정보보호론

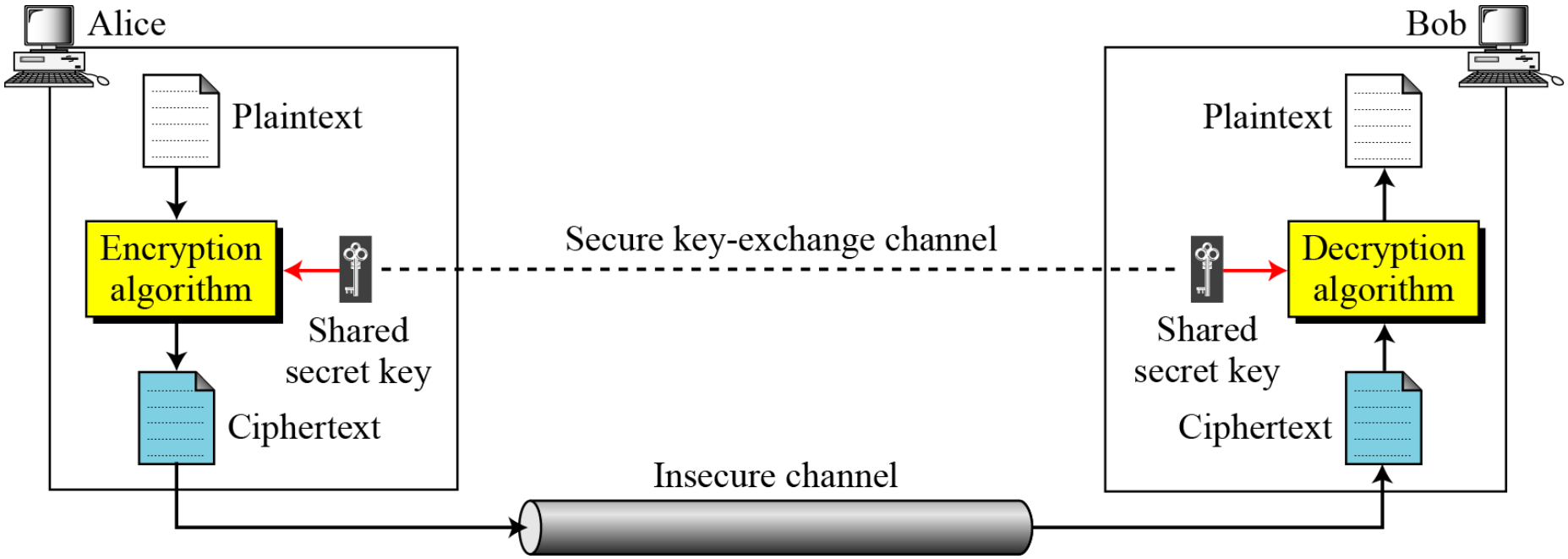
대칭키 암호 시스템

한림대학교 소프트웨어융합대학 조효진

Contents

- 대칭키 암호시스템
- Data Encryption Standard (DES)
- Advanced Encryption Standard (AES)
- 스트림 암호

대칭키 암호 시스템


Alice
Bob
Plaintext
Plaintext
Encryption algorithm
Secure key-exchange channel
Decryption algorithm
Shared secret key
Shared secret key
Ciphertext
Ciphertext
Insecure channel

Review: Terms related to encryption

- Plaintext: This is the original message or data that is fed into the algorithm as input.

- Encryption algorithm: The encryption algorithm performs various substitutions and transformations on the plaintext.

- Secret key: The secret key is also input to the algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.

- Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts.

- Decryption algorithm: This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the same secret key and produces the original plaintext.

대칭키 암호 시스템

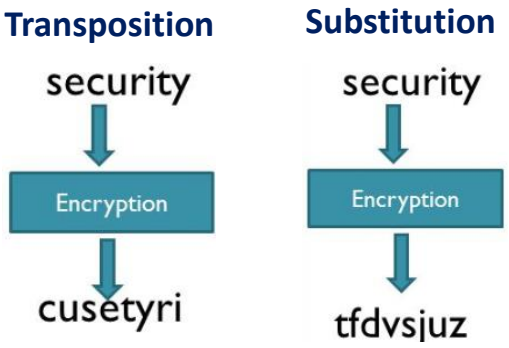
❒ 혼돈 (Confusion)과 확산 (Diffusion)

- 정보이론 학자인 샤논(Claude Shannon) 제안
  - 혼돈 (Confusion): 키와 암호문과의 관계를 감추는 성질
  - 현대 블록암호는 혼돈을 위해 치환 (Substitution)사용
- 확산 (Diffusion): 평문과 암호문과의 관계를 감추는 성질
  - 평문 한 비트의 변화가 암호문의 모든 비트에 확산
  - 주로 평문과 암호문의 통계적 성질을 감추기 위해 사용 (주로 Transposition과 연관됨)

Shannon

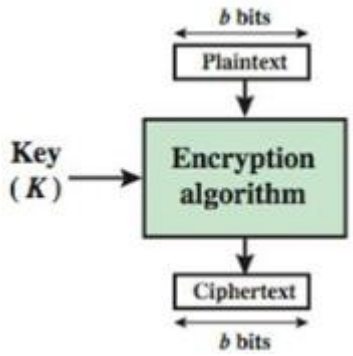
❒ Product 시스템

- Substitution과 Transposition이 함께 사용되는 암호



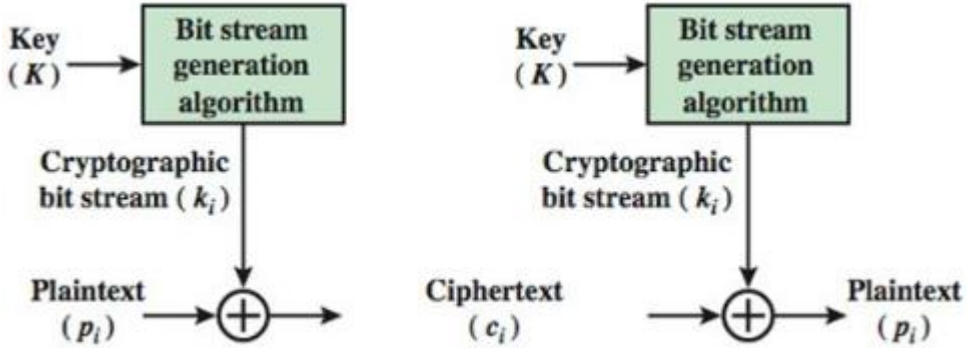
대칭키 암호 시스템

- 대칭키 암호 시스템은 블록 암호와 스트림 암호 2가지 형태로 나뉘어짐
  - 블록 암호(Block Cipher): 암호화/복호화 과정을 하나의 블록단위로 수행함
  - 스트림 암호(Stream Cipher): 암호화/복호화 과정을 하나의 구성단위(i.e., 1bit)로 수행함



Block cipher

vs.

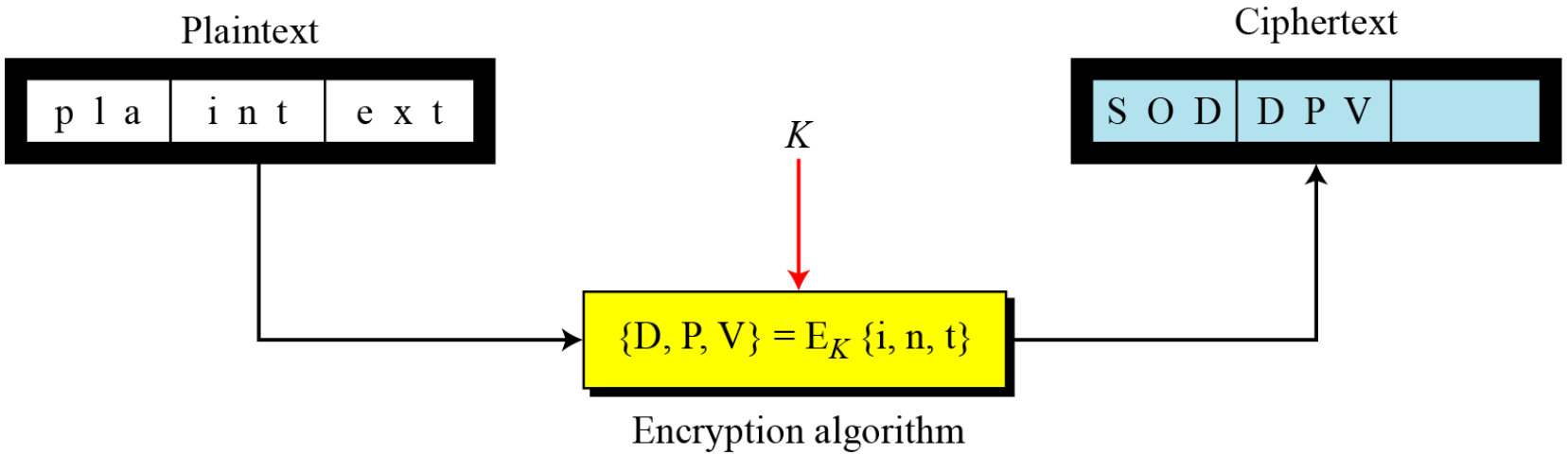


Stream cipher

대칭키 암호 시스템

❒ 블록 암호

- 실생활에서 많이 쓰이는 암호 시스템
- 평문/암호문을 정해진 크기의 블록으로 나누어 블록단위로 암호화/복호화를 진행함



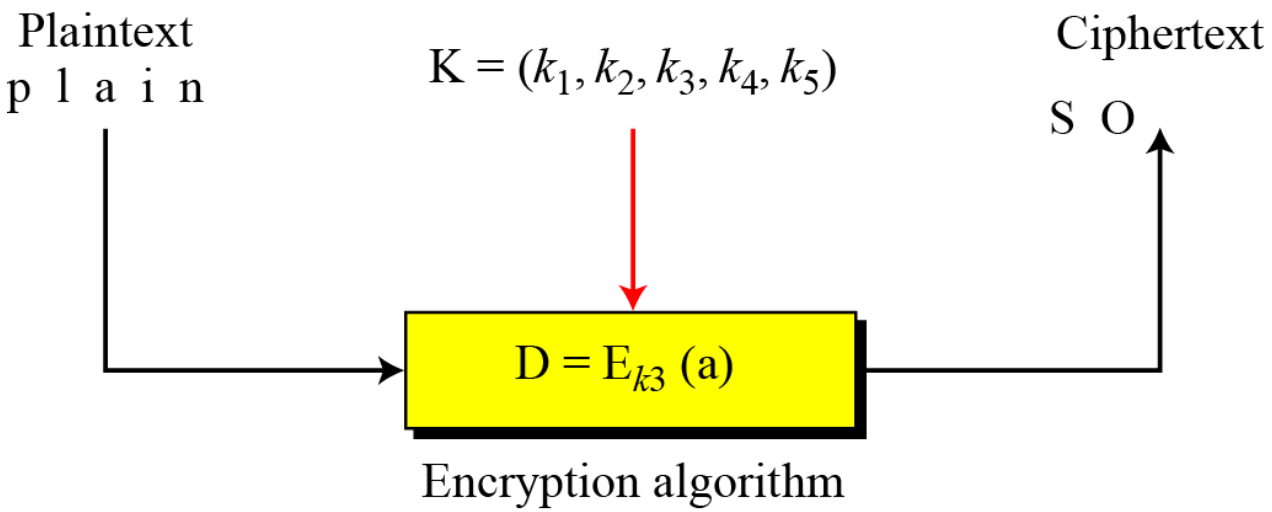
대칭키 암호 시스템

❒ 스트림 암호

- 평문/암호문을 1비트 단위로 암호화/복호화를 진행함

$$P = P_1P_2P_3, \ldots \qquad C = C_1C_2C_3, \ldots \qquad K = (k_1, k_2, k_3, \ldots)$$

$$C_1 = E_{k1}(P_1) \qquad C_2 = E_{k2}(P_2) \qquad C_3 = E_{k3}(P_3) \ldots$$



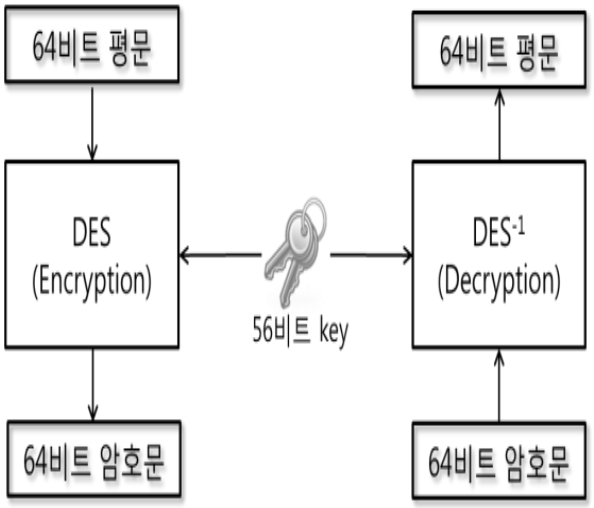
Data Encryption Standard(DES)

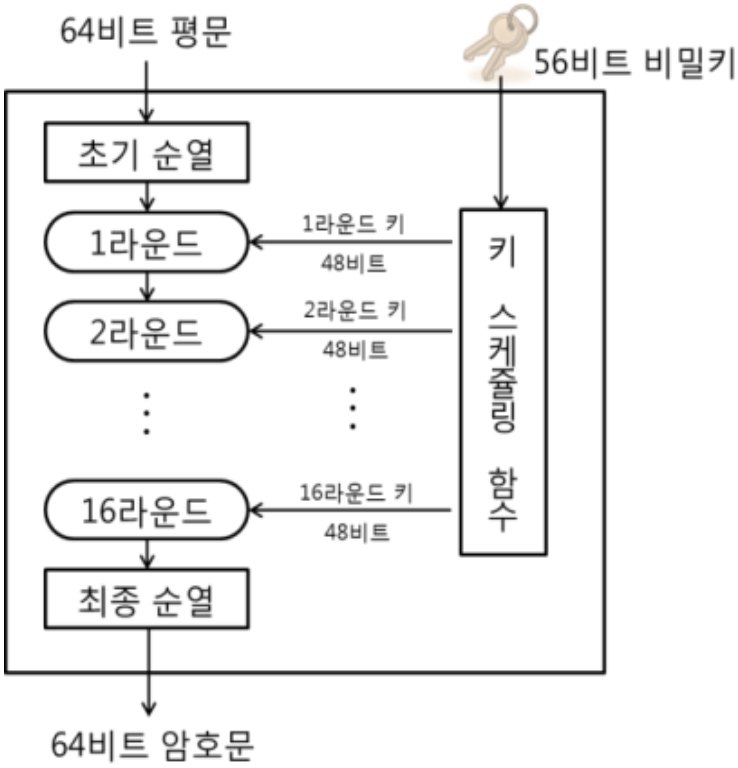
DES(Data Encryption Standard)

- 1973년 미국의 연방 표준국(National Bureau of Standards, NBS, 현재의) DES 공모
- IBM은 자사의 루시퍼(Lucifer)를 제출
- 미 연방 표준국은 1977년 루시퍼를 수정하여 DES로 선정
  - FIPS PUB 46
  - 국가안보국(National Security Agency, NSA)는 루시퍼에서 사용된 64비트 키를 56비트로 변경함
  - Most widely used block cipher in world
- 64-bit 데이터 블록, 56-bit 암호키
- 현재는 DES를 사용하지 않고 3-DES(triple DES)와 AES(Advanced Encryption Standard) 사용

Data Encryption Standard(DES)

- 64비트의 평문을 56비트의 키로 암호화하여 64비트의 암호문을 생성





Data Encryption Standard(DES)

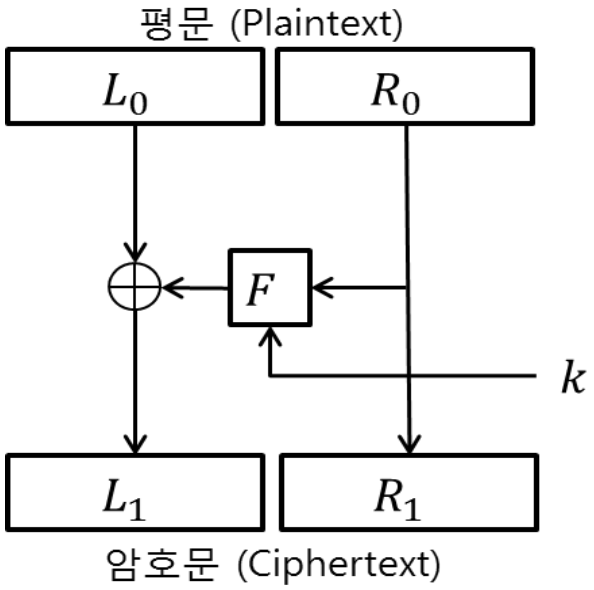
- 페이스텔(Feistel) 암호: 가역(Invertible)요소와 비가역(Non-Invertible) 요소 모두를 사용
  - 암호화와 복호화 과정이 동일
  - 참고
    - 비페이스텔(Non-Feistel) 암호: 가역 요소만 사용

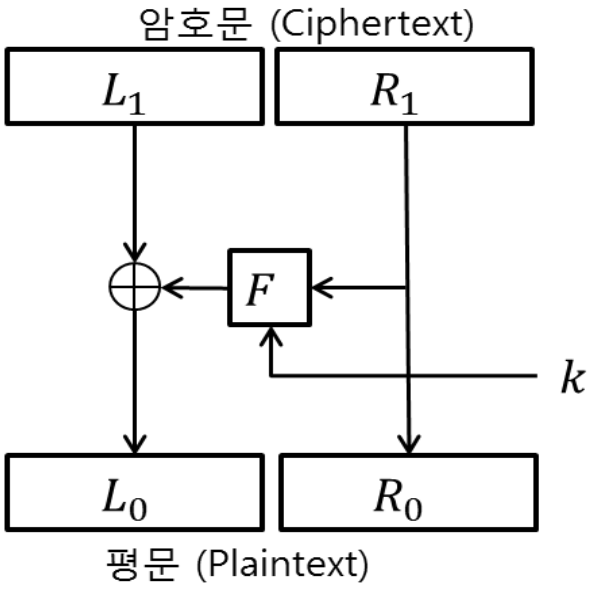
Note : 전단사 함수 $f: X \rightarrow Y$에 대하여 Y에서 X로의 역관계가 존재하면 이를 역함수(Inverse Function)라고 하며 $f^{-1}: Y \rightarrow X$로 나타낸다. 가역함수는 바로 역함수가 존재하는 전단사함수를 의미

Data Encryption Standard(DES)

1라운드 페이스텔(Feistel) 구조

- **암호화**
  - $L_1 = L_0 \oplus F(k, R_0);\ R_1 = R_0$
- **복호화**
  - $L_1 \oplus F(k, R_1) \ = L_1 \oplus F(k, R_0) = L_0 \oplus F(k, R_0) \oplus F(k, R_0) = L_0$
  - $R_0 = R_1$





Data Encryption Standard(DES)

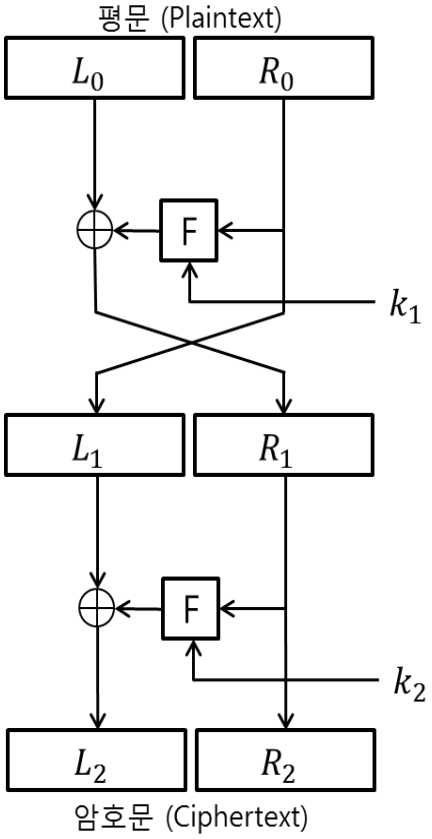
❒ [참고] 2 라운드 페이스텔(Feistel) 구조

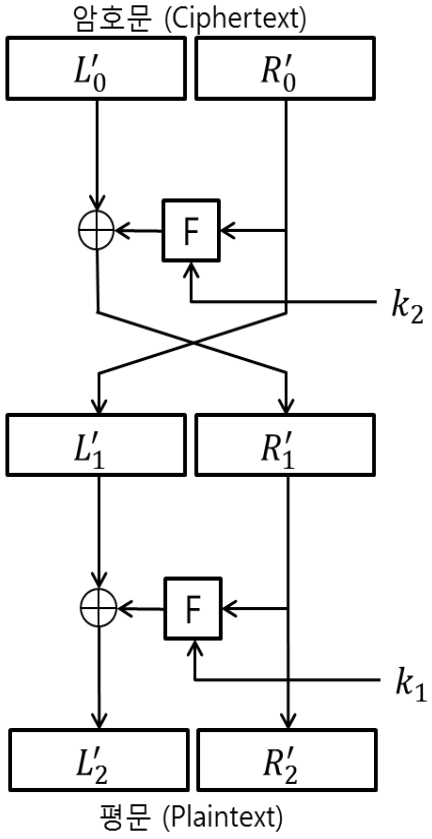
- **암호화**

$L_1 = R_0$
$R_1 = L_0 \oplus F(k_1, R_0)$
$L_2 = L_1 \oplus F(k_2, R_1)$
$R_2 = R_1$

- **복호화**

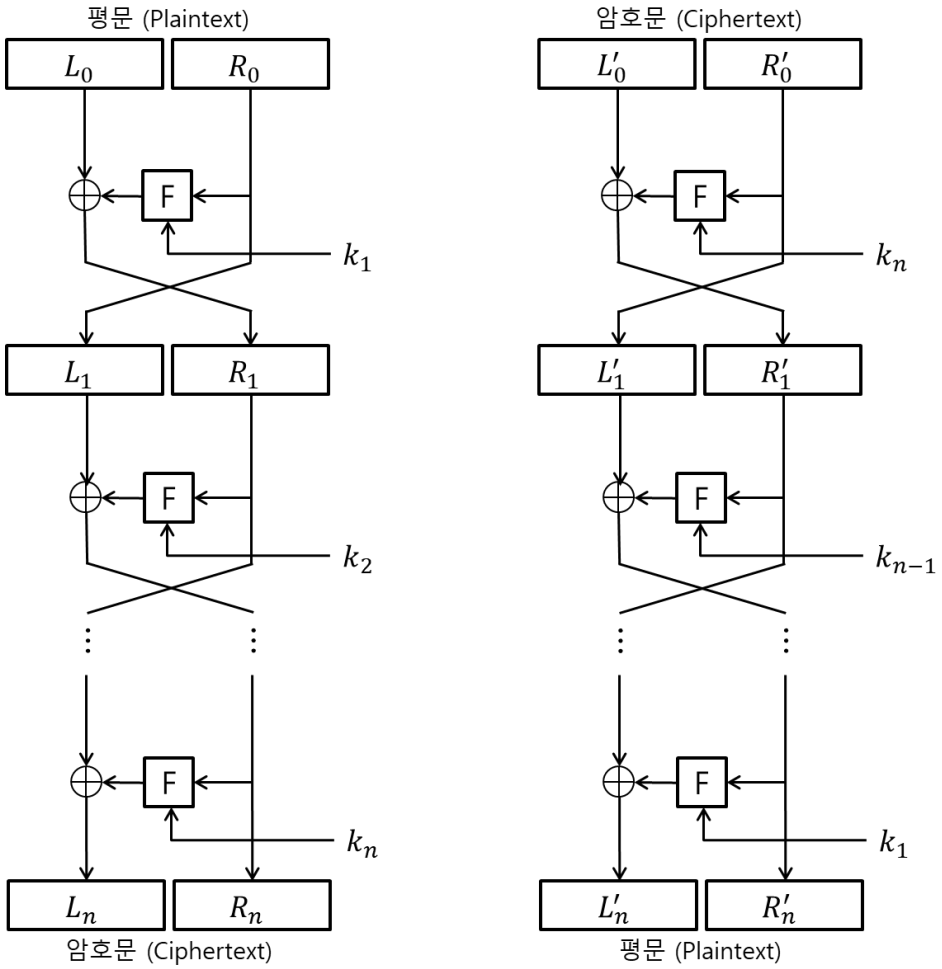
$L'_0 = L_2, \quad R'_0 = R_2$
$L'_1 = R'_0 = R_2$
$R'_1 = L'_0 \oplus F(k_2, R'_0)$
$= L_1 \oplus F(k_2, R_1) \oplus F(k_2, R_2)$
$= L_1 \oplus F(k_2, R_2) \oplus F(k_2, R_2)$
$= L_1$
$R'_2 = R'_1 = L_1 = R_0$
$L'_2 = L'_1 \oplus F(k_1, R'_1)$
$= R_2 \oplus F(k_1, L_1)$
$= L_0 \oplus F(k_1, R_0) \oplus F(k_1, R_0)$
$= L_0$





Data Encryption Standard(DES)

- 다중 라운드 페이스텔(Feistel) 구조



Data Encryption Standard(DES)

- DES에서는 함수 F를 Cryptanalysis 공격으로부터 안전해야 함

- 함수 F에 대한 상세한 설명은 Skip!
  - 함수 F를 자세히 이해하는 것보다, DES의 구조를 이해하는 것이 중요함
  - 함수 F에 대한 이해는 보안관련 대학원의 교육과정(보안전공 대학원생)에 적합함
  - 관심있는 학생들은 교재 참고

Data Encryption Standard(DES)

- **Brute Force Attack on DES**
  - 1981: estimated breakable in 2 days by $50M machine
  - DES Challenge I(1997): broken in 96 days by 70,000 machines, testing 7 billion keys/s (DESCHALL project)
  - DES Challenge II-1(1998): broken by distributed.net in 41 days
  - DES Challenge II-2(1998): less than 56 hours by special hardware, $250K incl design and development (“Deep Crack”)
  - DES Challenge III(1999): 22 h 15 min, “Deep Crack” + 100 000 machines, testing 245 billion keys/s
  - 2007: 6.4 days, $10K hardware, 120 FPGAs (COPACOBANA project)

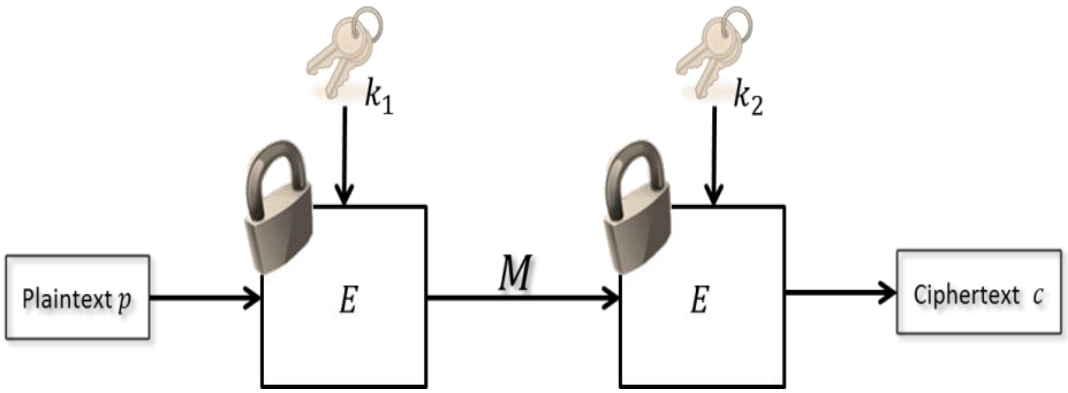
- **그 외 다양한 cryptanalysis 분석도 DES가 안전하지 않음을 보임**

Data Encryption Standard(DES)

❒ 다중 DES

- 56비트인 DES암호의 짧은 키 길이를 보완하기 위해 DES를 여러 번 사용하는 다중 DES가 제안

❒ 2중 DES (Double DES)

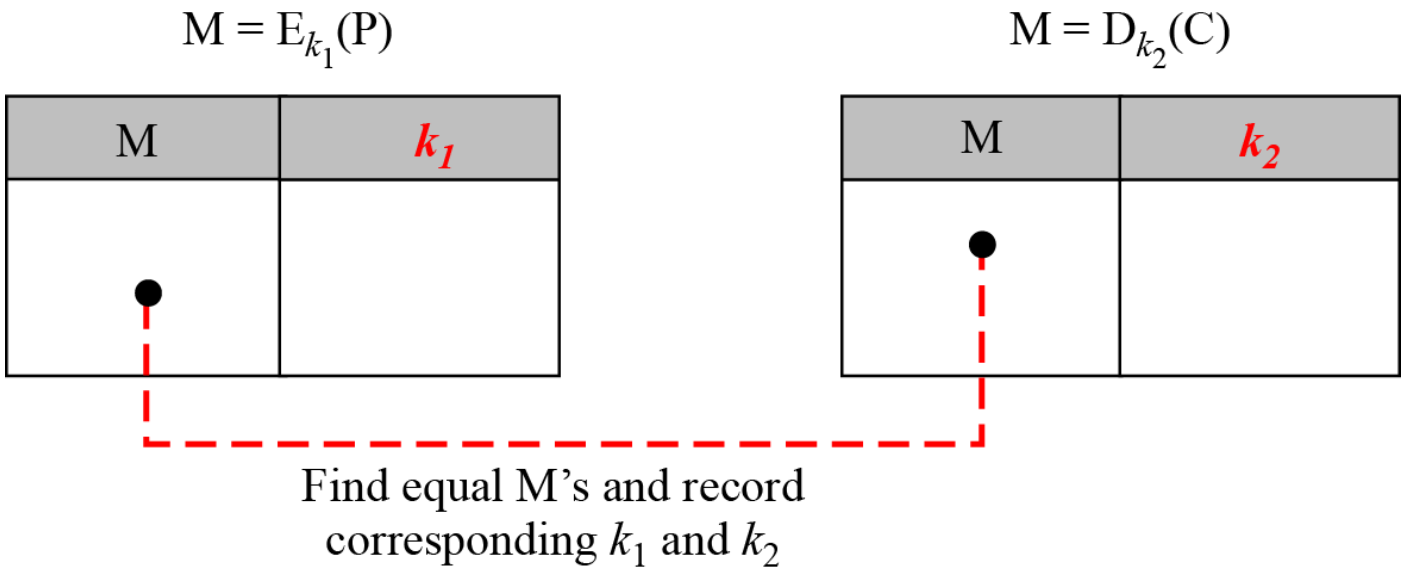


- 키의 길이가 56 비트 + 56 비트 = 112비트의 효과가 있을까?

Data Encryption Standard(DES)

❒ 2중 DES (Double DES)

- 중간 일치 공격(Meet-in-the-Middle Attack)
- $E_{k_1}(p) = m = D_{k_2}(c)$

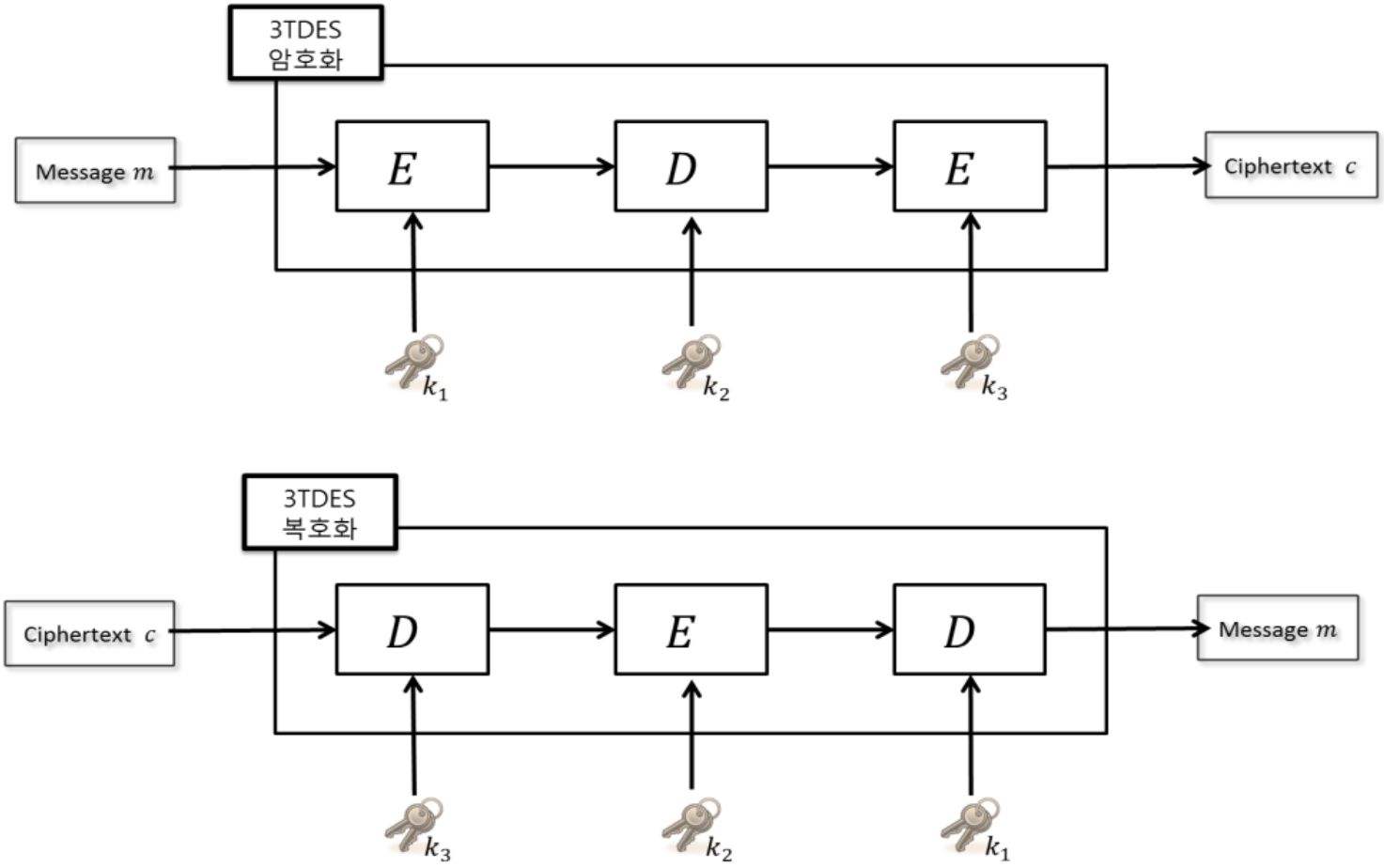


약 $2 \times 2^{56} = 2^{57}$번의 계산이 필요하다. 두 표에서 일치하는 값을 찾기 위하여 정렬하고 검색하는 연산까지를 포함하면 **약 2^{63}번의 계산이 필요**

Data Encryption Standard(DES)

❒ 3중 DES (Triple DES)

- National Institute of Standards and Technology (NIST) 표준임



Advanced Encryption Standard (AES)

- DES의 2^{56}개의 키에 대한 전사적 공격이 가능
  - 1999년 distributed.net 과 Electronic Frontier Foundation이 협력한 공격에서 DES의 비밀키를 22시간 15분만에 찾아냄
- TDES가 있지만 다음 이유로 NIST에서는 AES 공모
  - TDES는 DES를 세 번 사용하기 때문에 속도가 느림
  - DES의 블록 크기인 64 비트는 여러 가지 응용분야에 적합하지 않음
    - 예로 블록 암호를 이용하여 설계한 인증코드 (Message Authentication Code)경우 64비트의 블록 크기는 해쉬 함수의 안전성에 문제
  - 가까운 미래에 양자컴퓨터가 현실화 될 수 있으며, 양자컴퓨터를 이용하여 공격할 경우 적어도 256 비트 크기의 키가 바람직함

Advanced Encryption Standard (AES)

History

- US NIST issued call for ciphers in 1997
  - 15 candidates accepted in Jun 98
  - 5 were shortlisted in Aug-99
  - Rijndael was selected as the AES in Oct-2000
  - Rijndael issued as FIPS PUB 197 standard in Nov-2001
- AES의 공모 시 요구사항
  - 블록의 크기는 128 비트
  - 대칭키 암호이며 세 종류의 키(128 비트, 192 비트, 256 비트)를 사용할 수 있어야 함
  - 소프트웨어와 하드웨어로 구현될 경우 모두 효율적
  - 모든 키를 다 찾는 전수 키 조사 이외에 현재 알려진 다른 암호 분석 공격에 강해야 함

Advanced Encryption Standard (AES)

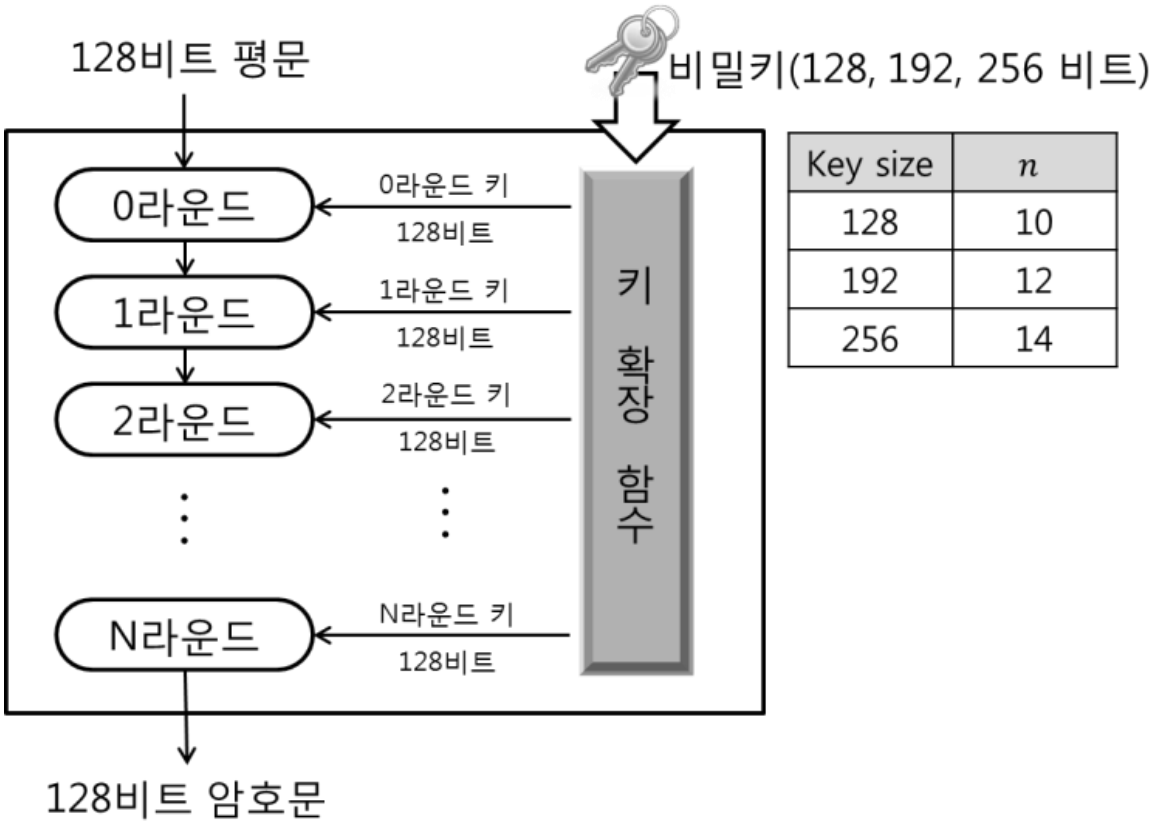
- 당시 벨기에 루벤대학의 대학원생인 Rijmen과 Daemen이 설계 AES 공모의 모든 요구사항을 만족시킴

  - 128/192/256 bit keys, 128 bit data
  - an iterative rather than feistel cipher
  - 설계:
    - 하드웨어나 소프트웨어로 구현할 때 속도나 코드 간결성(Compactness) 면에서 효율적
    - 알려진 블록 암호 알고리즘에 대한 공격들에 안전
    - 현재 AES에 대한 가장 실질적인 공격은 전수 키 조사
    - 최악의 경우(in the worst case) 2^{128} 번의 계산이 필요 (이러한 계산량은 현재 가장 빠른 슈퍼컴퓨터가 계산을 수행해도 태양계의 수명보다 긴 시간이 필요)

Advanced Encryption Standard (AES)

❒ AES 구조

- 한 블록 : 128 비트
- 128, 192, 256비트의 비밀키에 대해 라운드의 수는 각각 10, 12, 14라운드가 실행

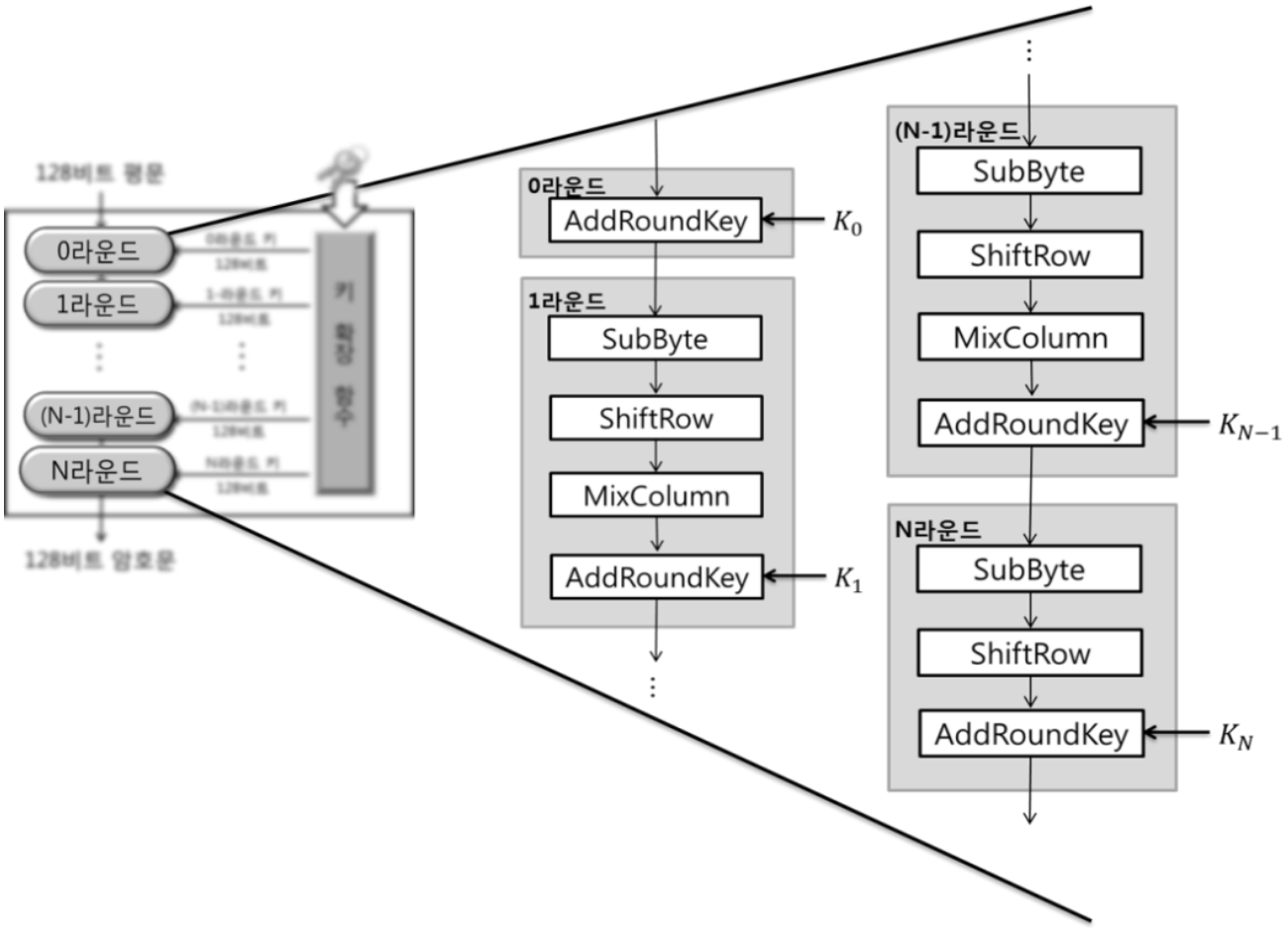


Key size	n
128	10
192	12
256	14

Advanced Encryption Standard (AES)

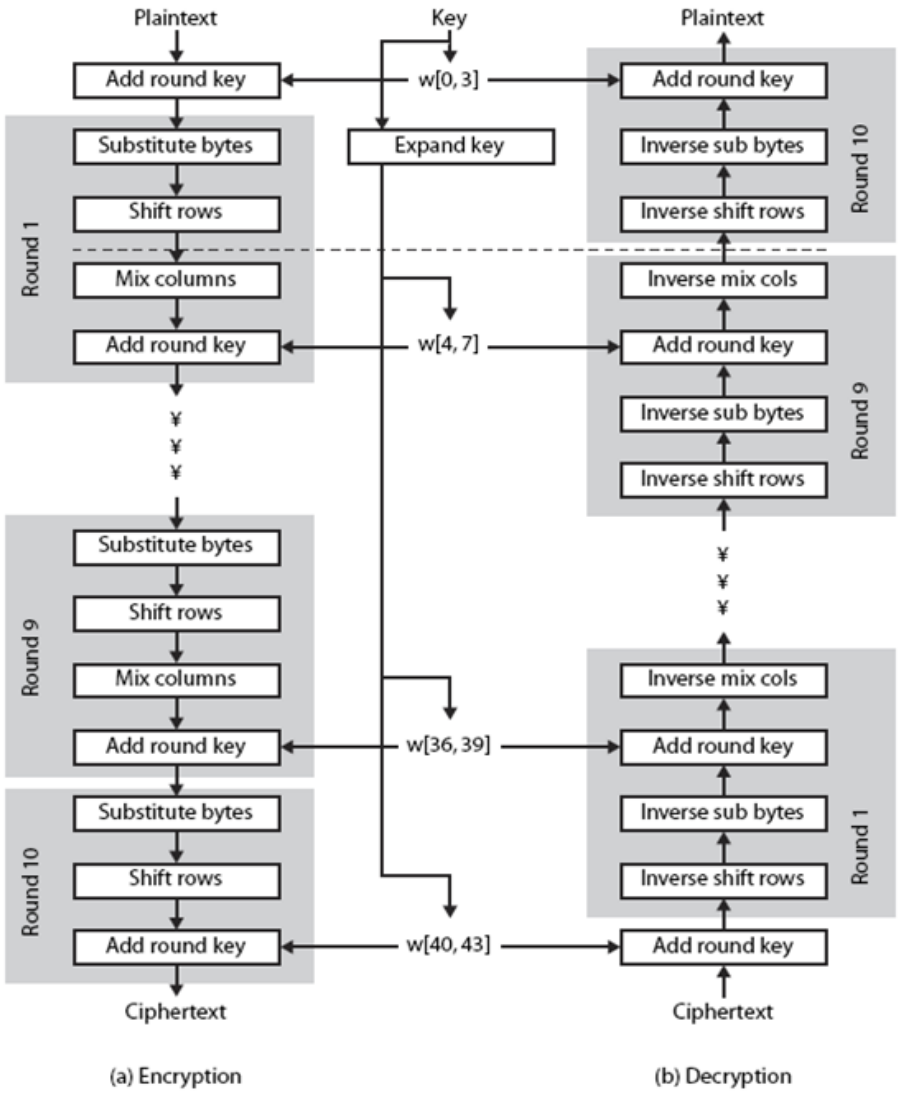
- 한 블록인 16 바이트(=128 비트)는 원소가 한 바이트인 4x4 행렬로 변환됨
  - 이 행렬을 상태(state)라 부름
- 한 라운드는 네 가지 계층(Layer)으로 구성
  - SubBytes : DES의 S-Box에 해당하며 한 바이트 단위로 치환을 수행.
    - 상태(state)의 한 바이트를 대응되는 S-Box의 한 바이트로 치환한다. 이 계층은 혼돈의 원리를 구현한다.
  - ShiftRows : 상태의 한 행안에서 바이트 단위로 자리바꿈이 수행
  - MixColumns : 상태가 한 열안에서 혼합이 수행. ShiftRows와 함께 분산의 원리를 구현
  - AddRoundKey : 비밀키(128/192/256 비트)에서 생성된 128 비트의 라운드 키와 상태가 XOR됨

Advanced Encryption Standard (AES)



Advanced Encryption Standard (AES)

- Encryption and Decryption
  - Iterative이기 때문에 모든 component가 inversable해야 함
  - Round key는 DES와 동일하게 역순임



Advanced Encryption Standard (AES)

- 블록이 상태(State)의 형태로 표현
  - 상태는 원소가 한 바이트인 "4×4"행렬
  - AES의 한 블록이 "EASYCRYPTOGRAPHY" 인 경우

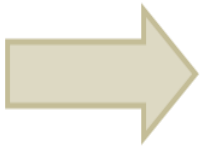
A	00	H	07	O	0E	V	15
B	01	I	08	P	0F	W	16
C	02	J	09	Q	10	X	17
D	03	K	0A	R	11	Y	18
E	04	L	0B	S	12	Z	19
F	05	M	0C	T	13		
G	06	N	0D	U	14		

16진수로 표현된 알파벳

16 바이트

E	A	S	Y	C	R	Y	P	T	O	G	R	A	P	H	Y
04	00	12	18	02	11	18	0F	13	0E	06	11	00	0F	07	18

(텍스트를 16진수로 표현)



상태(State) 4×4

04	02	13	00
00	11	0E	0F
12	18	06	07
18	0F	11	18

Advanced Encryption Standard (AES)

❒ Substitute Bytes(SubBytes) 계층

- 한 원소가 16진수로 (xy)인 경우 상위 4 비트 값인 x가 S-Box의 행을 결정하고 하위 4 비트 값인 y가 열을 결정

	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
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	CB	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

19	11	00	14
19	13	C6	12
04	04	00	23
00	12	0C	08

상태 (State)

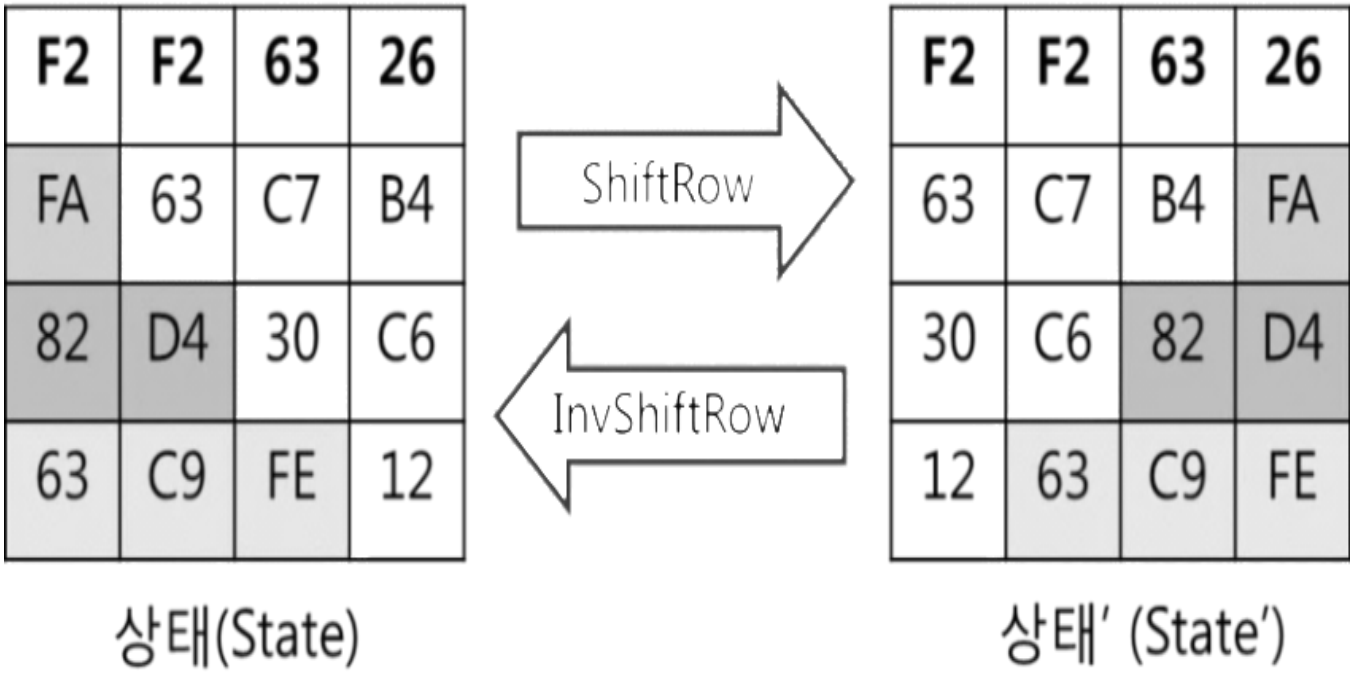
SubByte

D4	82	63	FA
D4	7D	B4	C9
F2	F2	63	26
63	C9	FE	30

상태' (State')

Advanced Encryption Standard (AES)

- ShiftRows 계층



Advanced Encryption Standard (AES)

MixColumns 계층

- SubBytes 계층과 ShiftRows 계층은 바이트 단위로 처리
- 충분한 분산 효과를 발생시키기 위하여, MixColumns 계층에서는 상태의 각 열을 비트 단위로 섞어 줌

$$\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

$$s'_{0,0} = (02 \cdot s_{0,0}) \oplus (03 \cdot s_{1,0}) \oplus (01 \cdot s_{2,0}) \oplus (03 \cdot s_{3,0})$$
$$s'_{1,0} = (01 \cdot s_{0,0}) \oplus (02 \cdot s_{1,0}) \oplus (03 \cdot s_{2,0}) \oplus (01 \cdot s_{3,0})$$
$$\vdots$$
$$s'_{3,3} = (03 \cdot s_{0,3}) \oplus (01 \cdot s_{1,3}) \oplus (01 \cdot s_{2,3}) \oplus (02 \cdot s_{3,3})$$

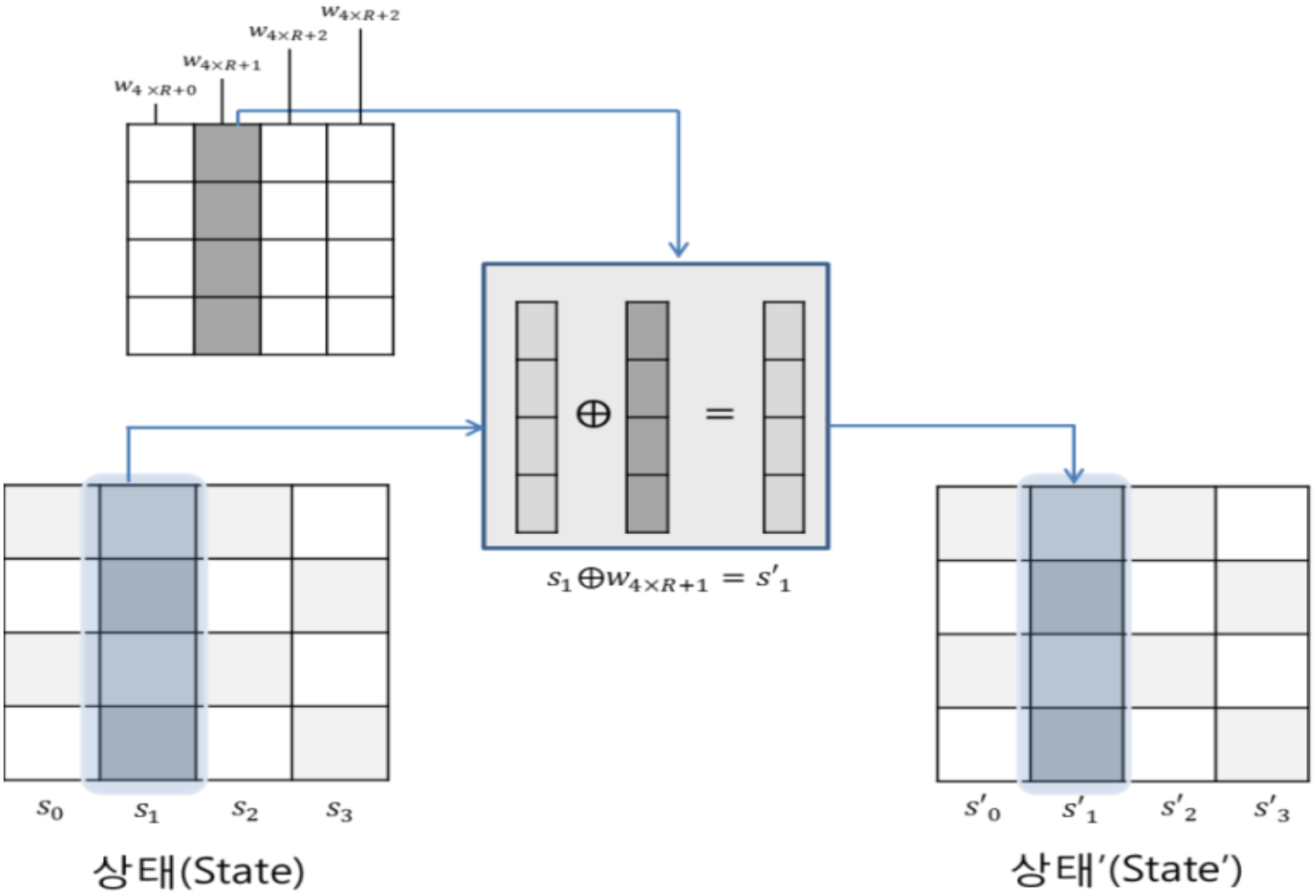
Advanced Encryption Standard (AES)

- Inverse MixColumns 계층

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}^{-1} = \begin{bmatrix} 0\mathrm{E} & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}$$

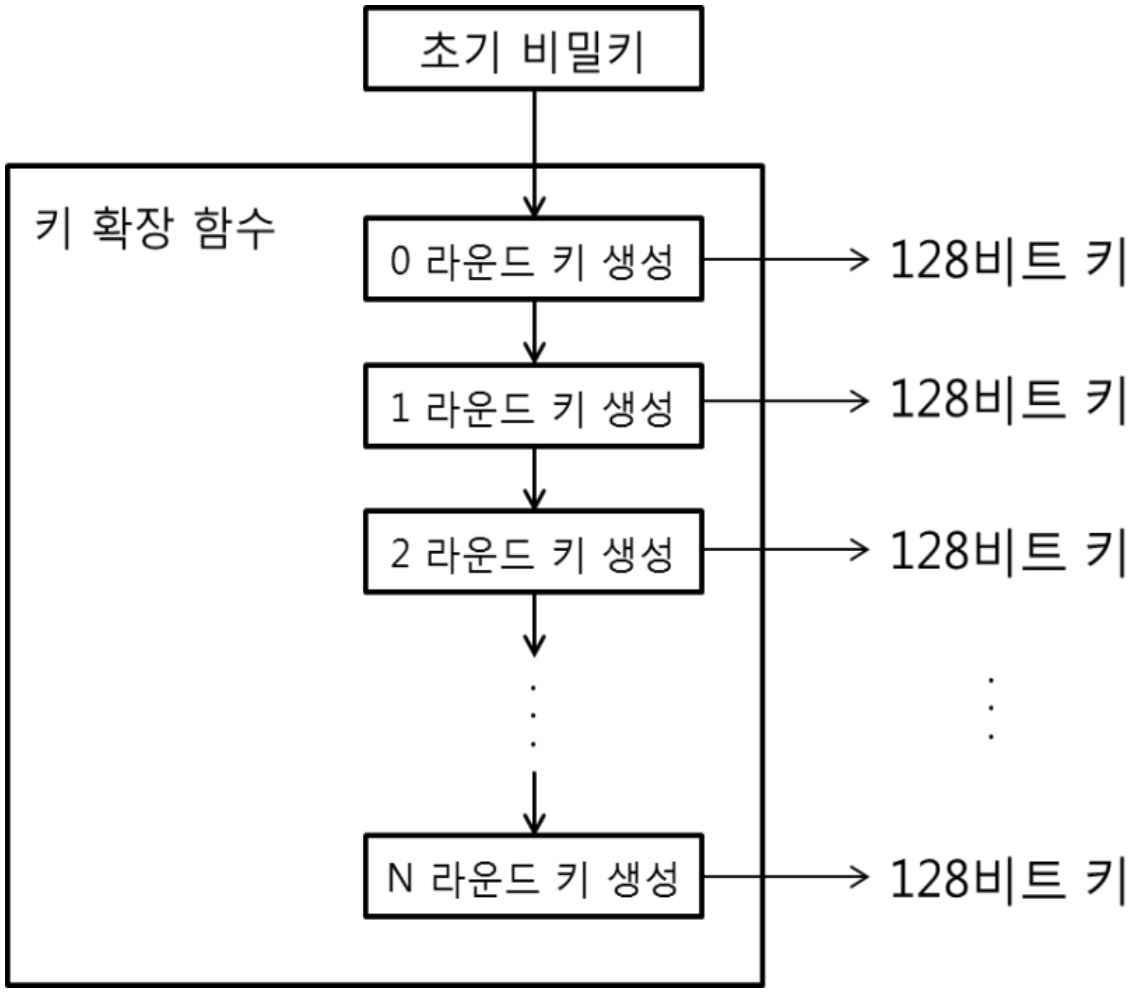
Advanced Encryption Standard (AES)

- AddRoundKey 계층



Advanced Encryption Standard (AES)

❐ 키 확장(Key Expansion)



Advanced Encryption Standard (AES)

- AES 과정
  - Animation of AES

Advanced Encryption Standard (AES)

❒ 안전성

- 취약키(Weak Keys)와 차분 분석 방법(Differential Cryptanalysis), 선형 분석 방법(Linear Cryptanalysis) 등을 이용한 공격에 대해 안전
- 2011년 “Biclique 암호분석”
  - 소요시간: 2^{126}
  - AES에 대한 가장 최선의 공격이라고 믿었던 전사적 공격(2^{128}의 연산이 필요)보다 4배정도 효율적인 공격
  - 이는 키 길이가 56비트인 DES 암호 알고리즘에 대한 전수 조사 공격을 2^{70}번을 실시 하는 것과 동일
  - Bruce Shneier: “공격 수법은 언제나 진화한다” "
    - “현재 AES을 대체할 다른 암호가 필요한 것은 아니며 향후 새로운 공격에 대비하여 AES의 라운드 수를 증가시켜야 한다고 주장"

HW #3

- 대칭키 암호에 대해 설명하시오.
- 2라운드의 DES의 암/복호화 과정을 설명하시오.
- Double DES가 안전하지 않은 이유에 대해 설명하시오.
- 3페이지 이내 (A4, 10 pt)
  - 6월 26일 정오 12시까지
  - 늦은 제출 시, 감점
  - 과제 copy 시, 관련된 과제들 모두 0점 처리
  - 제출양식: hwp or pdf

스트림 암호

❒ 블록 암호 vs. 스트림 암호

- 블록 단위로 암호화 vs. 비트 단위로 암호화

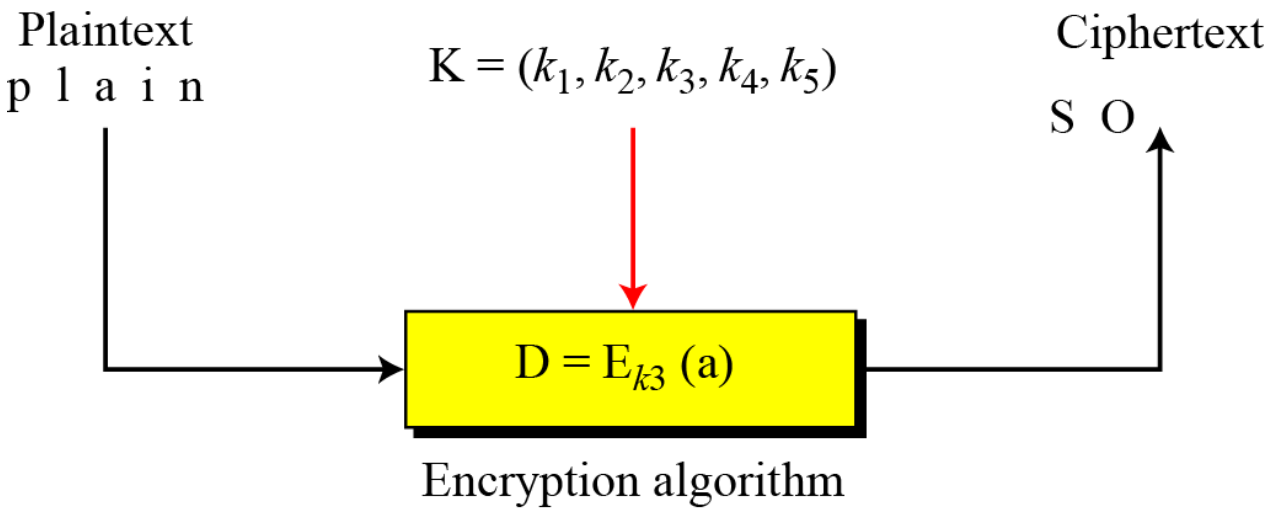
❒ 스트림 암호

- 패딩과 운영모드에 대한 개념 X (패딩과 운영모드는 다음 챕터에서 배움)
- 빠른 암/복호화가 가능
- 효율적인 “PRESENT” 나 “HIGHT” 와 같은 블록 암호가 존재

스트림 암호

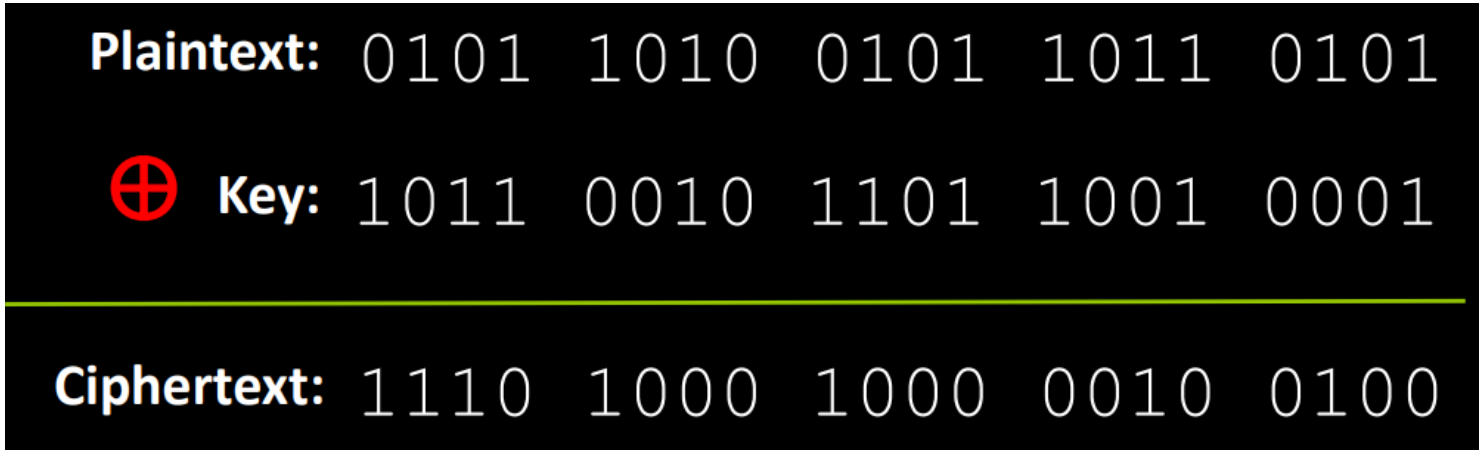
$P = P_1P_2P_3, \ldots$ $C = C_1C_2C_3, \ldots$ $K = (k_1, k_2, k_3, \ldots)$

$C_1 = E_{k1}(P_1)$ $C_2 = E_{k2}(P_2)$ $C_3 = E_{k3}(P_3) \ldots$



스트림 암호: One-time pad

- One-time pad
  - (Ideal) one-time pad is considered to be secure!

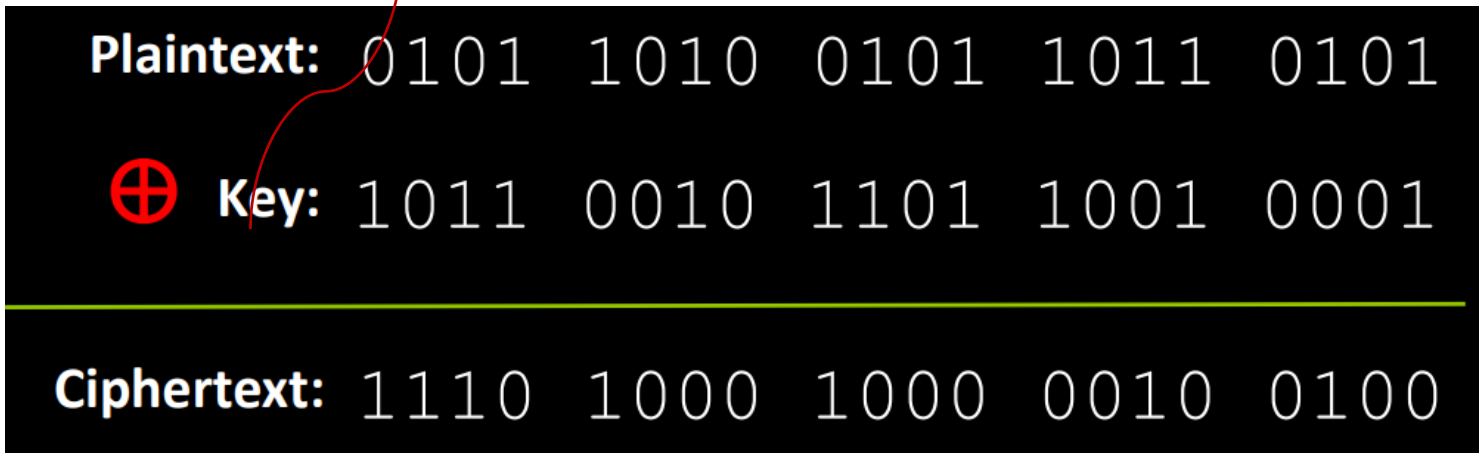


What is the problem?

스트림 암호: One-time pad

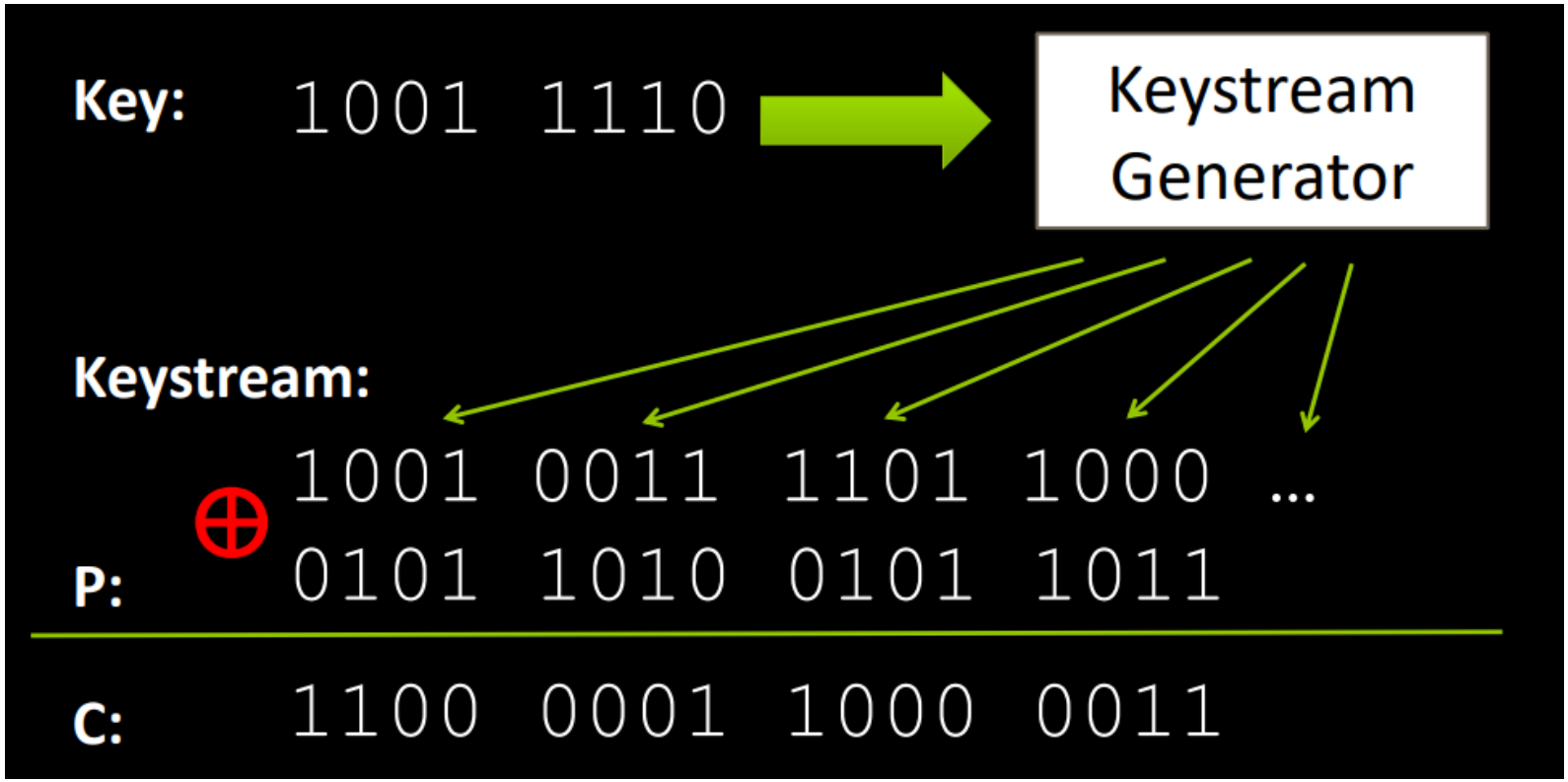
- One-time pad

Key is as long as the original message



스트림 암호: One-time pad

- One-time pad

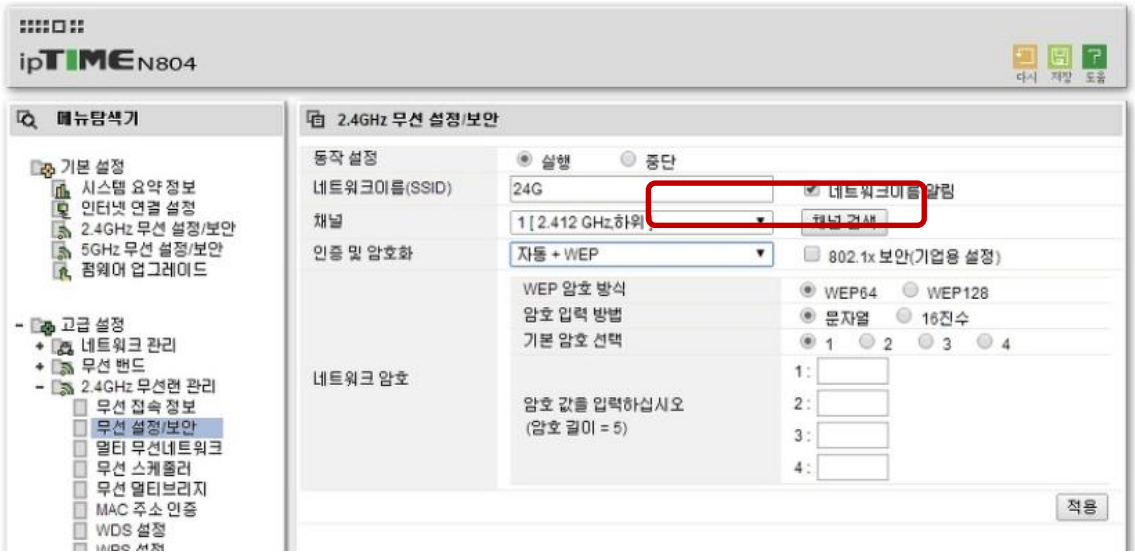


스트림 암호: One-time pad

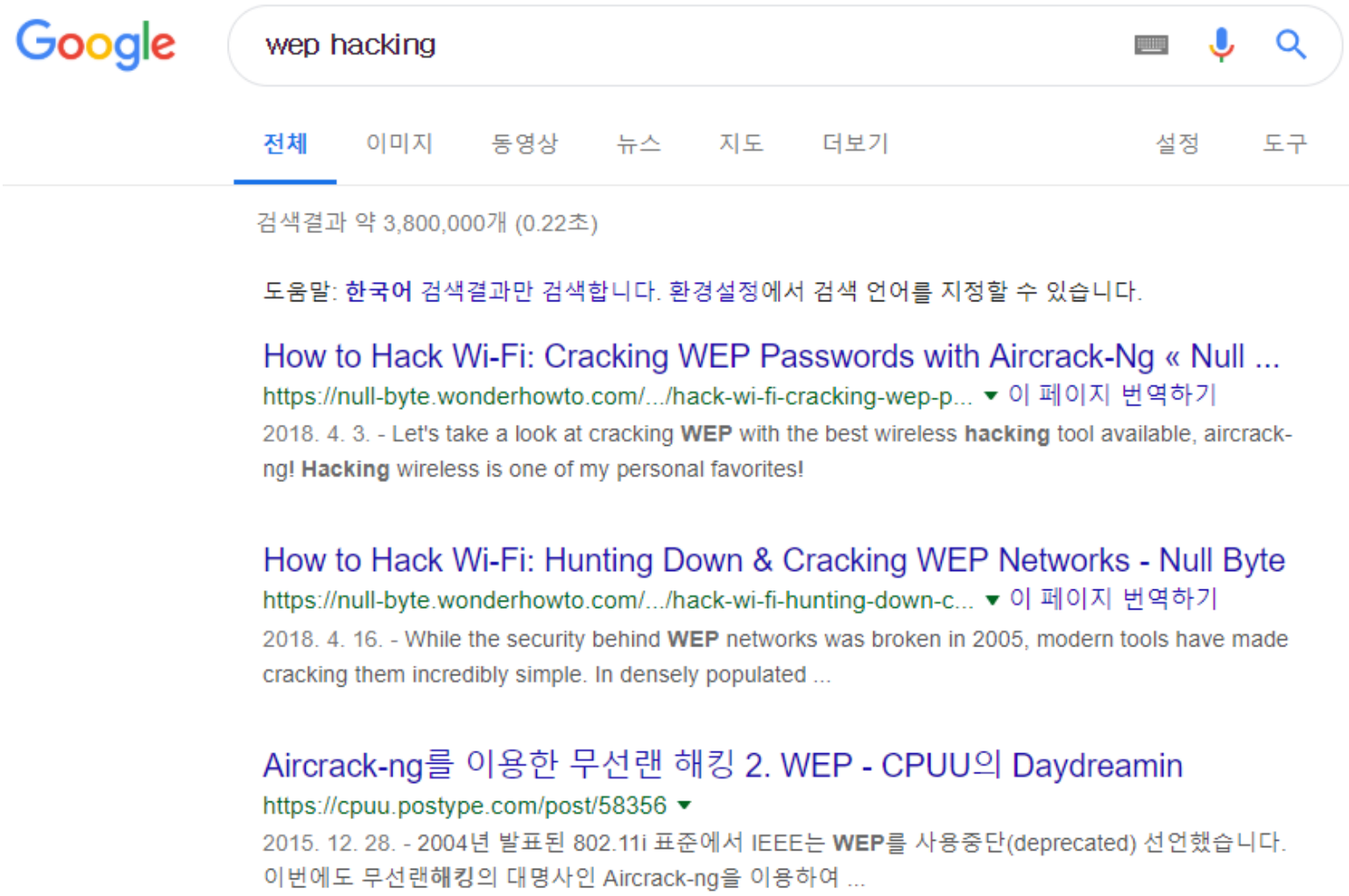
- One-time pad의 안전성은 Key generator의 랜덤성에 의존함
  - The keystream should approximate the properties of a true random number stream as close as possible
  - The more random-appearing the keystream is, the more randomized the ciphertext is, making cryptanalysis more difficult

스트림 암호: Rivest Cipher 4 (RC4)

- RC4는 스트림 암호중 하나로 wireless 통신에 많이 사용됨
  - WEP, WPA, etc.
- RC4는 소프트웨어 구현이 효율적으로 될 수 있도록 설계됨



스트림 암호: Rivest Cipher 4 (RC4)



스트림 암호: Death of Stream Ciphers?

- Popular in the past
  - Efficient in hardware – Speed was needed to keep up with voice, etc.
- Today, processors are fast
  - Software-based crypto is usually fast enough
- Future of stream ciphers?
  - Shamir declared "the death of stream ciphers"
  - May be greatly exaggerated…

스트림 암호: eSTREAM

❐ eSTREAM (유럽 연합의 eSTREAM 공모사업)

- 2008년 4월 총 3단계에 걸친 심사를 통하여 최종 스트림 암호가 당선
- Profile1 : 높은 성능을 요구하는 소프트웨어 애플리케이션을 위한 스트림 암호
- Profile2 : 제한된 자원(저장공간,게이트의 수, 전력량)을 가진 하드웨어 애플리케이션을 위한 스트림 암호

Profile 1 (software)	Profile 2 (hardware)
HC-128 [1]	Grain [2]
Rabbit [3]	MICKEY [4]
Salsa20/12 [5]	Trivium [6]
SOSEMANUK [7]	

The project was completed in April 2008.

https://en.wikipedia.org/wiki/ESTREAM

Thank you