

4장 : 운영모드 (Mode of Operation)

정보보호이론

Spring 2015

4.0 운영모드

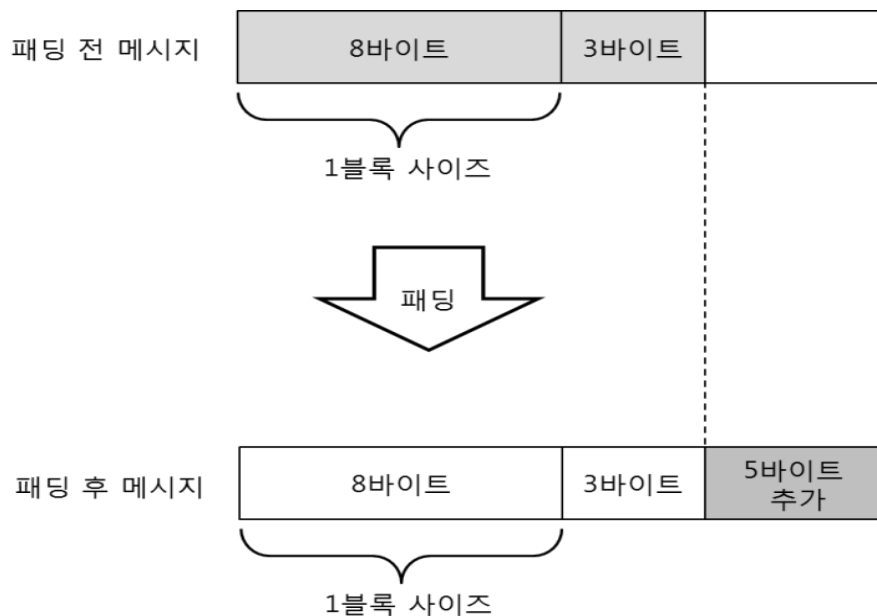
- 운영모드(mode of operation) : DES나 AES와 같은 블록 암호를 사용하여 다양한 크기의 데이터를 암호화하는 방식
 - ✕ 실제로 사용되는 평문은 다양한 크기를 가지며 보통 블록 크기보다 훨씬 큰 데이터
 - ✕ ECB, CBC, CFB, OFB, CTR

Test Schedule

- Quiz 1 : 2015년 3월 31일
- Mid-Term : 2015년 4월 14일

4.1 패딩(Padding)

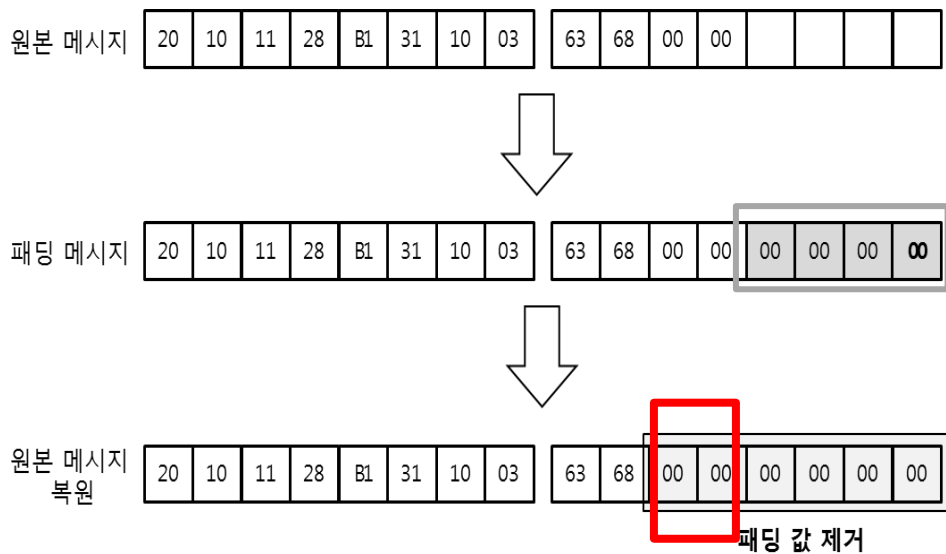
- 블록 Cipher의 경우, 평문의 길이가 정확하게 해당 블록 암호의 블록 크기의 배수가 되어야 함
 - ✕ 패딩은 평문의 전체가 블록 크기의 배수가 되도록 마지막 부분의 빈 공간을 채워 하나의 완전한 블록으로 만드는 작업



4.1 패딩(Padding)

■ 제로 패딩(Zero padding, Null padding)

... | 31 AB 34 FE 52 5E 97 12 | 3A FE 5A 00 00 00 00 00 |₍₁₆₎

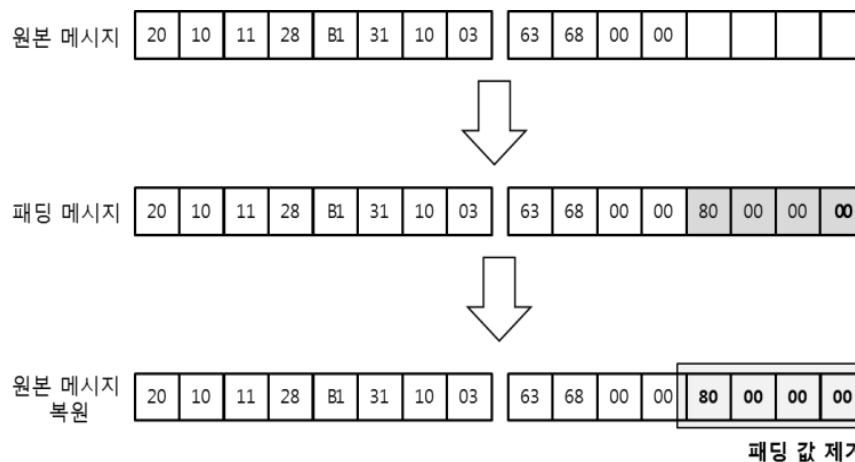


4.1 패딩(Padding)

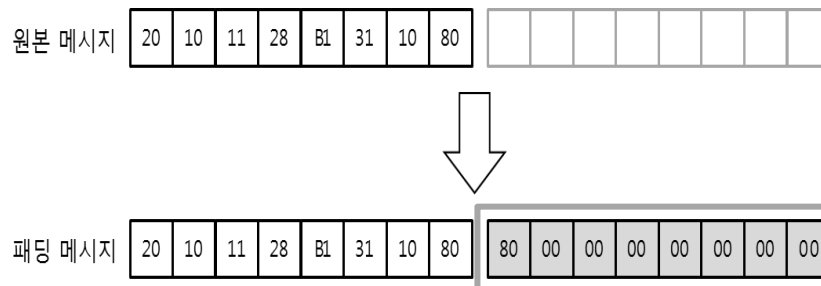
■ 비트 패딩(Bit padding)

... | 1001 1101 0011 1110 | 1101 1001 **1000 0000** | (2)

→ 최상위 비트



✕ 메시지 길이가 블록 크기의 배수일 때 비트 패딩

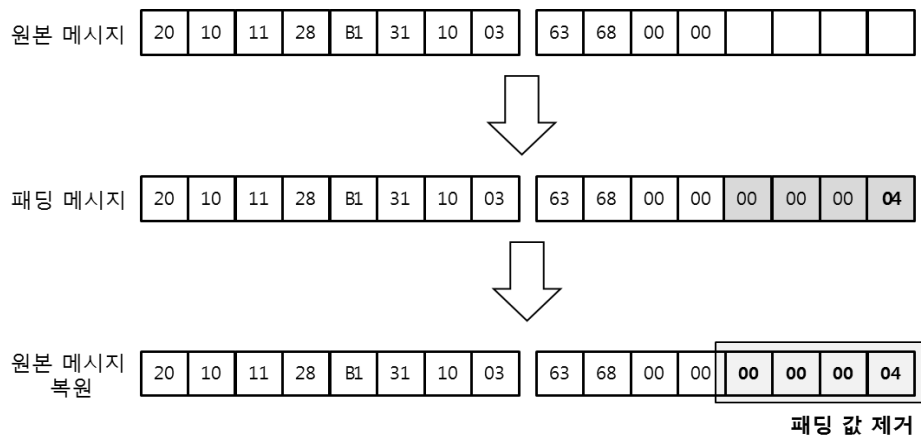


4.1 패딩(Padding)

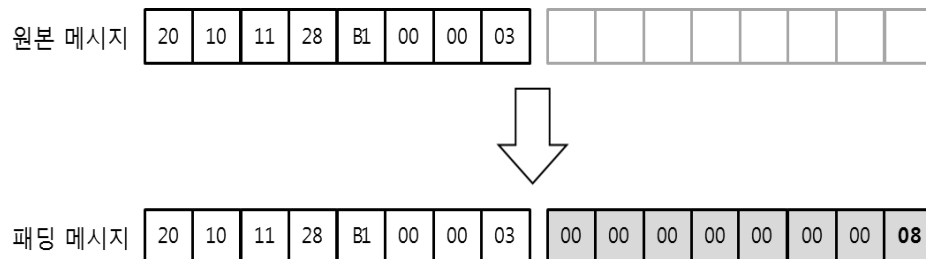
■ 바이트 패딩(Byte padding)

... | 31 AB 34 FE 52 5E 97 12 | 3A FE 5A 00 00 00 00 05 | (16)

최하위 바이트에 패딩
바이트 표시



✕ 메시지 길이가 블록 크기의 배수일 때 바이트 패딩



4.1 패딩(Padding)

■ PKCS7 패딩

✕ 패딩 바이트 값을 패딩 바이트 크기로 사용

원본 메시지

23	AF	4E	30	50	AF	4E	30
AB	3E	7F	97	AB	3E	84	97
64	64	90	5E	64	64		
6F	26	8A	6F				



패딩 메시지

23	AF	4E	30	50	AF	4E	30
AB	3E	7F	97	AB	3E	84	97
64	64	90	5E	64	64	06	06
6F	26	8A	6F	06	06	06	06

원본 메시지

23	AF	4E	30	50	AF	4E	30
AB	3E	7F	97	AB	3E	84	97
64	64	90	5E	64	64	98	6F
6F	26	8A	6F	3E	AC	68	20



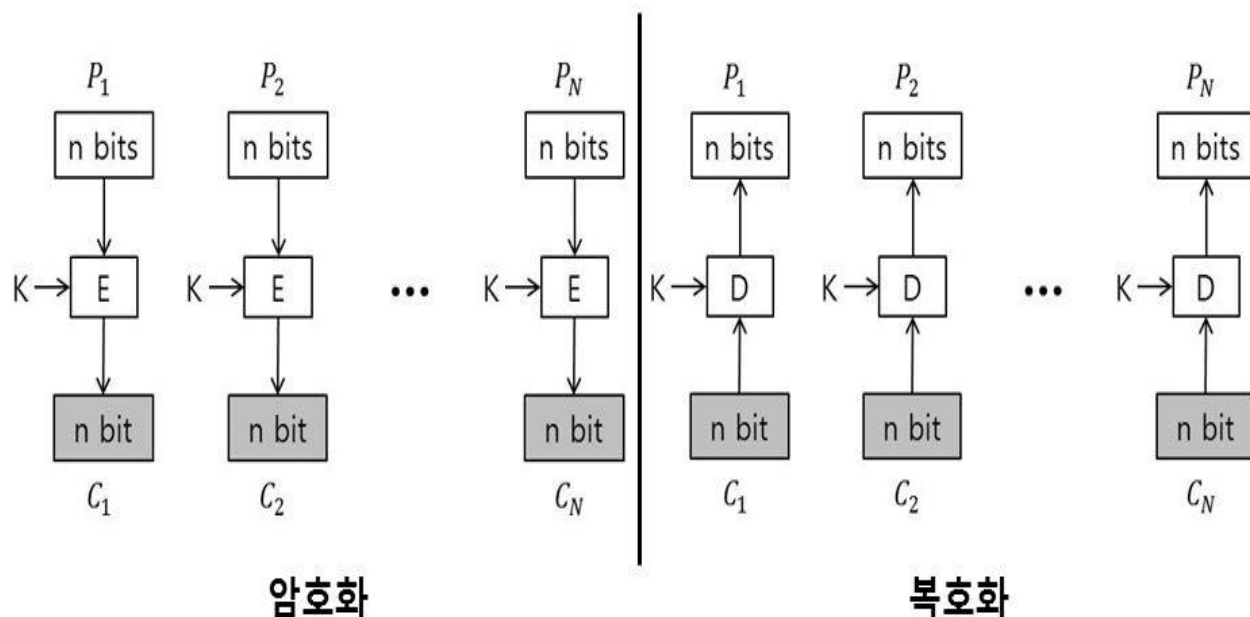
패딩 메시지

23	AF	4E	30	50	AF	4E	30	10	10	10	10
AB	3E	7F	97	AB	3E	84	97	10	10	10	10
64	64	90	5E	64	64	98	6F	10	10	10	10
6F	26	8A	6F	3E	AC	68	20	10	10	10	10

4.2 ECB 모드

- 한 블록의 평문은 한 블록의 암호문으로 암호화된다

✕ 암호화 : $C_i = E_K(P_i)$ 복호화 : $P_i = D_K(C_i)$



4.2 ECB 모드

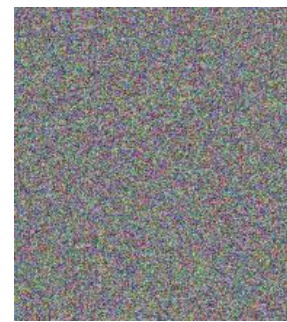
- 장점 : 병렬 처리가 가능 & 오류확산 x
- 단점 : 같은 평문에 대해 같은 암호문
 1. 블록 단위의 패턴 유지



Original



*Encrypted using ECB
mode*



*Encrypted using other
modes*


4.2 ECB 모드

- 단점 : 같은 평문에 대해 같은 암호문
2. 블록 재사용 (Block Replay)

이름	암호화된 점수 (원본 점수)
Alice	0F14D3F2 (90)
Bob	3DE9001F (80)
Eve	549F2D4F (50)



이름	암호화된 점수 (원본 점수)
Alice	0F14D3F2 (90)
Bob	3DE9001F (80)
Eve	0F14D3F2 (90)



4.2 ECB 모드

■ 암호문 스틸링 기법 (Ciphertext stealing)

$$\times X = E_K(P_{N-1})$$

→

$$C_N = head_m(X)$$

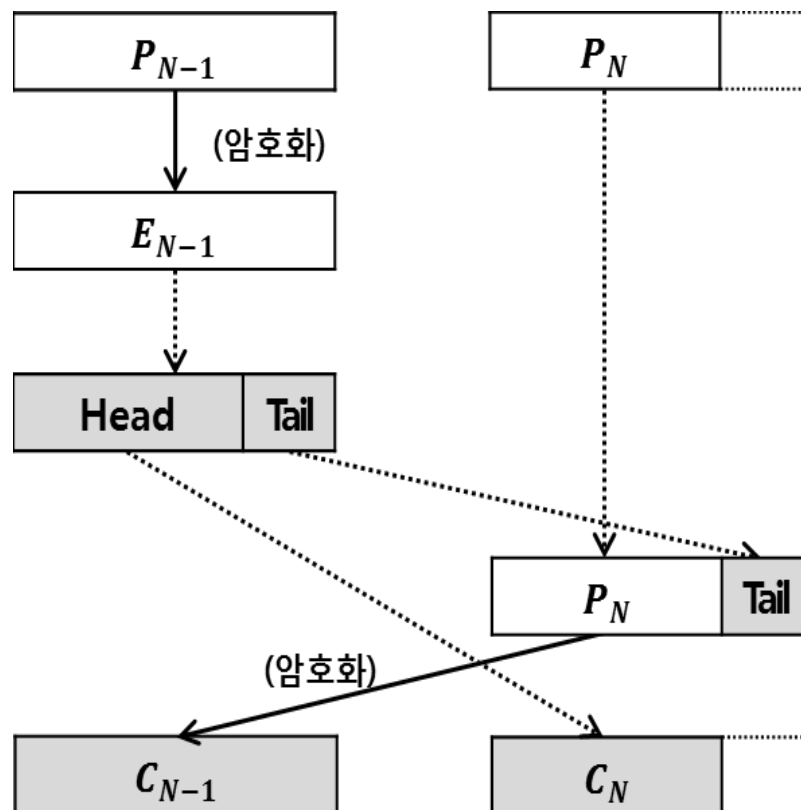
$$\times Y = P_N | tail_{n-m}(X)$$

→

$$C_{N-1} = E_K(Y)$$

$head_m$: 왼쪽 최상위 m 비트를
선택하는 함수

$tail_{n-m}$: 오른쪽 최상위
($n - m$)비트를 선택하는 함수

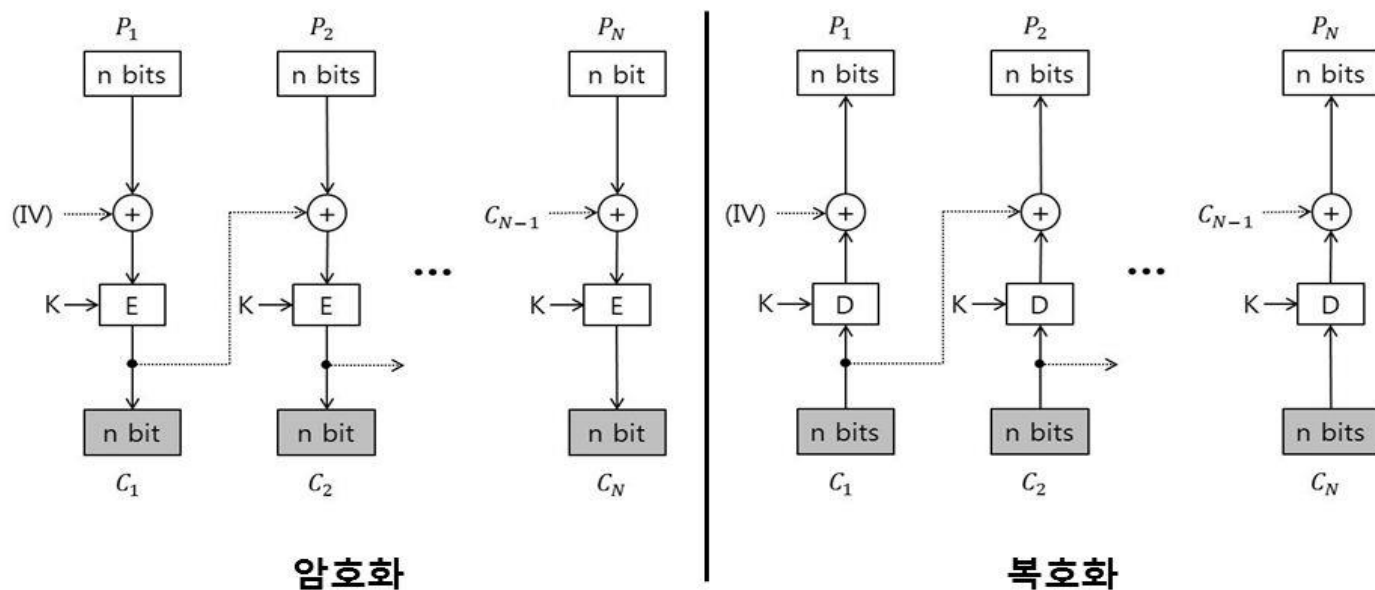


4.3 CBC (Cipher Block Chaining) 모드

- 한 평문 블록이 암호화 되기 이전에 바로 앞 평문 블록의 암호문과 XOR

✕ 암호화 : $C_0 = IV, \quad C_i = E_K(P_i \oplus C_{i-1}), i = 1, 2, 3, \dots, N$

✕ 복호화 : $C_0 = IV, \quad P_i = D_K(C_i) \oplus C_{i-1}, i = 1, 2, 3, \dots, N$



4.3 CBC (Cipher Block Chaining) 모드

■ 초기 벡터(IV, Initialization Vector)

- ✕ 평문을 암호화할 때마다 초기 벡터(IV)를 바꿈으로 임의화 (randomization) → 확률적 암호 알고리즘 (probabilistic encryption algorithm)
 - ▶ 동일한 평문이 암호화될 때 마다 통계적으로 독립된 서로 다른 암호문이 생성되는 성질
 - ▶ 현대 암호에서는 반드시 만족되어야 하는 성질
 - ▶ ECB 모드의 경우 동일한 평문에 대하여 동일한 암호문이 생성 → 결정적 암호 알고리즘(deterministic encryption algorithm)

4.3 CBC (Cipher Block Chaining) 모드

■ 초기 벡터(IV, Initialization Vector)

✕ Nonce 사용

- ▶ 난수를 만들어 송신자가 수신자에게 그대로 보내는 방법
- ▶ 서로 동기화된 카운터(counter)를 사용

✕ 안전성을 강화하기 위하여 논스를 ECB모드로 암호화하여 생성된 암호문을 IV로 사용할 수도 있다.

- ▶ ECB모드에 사용되는 키는 송신자와 수신자가 사전에 공유

✕ 실제 환경에서 IV의 비밀성이 아니라 무결성이 중요

- ▶ 만약 공격자가 전송되는 IV의 한 비트를 변경시킨다면 수신자는 제대로된 평문을 얻을 수 없기 때문

4.3 CBC (Cipher Block Chaining) 모드

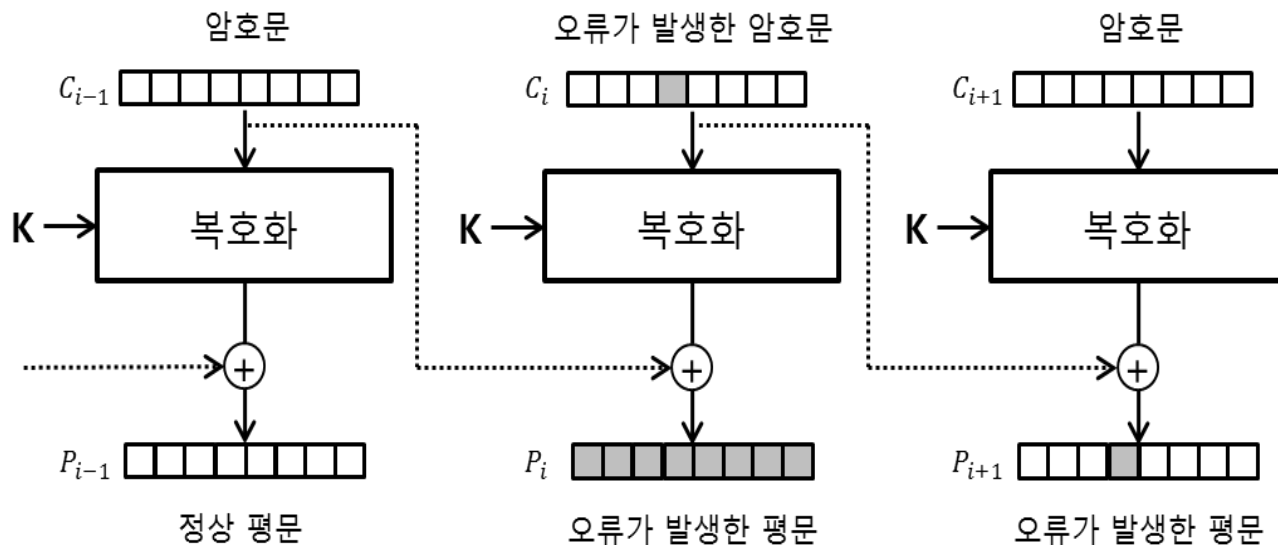
■ 연결성(chaining)

- ✕ 한 평문 안에 동일한 두 개의 블록에 대응되는 암호문 블록이 상이
- ✕ ECB모드에서 보이는 평문의 블록 패턴들이 CBC의 암호문에 서는 더 이상 보이지 않게 됨
- ✕ 블록단위의 재사용이 불가능

4.3 CBC (Cipher Block Chaining) 모드

■ 오류 확산(Error Propagation)

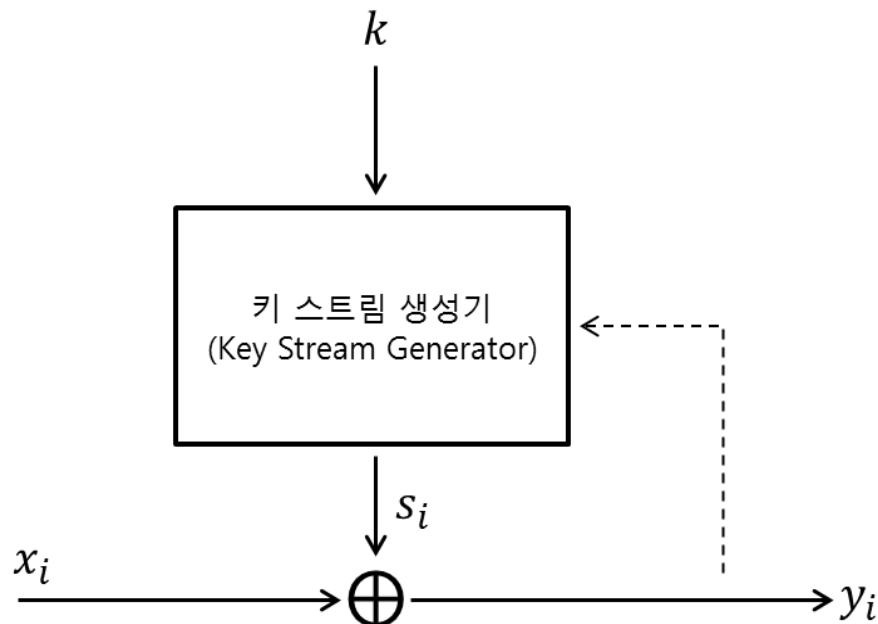
- ✕ stage i : $D_k(c_i) \oplus c_{i-1} = P_i$
- ✕ stage $(i+1)$: $D_k(c_{i+1}) \oplus c_i = P_{i+1}$
- ✕ after stage $(i+1)$, CBC is **self-recovering**.



4.4 CFB (Cipher Feedback) 모드

■ 스트림 암호 (5장)

- ✕ 암호화 : $c_i = E_{s_i}(p_i) = p_i \oplus s_i$
- ✕ 복호화 : $p_i = D_{s_i}(c_i) = c_i \oplus s_i$.



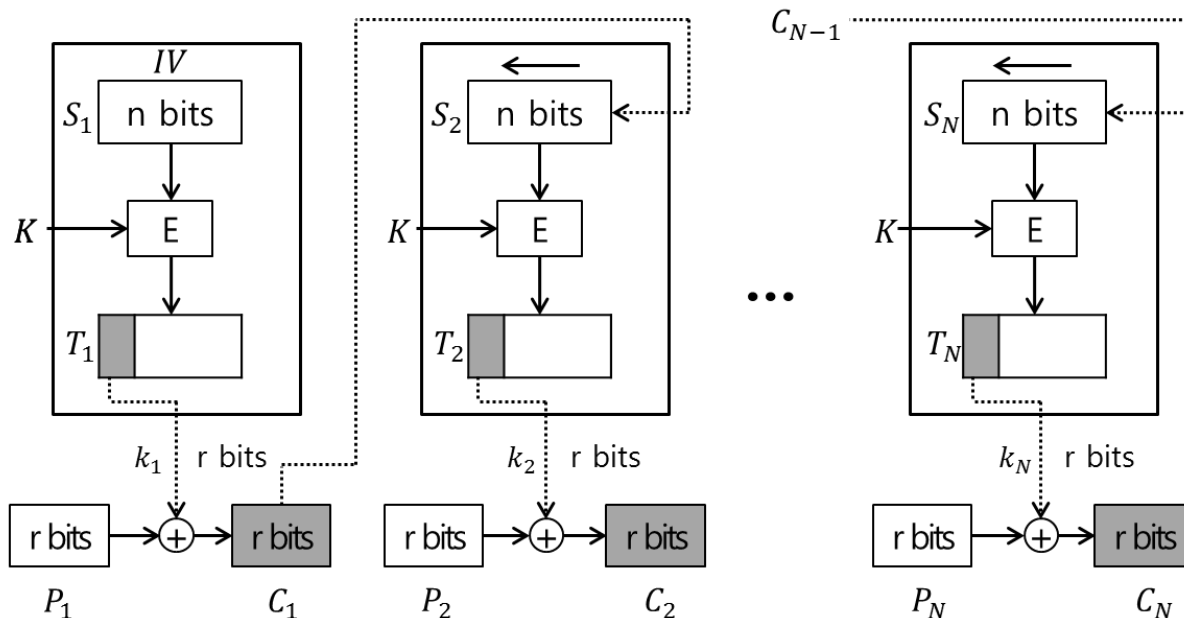
4.4 CFB (Cipher Feedback) 모드

- CFB, OFB, CTR 모드는 스트림 암호를 만들기 위해서
사용 : 블록 단위보다 작은 단위로 암호화를 진행

암호화 : $C_i = P_i \oplus \text{SelectLeft}_r\{E_K[\text{ShiftLeft}_r[(S_{i-1})|C_{i-1}]]\}$, $S_1 = IV$,

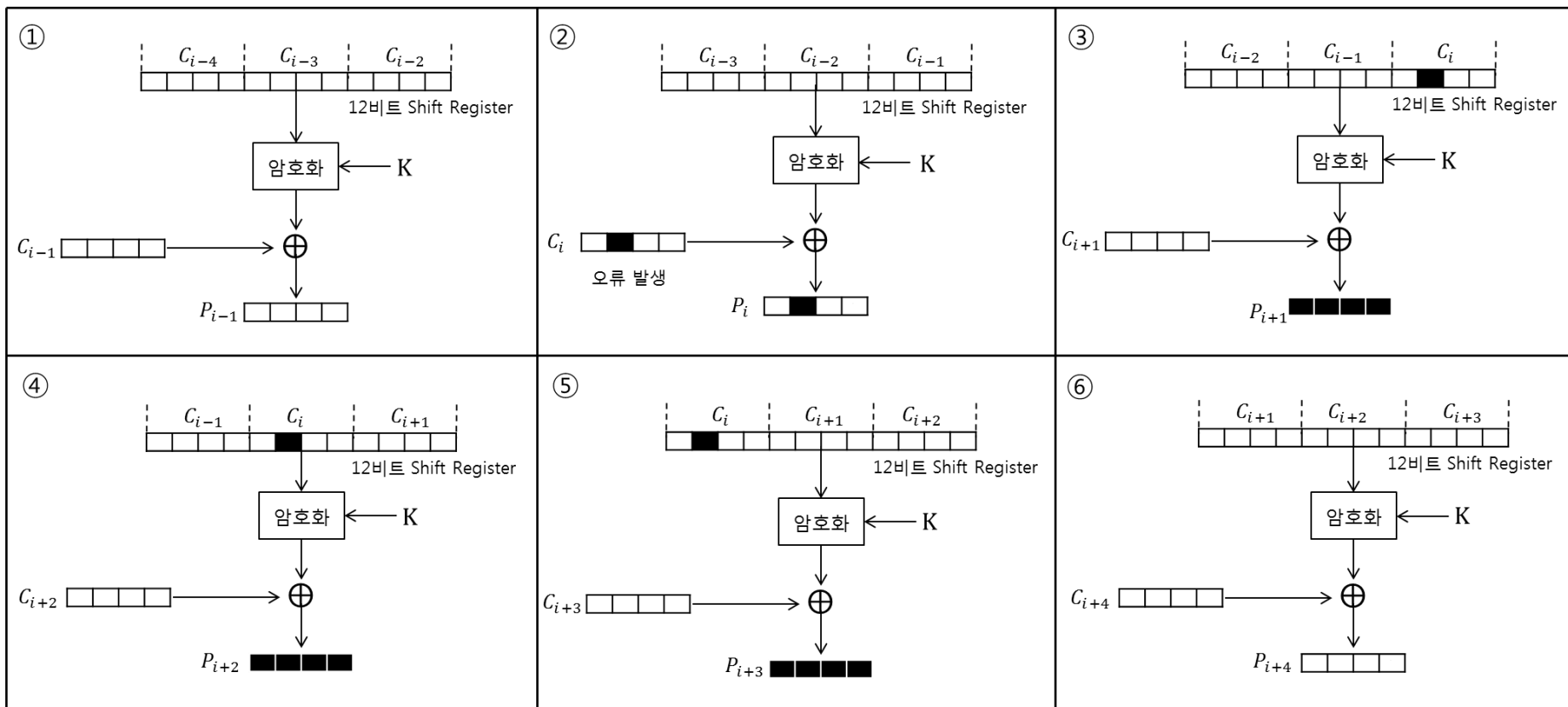
복호화 : $P_i = C_i \oplus \text{SelectLeft}_r\{E_K[\text{ShiftLeft}_r(S_{i-1})|C_{i-1}]]\}$, $S_1 = IV$,

키 스트림
생성기



4.4 CFB (Cipher Feedback) 모드

- 오류 확산(Error Propagation)
 - ✗ 자기 동기식(self-synchronizing)



4.5 OFB (Output Feedback) 모드

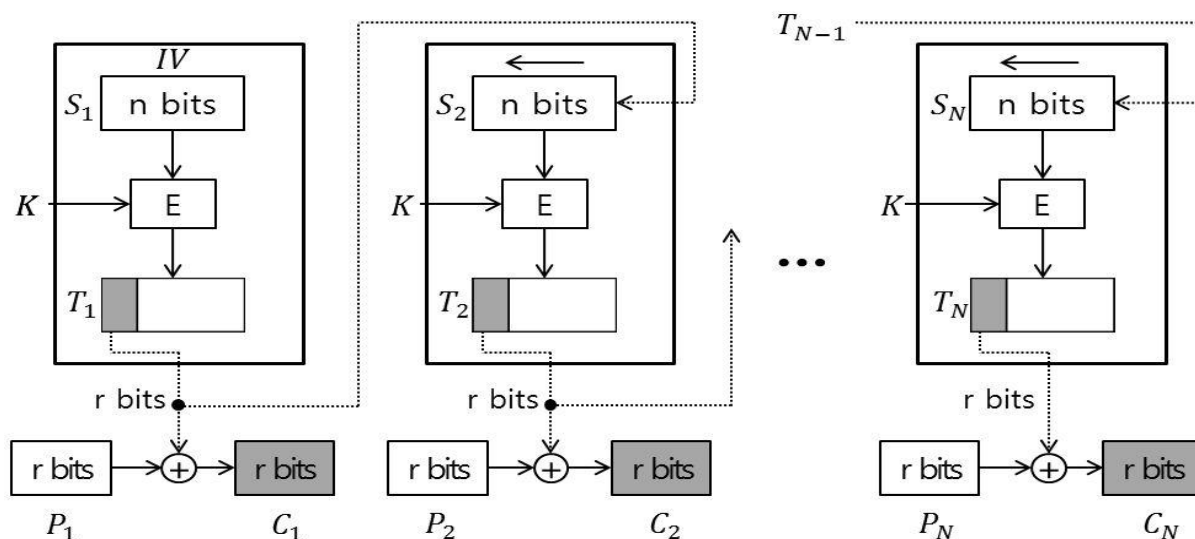
■ 동기식 스트림 암호(synchronizing stream cipher)

- ✕ 암호화와 복호화 과정에서 XOR연산되는 키 스트림이 평문과 암호문에 독립적

암호화 : $C_i = P_i \oplus k_i, i = 1, 2, 3, \dots, N$

복호화 : $P_i = C_i \oplus k_i, i = 1, 2, 3, \dots, N$

- ▶ $k_i = \text{SelectLeft}_r(E_K(S_i)), S_1 = IV,$
 $S_i = \text{ShiftLeft}_r(S_{i-1}) \parallel k_{i-1} (i = 2, 3, \dots, N)$



4.6 CTR (Counter) 모드

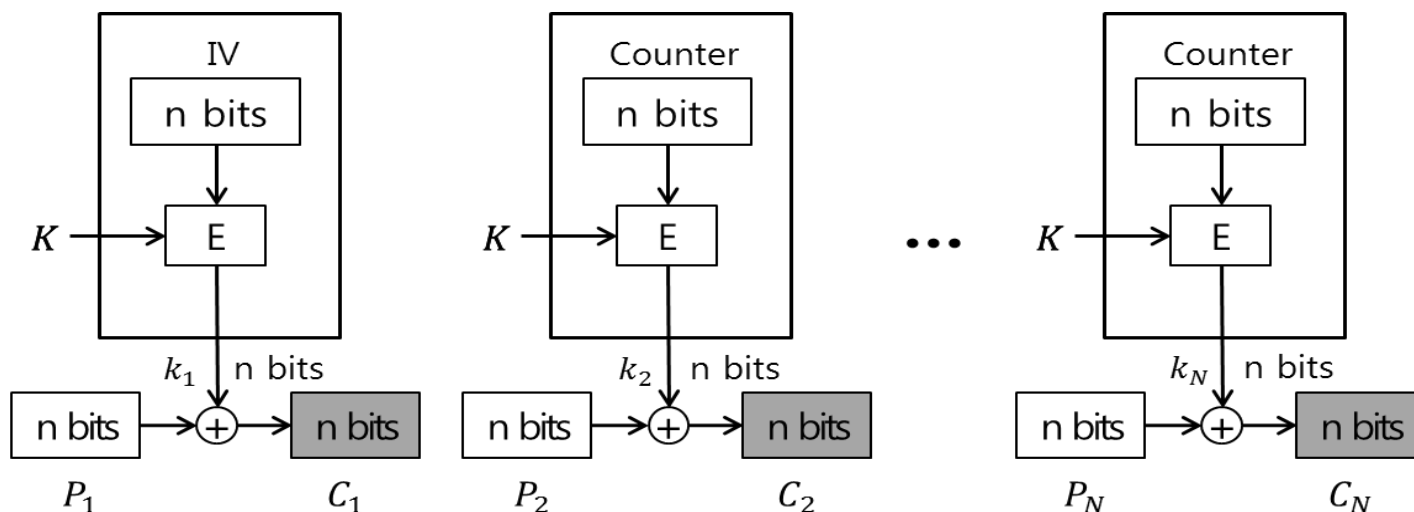
- CTR이 암호화됨 → 단 모든 평문 블록마다 CTR은 달라야 함

- ▶ 가장 간단한 방법은 $\text{CTR} := \text{CTR} + 1$

- ✖ 전처리, 병렬처리 가능

암호화 : $C_i = P_i \oplus E_K(\text{Counter})$, $i = 1, 2, 3, \dots, N$

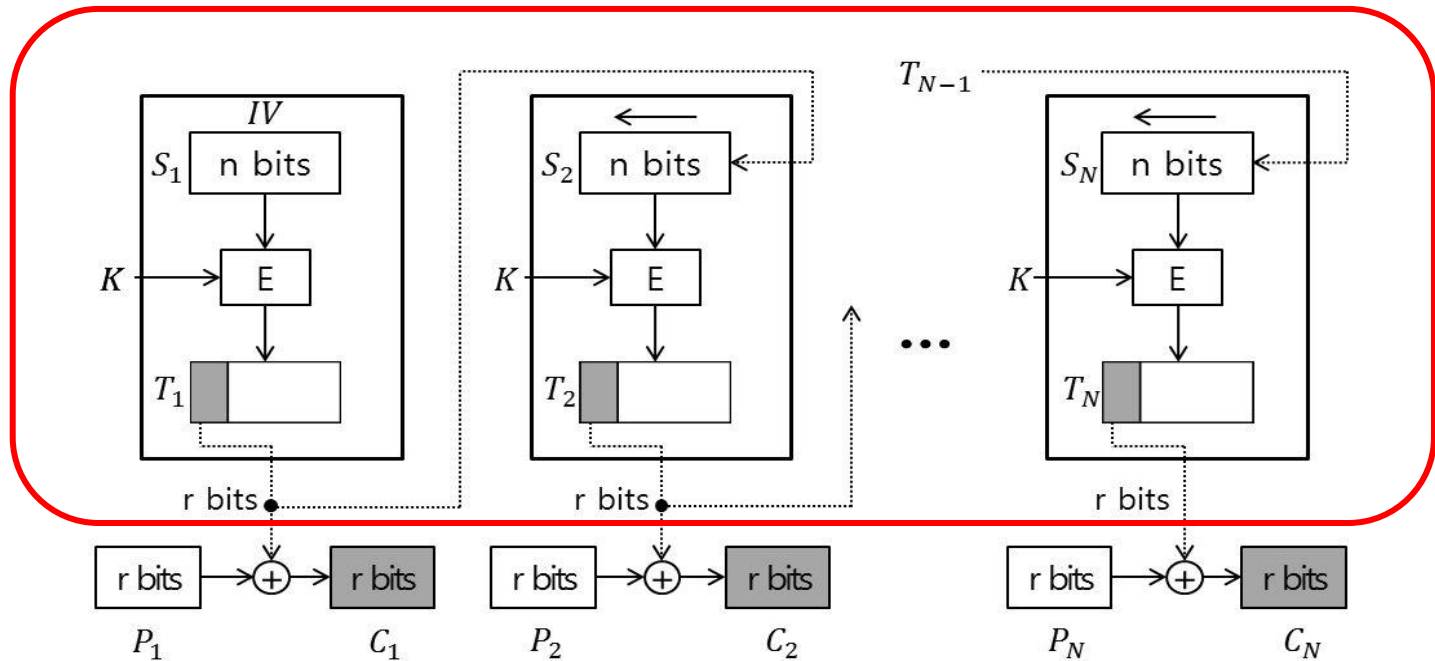
복호화 : $P_i = C_i \oplus E_K(\text{Counter})$, $i = 1, 2, 3, \dots, N$



4.7 각 운영모드의 특징 비교

■ 전처리

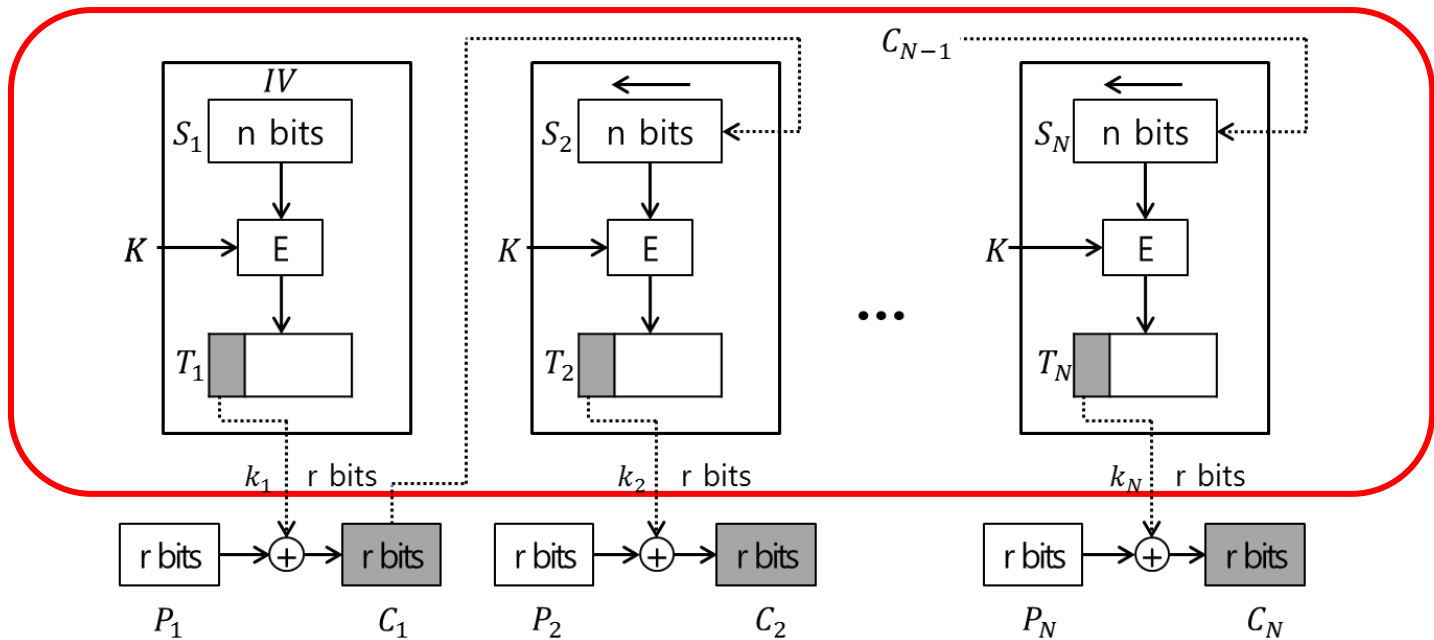
✕ OFB



4.7 각 운영모드의 특징 비교

■ 병렬처리

✕ CFB의 복화시 : 암호문을 이용하여 쉬프트 레지스터를 구성



4.7 각 운영모드의 특징 비교

	ECB	CBC	CFB	OFB	CTR
블록 패턴 유지	○	X	X	X	X
전처리 가능성	X	X	X	○	○
병렬 처리	○	복호화시 가능	복호화시 가능	○	○
오류 확산	X	(P_i, P_{i+1}) 블록에 영향	(P_i, P_{i+1}) 블록에 영향	X	X
암호화 단위	n	n	$r \leq n$	$r \leq n$	$r \leq n$

Appendix: Formal Security Proof

■ Information-theoretical security

- ✗ Requires impractical key length → need a practical one and compromise on perfect security

■ Computational security (page 639)

- ✗ Rely On unproven assumption ($P \stackrel{?}{=} NP$)

✗ relaxation

- ▶ Probabilistic poly-time adversary with very small success (negligible) prob.
- ▶ f is **negligible** if for every polynomial $p()$, there is an N s.t. for all integer $n > N$, $f(n) < 1/p(n)$.

1. Concrete approach

- ▶ Quantifies the security by explicitly bounding the max. success prob. of an adversary running for at most some amount of time

2. Asymptotic approach

- ▶ Quantifies the security by functions of some security parameter (integer n)

Appendix: Formal Security Proof

■ **semantically secure-** by Goldwasser and Micali 1982

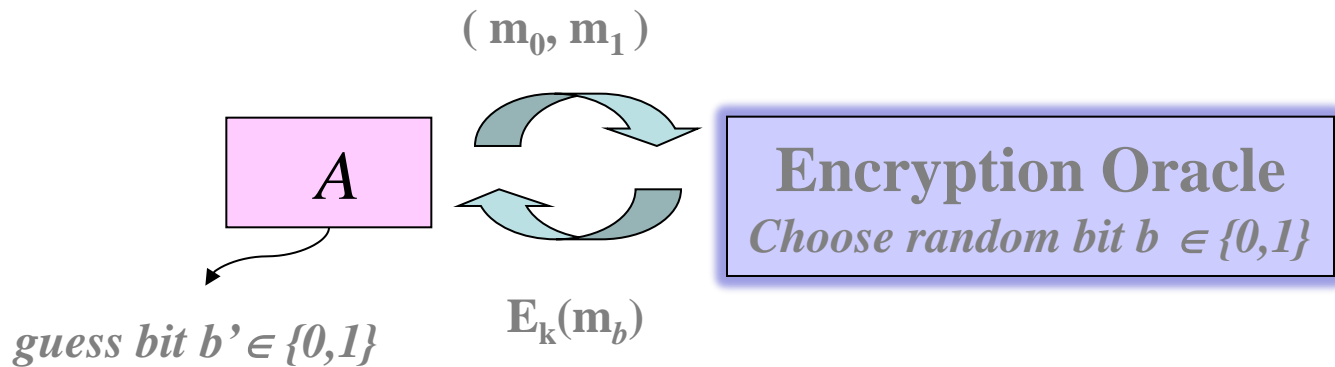
- ✗ infeasible for a computationally-bounded adversary to derive significant information about a plaintext when given only its ciphertext.
- ✗ Considers only the case of a "passive" attacker, and does not consider the case of "active" attacker (CPA & CCA).
- ✗ equivalent to the property of **ciphertext indistinguishability**.
 - ▶ an adversary will be unable to distinguish pairs of ciphertexts based on the message they encrypt
- ✗ This equivalence allowed for security proofs of practical cryptosystems, and consequently the indistinguishability definition is used more commonly than the original definition of semantic security.

Appendix: Formal Security Proof

- **Ciphertext Indistinguishability (= Semantic Security) :**
Indistinguishability under passive attacker is defined by the following game:
 1. A probabilistic polynomial time-bounded adversary generates m_0 and m_1 with $|m_0|=|m_1|$, and transmits them to an encryption oracle.
 2. The encryption oracle selects one of the messages randomly, encrypts the message under the encryption key, and returns the resulting **challenge ciphertext** c to the adversary.
 3. The underlying cryptosystem is Ciphertext indistinguishable if the adversary cannot determine which of the two messages was chosen by the oracle, with probability significantly greater than $1/2$.

Appendix: Formal Security Proof

■ Ciphertext Indistinguishability (= Semantic Security)



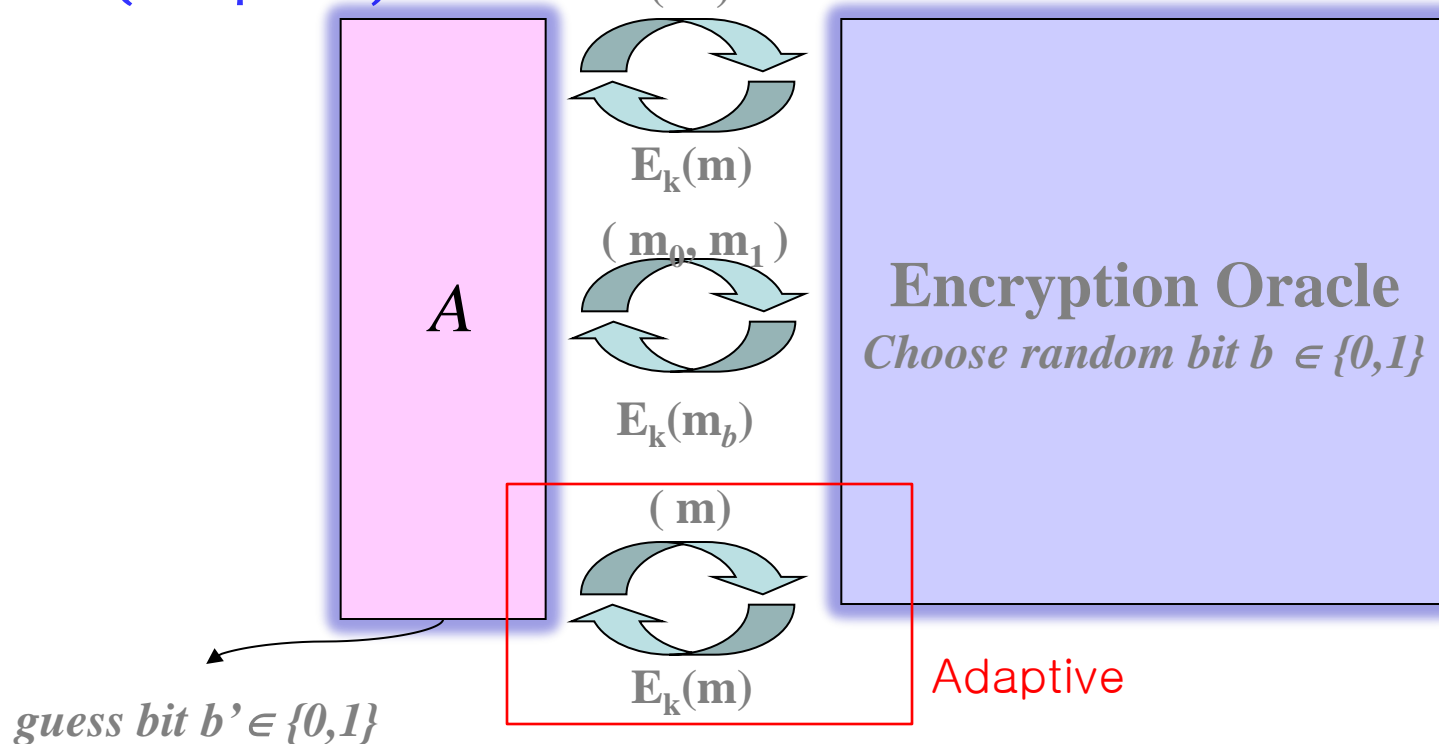
- The cryptosystem has indistinguishable encryptions if any PPT adversary guesses b correctly with probability at most $0.5 + \epsilon(n)$, where ϵ is negligible.

Appendix: Formal Security Proof

- **IND-CPA** : Indistinguishability under CPA is defined by the following game:
 1. A probabilistic polynomial time-bounded adversary is given an encryption oracle access, which it may use to generate any number of ciphertexts (within polynomial bounds).
(Encryption Training course)
 2. The adversary generates m_0 and m_1 with $|m_0|=|m_1|$, and transmits them to an encryption oracle.
 3. The encryption oracle selects one of the messages randomly, encrypts the message under the encryption key, and returns the resulting **challenge ciphertext** c to the adversary.
 4. The underlying cryptosystem is IND-CPA if the adversary can not determine which of the two messages was chosen by the oracle, with probability significantly greater than $1/2$.

Appendix: Formal Security Proof

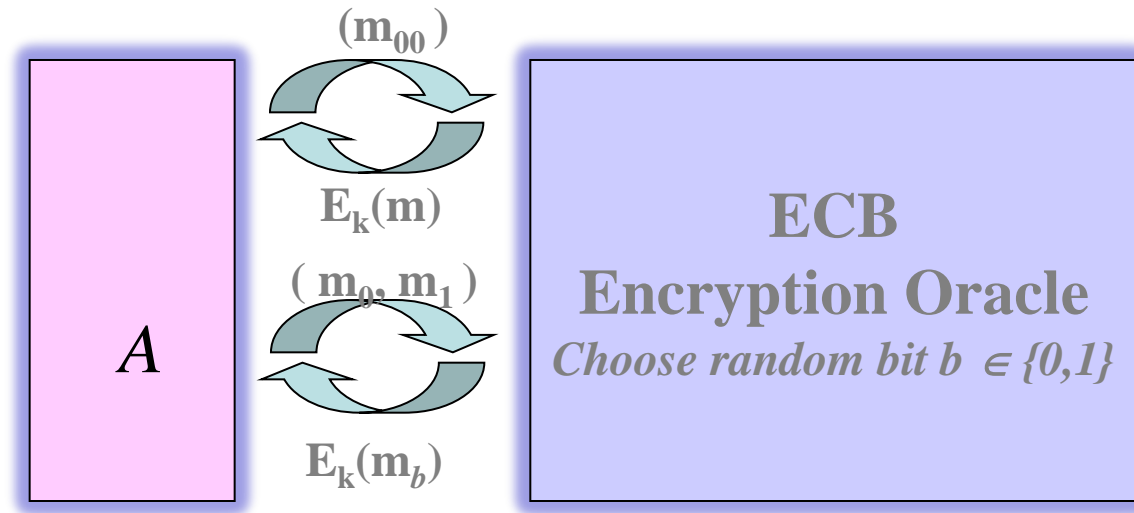
■ (adaptive) IND-CPA



- The cryptosystem has indistinguishable encryptions under CPA if any PPT adversary guesses b correctly with probability at most $0.5 + \epsilon(n)$, where ϵ is negligible.

Example: ECB is not CPA-secure

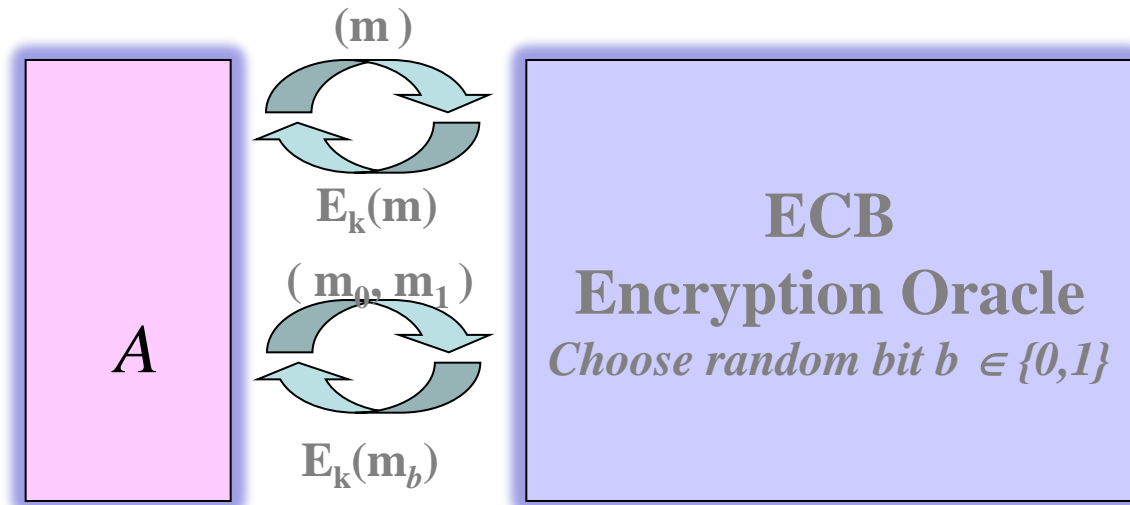
- When A is not allowed to query \mathbf{m}_0 and \mathbf{m}_1 in encryption training course.
- Let $\mathbf{m}_0 = \mathbf{m}_{00} || \mathbf{m}_{01}$ and $\mathbf{m}_1 = \mathbf{m}_{10} || \mathbf{m}_{11}$



- A can check if the first block of $E_k(\mathbf{m}_b)$ is equal to $E_k(\mathbf{m}_{00})$

Example: ECB is not CPA-secure

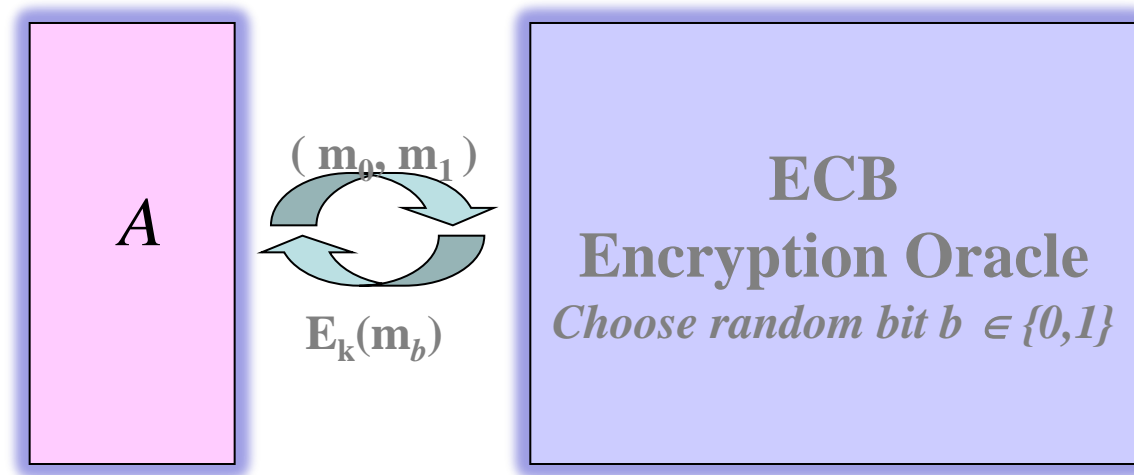
- In general, A is allowed to make any encryption query in encryption training course.



- An adversary A can encryption m_0 and m_1 to guess a bit b .
- An IND-CPA secure encryption scheme must by definition be probabilistic, possessing a component of randomness. Why?

Example: ECB is not CPA-secure

- Even worse, ECB is not semantically secure. WHY?



Let $m_0 = m_{00} || m_{00}$ and $m_1 = m_{10} || m_{11}$. It is easy to guess b !

Appendix: Formal Security Proof

- **IND-CCA** : Indistinguishability under CCA is defined by the following game:
 1. A probabilistic polynomial time-bounded adversary is given a **decryption oracle access** as well as an encryption oracle access. ([Encryption & Decryption training course](#))
 2. The adversary generates m_0 and m_1 with $|m_0|=|m_1|$, and transmits them to an encryption oracle.
 3. The encryption oracle selects one of the messages randomly, encrypts the message under the public key, and returns the resulting **challenge ciphertext** c to the adversary.
 4. The underlying cryptosystem is IND-CCA if the adversary can not determine which of the two messages was chosen by the oracle, with probability significantly greater than $1/2$.

Appendix: Formal Security Proof

■ (adaptive) IND-CCA

