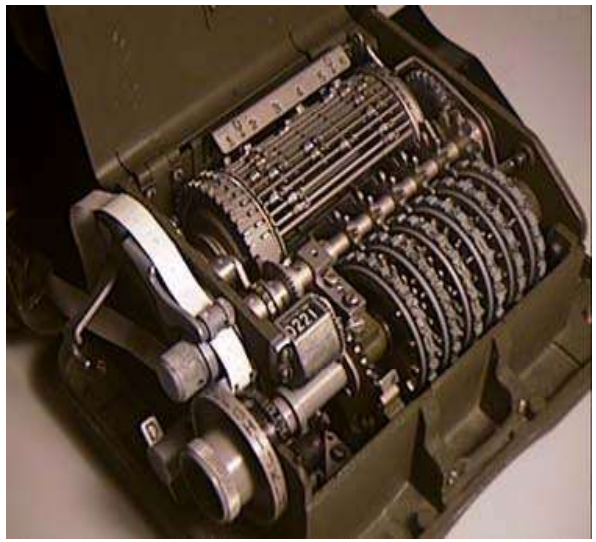
기계 암호

❑ Enigma 암호

- 제 2차 세계대전 시 독일군이 사용
- 자판에 평문을 입력하면 각 회전자에 의하여 변환된 암호문이 나오도록 설계

❑ Hagelin 암호

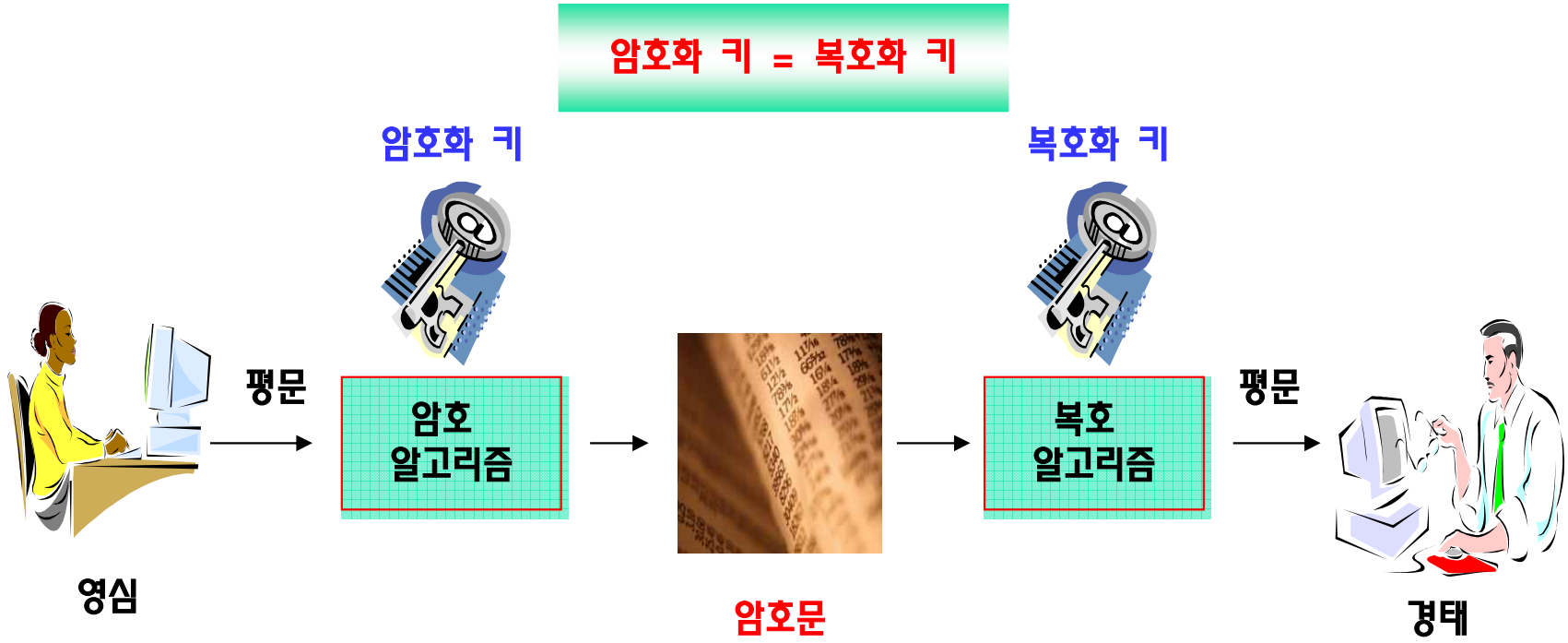
- 제 2차 세계대전시 연합군이 사용
- 1950년 한국 전쟁 때 미군 사용



현대 암호의 기술(1/2)

❑ 대칭키(관용키) 암호 시스템

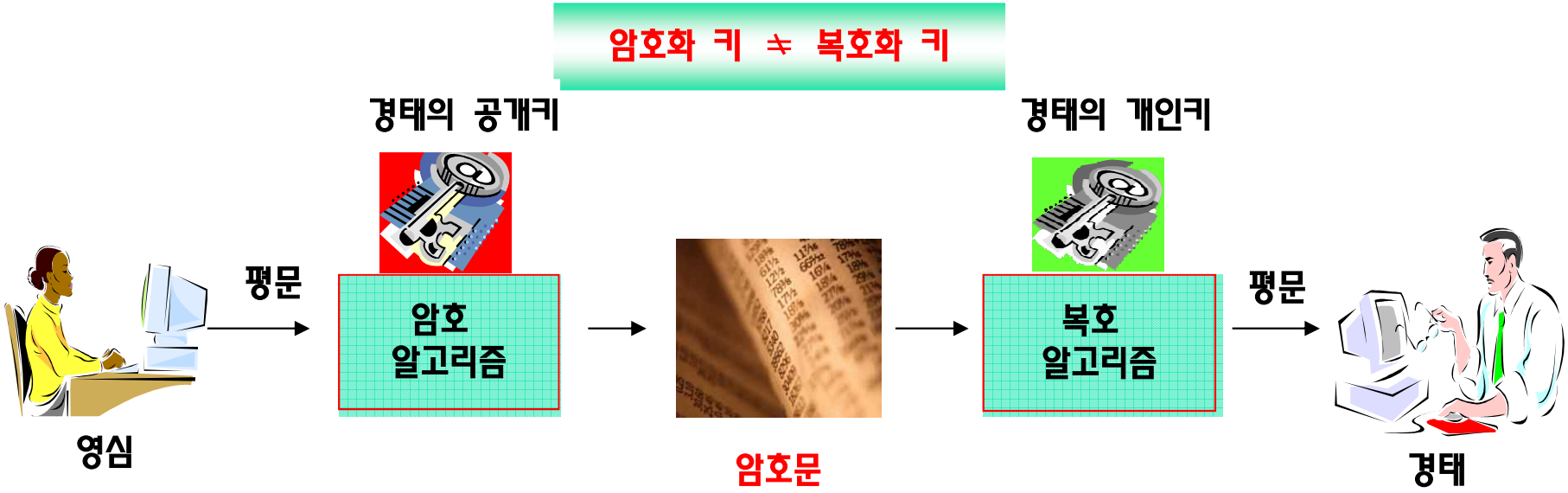
- 암호화와 복호화에 동일한 키 사용



현대 암호의 기술(2/2)

❑ 비대칭키 암호 시스템

- 송신자는 수신자의 공개키로 평문을 암호화
- 수신자는 자신의 개인키로 암호문을 복호화
- 공개키 암호 시스템



Feistel 암호

- IBM의 Horst Feistel: Shannon의 개념을 채용한 최초의 암호 시스템 개발 (1973년)

- Feistel 암호의 다음 특성은 대부분의 대칭키 암호 시스템에 적용
  - 2번 이상의 기본대치 및 순열치환(permutation)을 연속적으로 수행
  - 보통 암호 알고리즘에서 Feistel 연산은 짝수 라운드(DES에서는 16라운드) 적용
  - 키를 각 라운드에서 사용되는 서브키로 변환하는 키 스케줄 알고리즘 활용
  - 라운드 함수에 관계없이 역변환이 가능(암/복호화 과정이 같음)
  - H/W 및 S/W로 구현이 용이하며, 알고리즘의 수행 속도가 빠름

데이터암호화 표준과 기본 연산

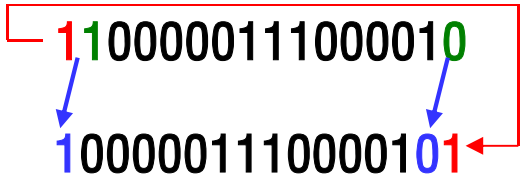
- 1973년: 현재 NIST로 알려진 미국의 국가표준국(NBS)에서 암호 알고리즘의 표준화 요구사항 제시
- 1976년: Horst Feistel이 이끄는 IBM의 연구팀에서 개발된 암호 시스템을 미국의 데이터암호화 표준(DES: Data Encryption Standard)으로 승인
  - 56 비트의 키를 이용하는 대칭키 암호 시스템
  - 데이터를 64비트 단위의 블록으로 분할 후, 순열, 배타적 OR, 회전 등으로 변경
  - 1983년, 1988년, 그리고 1993년에 다시 국가 표준으로 재확인

- 배타적 OR(XOR): 2 비트의 입력이 다르면 출력이 1

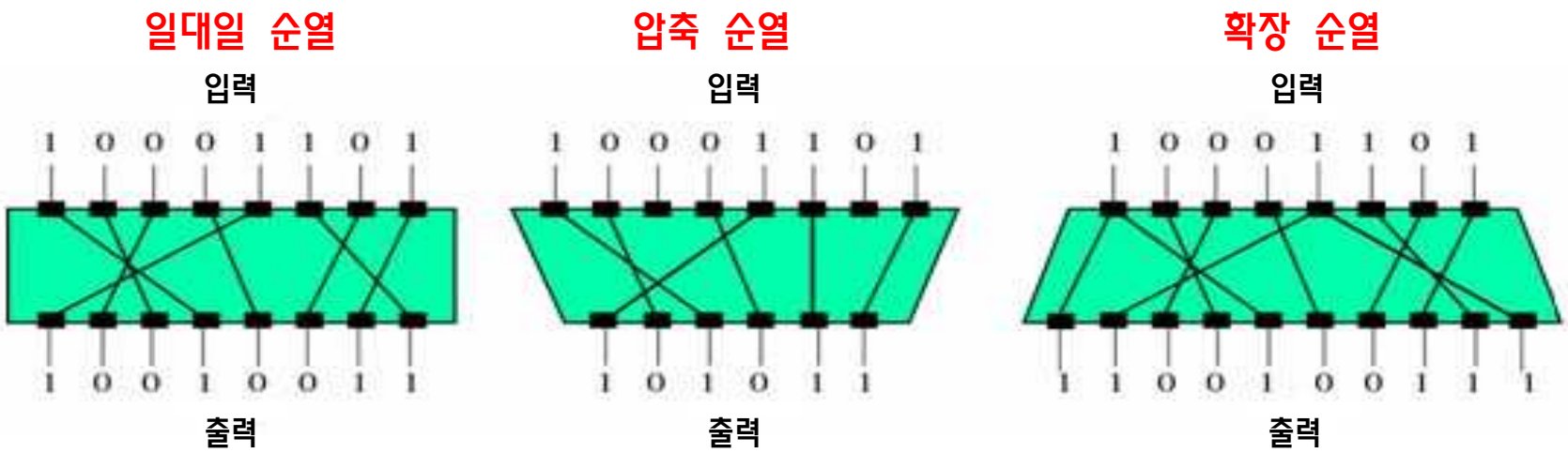
INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

DES의 기본 연산

- **순환:** 비트를 왼쪽이나 오른쪽으로 회전
  - 좌순환: 가장 왼쪽의 비트는 가장 오른 쪽으로 이동하며, 나머지 비트들은 왼쪽으로 한 비트씩 이동

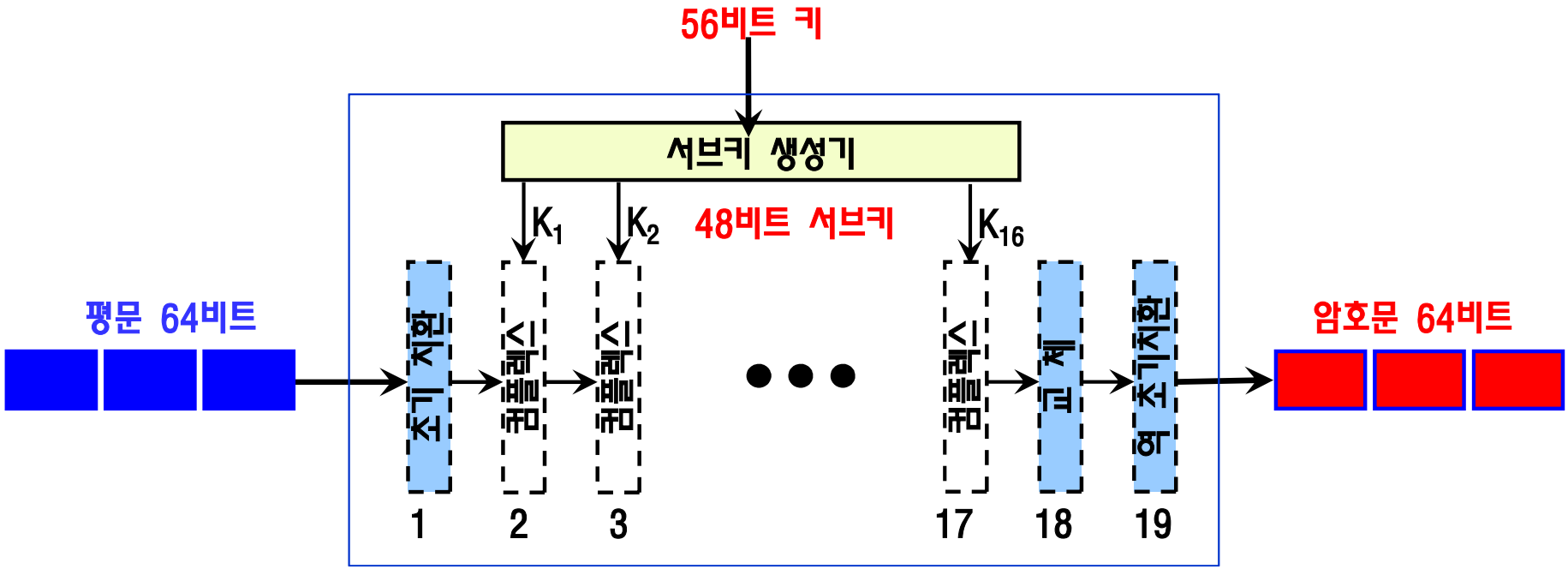


- **순열:** 비트 레벨의 치환



DES-암호화

- 암호화
  - 단계 1: 64비트 평문에 있는 각 비트의 위치를 바꿈(초기 치환)
  - 단계 2-17: 동일한 절차를 수행하나 서로 다른 서브키 사용
  - 단계 18: 64비트 입력을 32비트로 나누어서 위치를 교체
  - 단계 19: 단계 18의 출력인 64비트의 각 비트 위치를 바꿈(역초기 치환)



DES-초기 및 역초기 치환

❑ 1단계: $X_1 = IP(P)$
- P: 64비트 블록 평문
- X_1: 초기 치환함수(IP)의 64비트 출력

❑ 19단계: $C = IP^{-1}(X_{19})$
- X_{19}: 19 단계의 64비트 출력
- C: 역초기 치환함수(IP^{-1})의 출력인 64비트 암호문

$IP(b_1) = b_{40}$
$= IP^{-1}(b_{40}) = b_1$

$IP(b_2) = b_8$
$= IP^{-1}(b_8) = b_2$

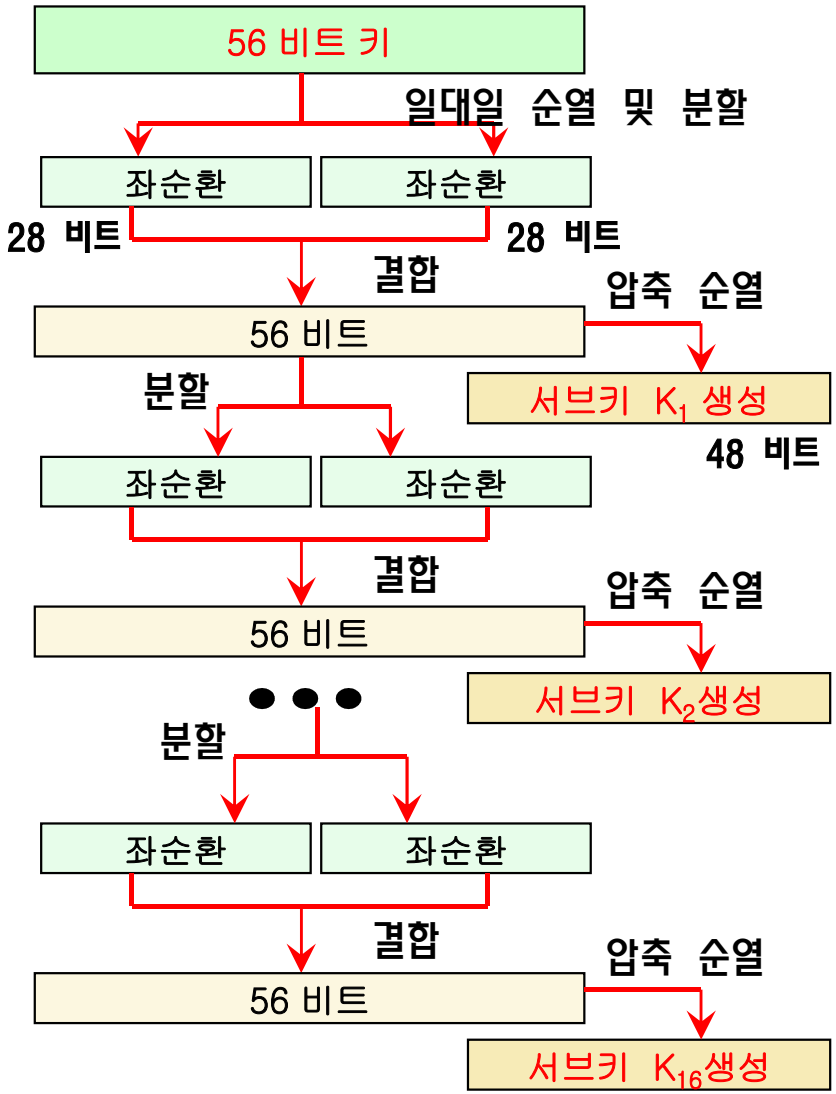
(a) 초기 순열(IP)

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

(b) 역초기 순열(IP^{-1})

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

DES-서브키 생성



① 56 비트 키를 치환 테이블 1(PC-1)을 이용하여 치환 후 28 비트로 분할

② 분할된 28비트 각각을 좌순환 테이블의 각 라운드의 횟수에 해당하는 만큼 좌순환

③ 위의 결과를 결합하여 56 비트를 만들고 치환 테이블 2(PC-2)를 이용하여 압축 순열을 수행하여 48비트의 서브키 생성

④ 위 과정을 16번째까지 반복하며, 각 단계에서 생성된 서브키가 각 단계의 컴플렉스에 서브키로 사용

PC-1

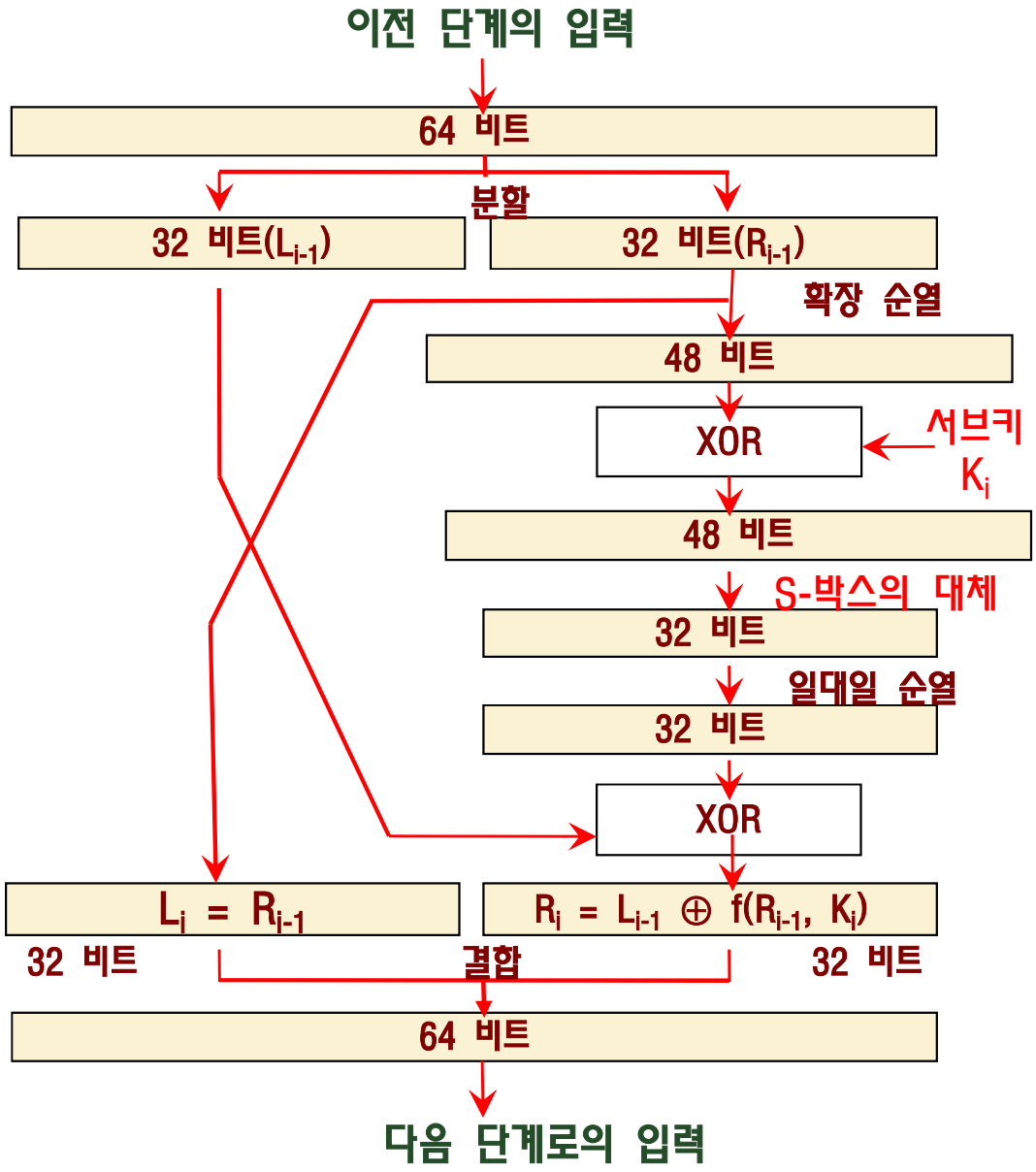
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	63	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC-2

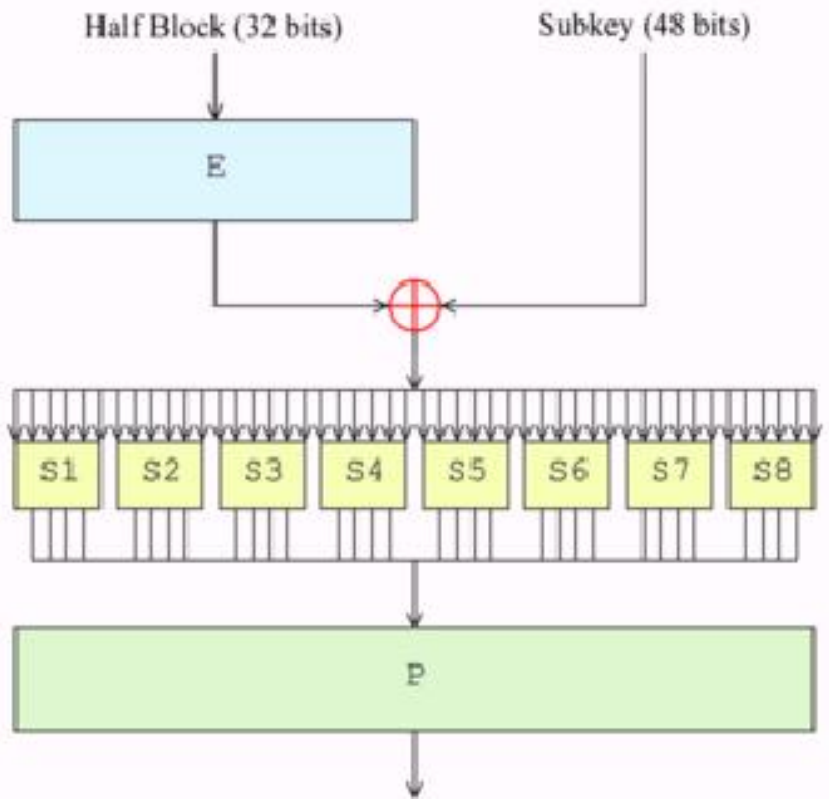
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
4	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

컴플렉스	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
좌순환	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

DES-컴플렉스



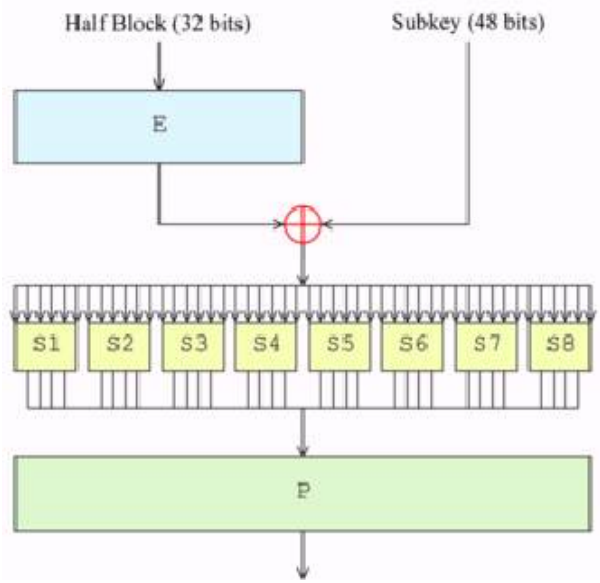
- 16 번의 동일한 컴플렉스 수행
  - 각 라운드에서 서로 다른 서브키 K_i(48 비트) 사용
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
- $f(R_{i-1}, K_i)$



DES 컴플렉스의 $f(R_{i-1}, K_i)$

❑ 치환 테이블

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



❑ S-박스 테이블의 대체(6비트 입력 → 4비트 출력)

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

S_8

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

❑ 입력(101011) S_1-박스 대체→ 출력(9)
- 11: 3 행
- 0101: 5 열

❑ 입력(110010) S_8-박스 대체→ 출력(6)
- 10: 2 행
- 1001: 9 열

DES의 8개 S-박스(1/2)

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

S_2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	14	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

DES의 8개 S-박스(2/2)

S_5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

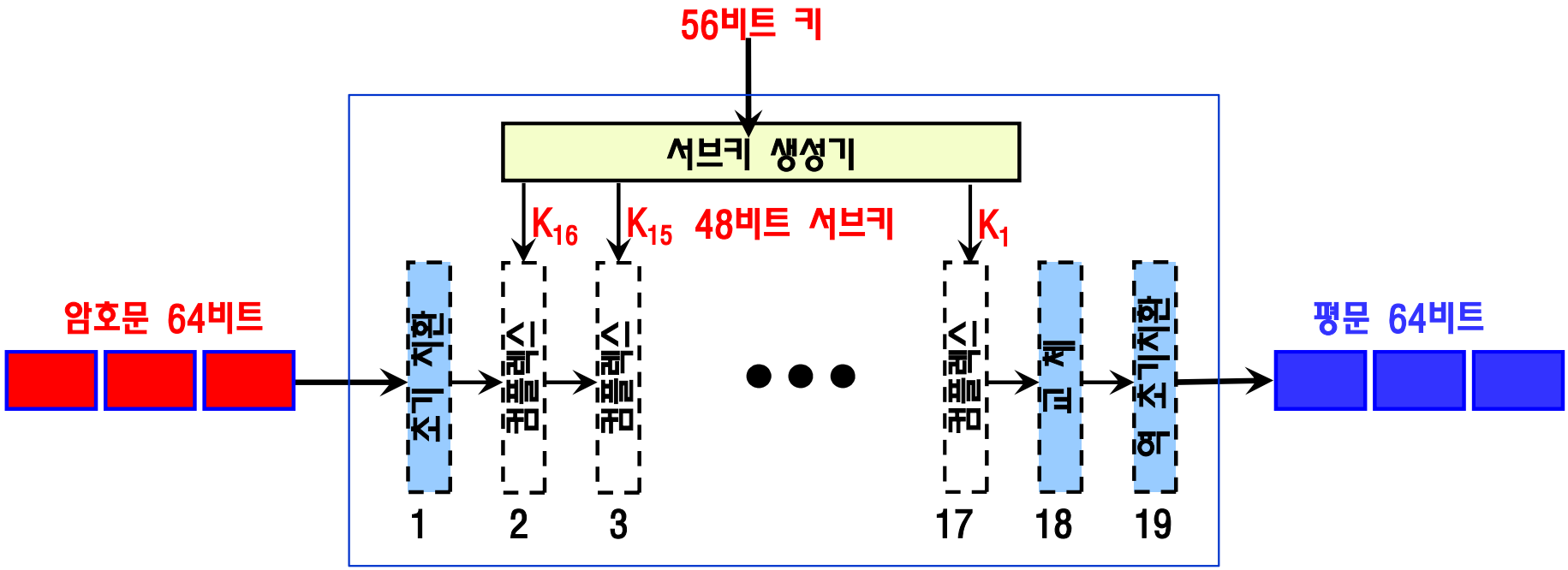
DES의 혼동/확산

- S-박스의 대체: 체계적으로 어떤 비트들의 유형을 다른 비트들로 전환함으로써 혼동 성질을 제공
  - 암호문의 통계적 특성과 암호 키 값과의 관계를 가능한 복잡하게 하는 혼동 기법

- 치환 테이블: 비트들의 순서를 치환(재배열)함으로 확산의 효과 제공
  - 암호문의 통계적 특성이 평문의 통계적 특성과 무관하도록 하는 확산 기법

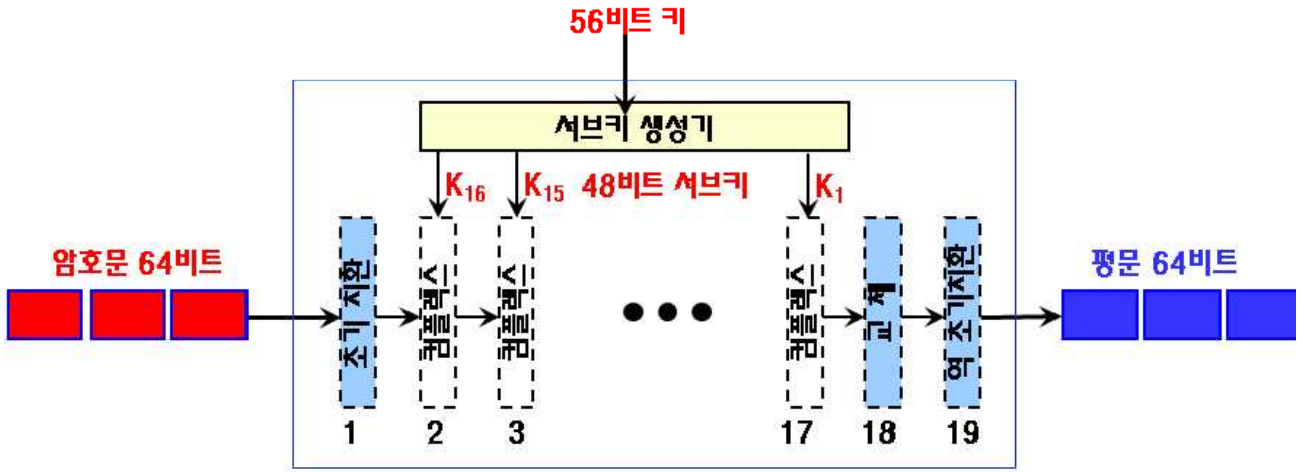
DES-복호화

- 복호화: 암호화와 동일한 치환 테이블 적용
  - 단계 1: 64비트 암호문의 초기 치환
  - 단계 2-17: 서브키의 생성 절차는 암호화와 동일하나 암호화의 역순으로 서브키를 적용
  - 단계 18: 64비트 입력을 32비트로 나누어서 위치를 교체
  - 단계 19: 단계 18의 출력인 64비트의 역초기 치환



DES-복호화와 암호화의 비교

- 암호화와 다른 부분
  - 입력으로 암호문이 주어지며 출력으로 평문이 출력된다.
  - 보조키 적용 순서는 암호화 과정의 역순으로 적용 ➔ 라운드 1(K16), 라운드 2(K15), … 라운드 16(K1)
- 암호화와 같은 부분
  - 서브키 생성 부분
  - 각 라운드의 F 함수 적용과 위치변환
  - 초기 치환과 역초기 치환의 적용 순서



DES의 취약성

- 1976년: 56 비트 키를 사용하는 DES를 미국의 암호 표준으로 승인
- 컴퓨터 속도의 비약적 발전으로 DES의 56 비트 키에 대한 안전성 문제 제기
  - 1997년 1월: RSA는 DES 키를 찾는데 $10,000 달러의 포상금 공모(DES Challenge)
  - 1997년 6월: Roche Verse는 인터넷에 연결된 70,000 대 이상의 컴퓨터를 이용하여 전사공격(brute force) 프로그램으로 96일 만에 DES 키를 찾음
  - 1998년: $250,000 달러의 비용으로 EFF(Electronic Frontier Foundation)가 설계한 크래커(Deep Crack)가 전사공격을 이용하여 56시간 만에 키를 깨뜨림
  - 1999년 1월: EFF의 Deep Crack와 distributed.net이 공동으로 22시간 15분만에 키를 깨뜨림
- EFF의 Deep Crack: 6개의 캐비닛, 29개의 보드, 각 보드에 64개의 키 검색 전용 마이크로 칩 탑재

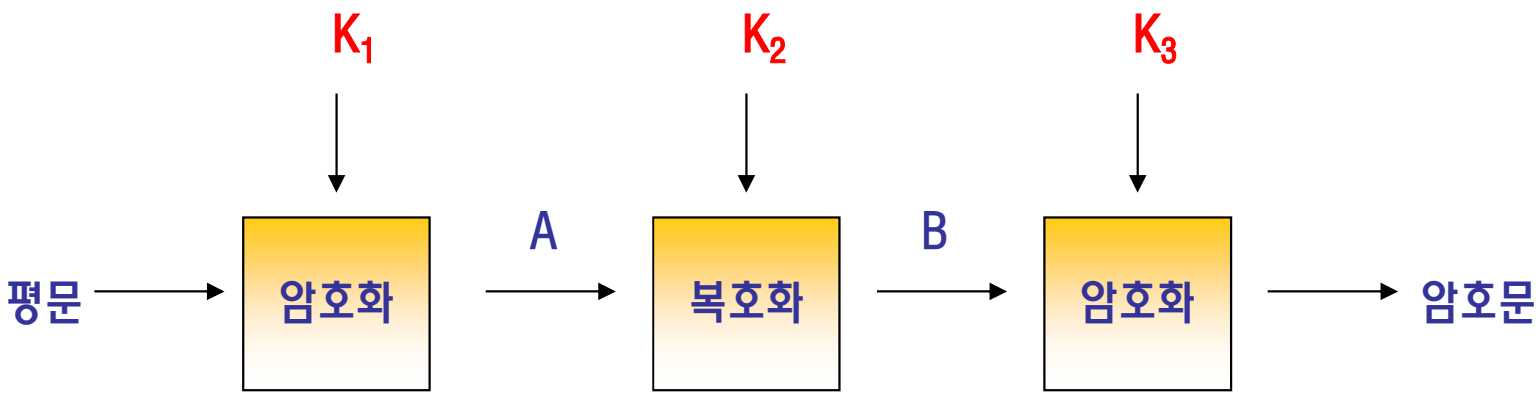
출처: http://www.eff.org/descracker.html

진보된암호화표준

- NIST는 DES의 취약성을 극복하기 위하여 진보된암호화표준(AES: Advanced Encryption Standard)의 개발 계획을 1997년 발표
- 1999년 10월: 과도기에 대비하여 DES의 안전한 변형인 삼중 DES를 미국의 국가 표준으로 채택
- AES 요구사항
  - 형태: 강력한 대칭키 블록 암호 알고리즘으로 정부 및 상업 부분에서 사용 가능
  - 효율성: 삼중 DES 보다 좋을 것
  - 비용: 알고리즘 공개 및 로열티 없이 무료로 이용
  - 안전성:
    - 블록: 적어도 128 비트의 크기
    - 키 범위: 128, 192, 그리고 256 비트
- 2000년 10월: 15개의 제안 중에서 벨기에 출신의 J. Daemen과 V. Rijmen이 제안한 Rijndael 알고리즘을 최종 선정

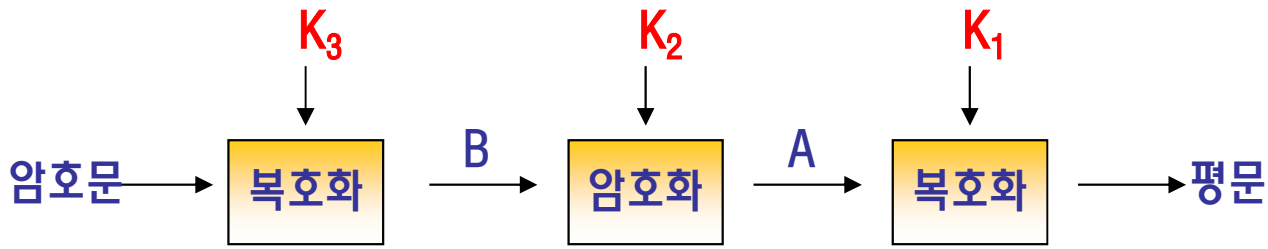
삼중 데이터암호화표준(1/2)

- DES의 안전한 변형 알고리즘으로 널리 사용되고 있는 64비트 블록 단위의 대칭키 암호방식 중 가장 안전
- NIST는 2005년 DES를 공식적으로 퇴출하였으며, 2030년까지 3 중 DES의 사용을 승인
- 암호화: 64비트 평문에서 64비트 암호문 생성
  - K_1, K_2, K_3는 56비트의 DES 키
  - 1단계: 평문을 키 K_1로 암호화
  - 2단계: 1 단계의 결과를 키 K_2로 복호화
  - 3단계: 2 단계의 결과를 키 K_3으로 암호화



삼중 데이터암호화표준(2/2)

- 복호화: 64비트 암호문에서 64비트 평문 유도
  - 1단계: 암호문을 키 K_3으로 복호화
  - 2단계: 1 단계의 결과를 키 K_2로 암호화
  - 3단계: 2 단계의 결과를 키 K_1로 복호화



- 높은 보안을 요구하는 경우: K_1, K_2, K_3은 서로 다른 키 사용
  - 56*3 즉, 168비트 길이의 키로 암호화하는 것과 동일한 안전성
- 일반적인 보안을 요구하는 경우: K_1과 K_3은 동일한 키 사용

SEED

- 2000년까지 미국 정부는 미국 내에서 사용되는 웹브라우저는 128 비트 수준의 보안접속을 허용했으나, 미국 외로 수출되는 Microsoft Internet Explorer, Netscape 웹브라우저는 40 비트 이상의 보안접속을 하지 못하도록 금지해 왔다.
  - 40 비트 수준의 암호화는 1997년 경의 연산능력으로도 3.5 시간 만에 깨짐
- 국내 보안업체들은 128 비트 보안접속을 위한 별도의 알고리즘 개발을 위하여 노력했으며, SEED는 1999년 한국정보보호진흥원의 기술진이 개발한 블록 암호 알고리즘이다.
  - 개발된 SEED는 웹브라우저의 플러그인으로 배포되어 국내 인터넷 뱅킹에 사용
  - 16-라운드 Feistel 네트워크
  - 128 비트 블록 및 128 비트 키

블록 암호와 모드

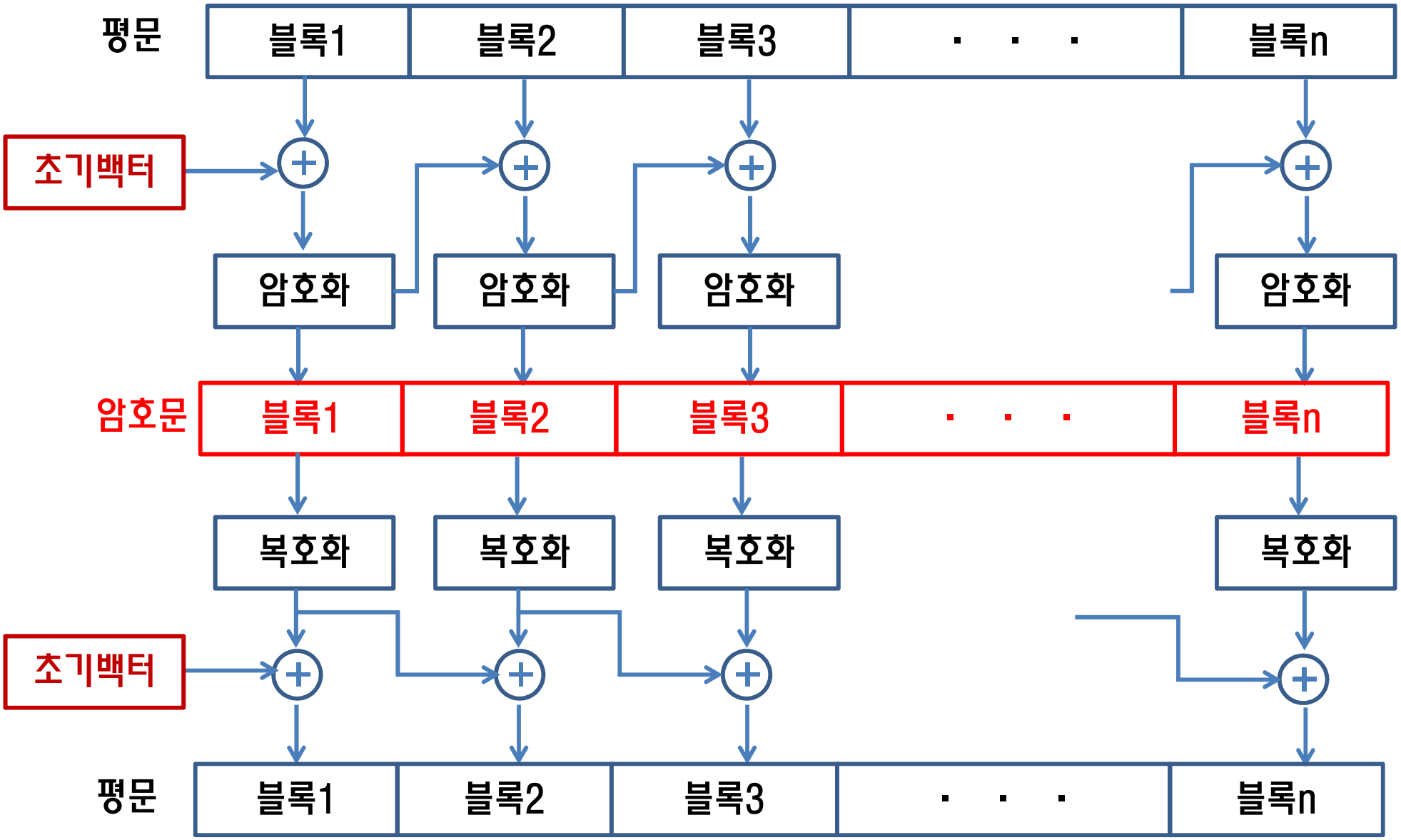
- 블록암호(block cipher) 알고리즘: 특정 비트 수의 집합을 한번에 처리하는 암호 알고리즘
  - DES, 3중 DES: 블록 길이가 64 비트
  - AES: 블록 길이가 128 비트
- 블록암호의 모드: 긴 평문을 암호화하기 위해서 블록 암호 알고리즘을 반복해서 사용하는 방법
  - ECB(Electric CodeBook) 모드
  - CBC(Cipher Block Chaining) 모드
  - CFB(Cipher-FeedBack) 모드
  - OFB(Output-FeedBack) 모드
  - CTR(CounTeR) 모드
- 모드의 선택
  - ECB 모드는 취약성이 높으므로 사용하지 않을 것을 주장
  - CBC와 CTR 모드의 사용을 권장
  - 참고: Practical Cryptography, Schneier 2003

ECB 모드의 암호화/복호화

❑ 평문 블록과 암호 블록이 일대일 관계를 유지

- 같은 값을 갖는 평문 블록이 여러 개 존재 ➔ 모두 같은 값의 암호문으로 변환
- 암호문을 살펴보므로 평문 속에 패턴의 반복이 있음을 알게 되므로 암호 해독의 실마리가 될 수 있음.

CBC 모드의 암호화/복호화(1/2)



CBC 모드의 암호화/복호화(2/2)

- 한 단계 앞에서 수행된 암호문 블록과 평문 블록을 XOR한 후 암호화를 수행
  - 평문 블록 1과 2의 값이 같은 경우에도 암호문 블록 1과 2의 값이 같다고 할수 없음
  - ECB 모드의 단점이 CBC 모드에서는 나타나지 않음

- IPSec에서 통신의 기밀성을 위하여 CBC 모드를 사용
  - 3DES-CBC, AES-CBC

요점 정리(1/3)

- 안전한 통신을 위한 요구사항
  - 기밀성: 정당한 사용자 만이 데이터의 내용을 파악할 수 있게 함
  - 무결성: 수신된 메시지에 불법적인 삽입이나 변조가 있는지 확인할 수 있게 함

- 현대암호 기술
  - 대칭키 암호 시스템: 암호화와 복호화에 동일한 키 사용
  - 비대칭키(공개키) 암호 시스템:
    - 송신자는 수신자의 공개키로 평문을 암호화
    - 수신자는 자신의 개인키로 암호문을 복호화

- 대칭키 암호 시스템에 적용된 Feistel 암호의 특성
  - 2번 이상의 기본대치 및 순열치환(permutation)을 연속적으로 수행
  - 보통 암호 알고리즘에서 짝수 라운드(DES에서는 16라운드) 적용
  - 키를 각 라운드에서 사용되는 서브키로 변환하는 키 스케줄 알고리즘 활용
  - 라운드 함수에 관계없이 암/복호화 과정이 같음

요점 정리(2/3)

- 1976년: Horst Feistel이 이끄는 IBM의 연구팀에서 개발된 암호 시스템을 미국의 데이터암호화표준(DES)으로 승인
  - 데이터를 64비트 단위의 블록으로 분할 → 56 비트의 키 적용

- DES의 혼동과 확산
  - S-박스의 대체: 암호문의 통계적 특성과 암호 키 값과의 관계를 가능한 복잡하게 하는 혼동 성질을 제공
  - 치환 테이블: 암호문의 통계적 특성이 평문의 통계적 특성과 무관하도록 하는 확산 성질 제공

- 과도기에 대비하여 DES의 안전한 변형인 삼중 DES를 미국의 국가 표준으로 채택(1999년 10월)

요점 정리(3/3)

- 진보된 암호화 표준(AES)
  - DES의 취약성을 극복하기 위하여 1997년 부터 개발 시작
  - 15개의 제안 중에서 벨기에 출신의 J. Daemen과 V. Rijmen이 제안한 Rijndale 알고리즘 선정(2000년 10월)
  - 블록: 적어도 128 비트의 크기
  - 키 범위: 128, 192, 그리고 256 비트
- SEED
  - 1999년 한국정보보호진흥원의 기술진이 개발한 블록 암호 알고리즘
  - 국내 인터넷 뱅킹에 사용
  - 16-라운드 Feistel 네트워크, 128 비트 블록 및 128 비트 키
- 블록암호의 모드
    - ECB(Electric Code Book) 모드
    - CBC(Cipher Block Chaining) 모드
    - CFB(Cipher-FeedBack) 모드
    - OFB(Output-FeedBack) 모드
    - CTR(CounTeR) 모드