

블록암호 (3)

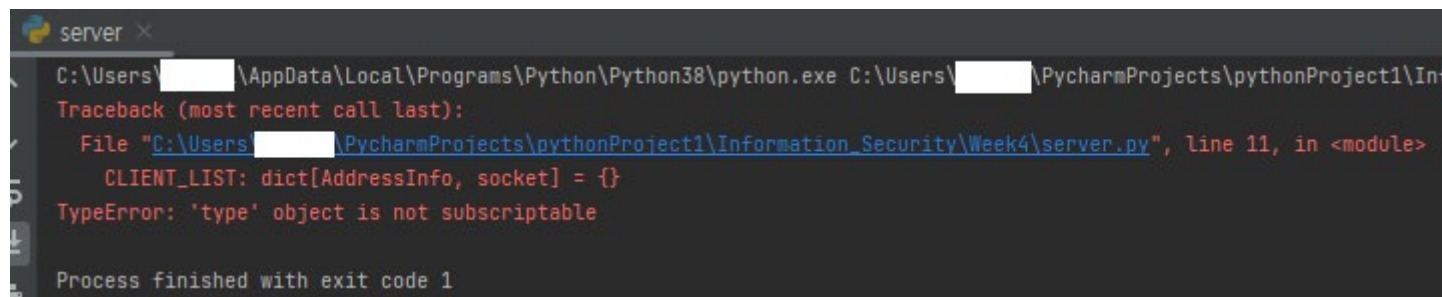
2022년 9월 28일 수요일

정보보호

충남대학교 정보보호연구실 허강준

과제?

- type is not subscriptable



```
server x
C:\Users\ [redacted] \AppData\Local\Programs\Python\Python38\python.exe C:\Users\ [redacted] \PycharmProjects\pythonProject1\Inf
Traceback (most recent call last):
  File "C:\Users\ [redacted] \PycharmProjects\pythonProject1\Information_Security\Week4\server.py", line 11, in <module>
    CLIENT_LIST: dict[AddressInfo, socket] = {}
TypeError: 'type' object is not subscriptable

Process finished with exit code 1
```

- Python 3.10 이상으로 업그레이드 or 타입 힌트 제거

과제?

- 종단간 (End-To-End Encryption; E2EE) 암호 통신기
 - 서버는 절대 메시지를 볼 수 없지 (사실 아님)

```
PS C:\Users\patche\Documents\GitHub\CipherCommunicator> python server.py
[*] Key generated: b'\xfc\xad\xdb\x73|9\x9f\xab\x93\xae\xee'
[*] Server started on 0.0.0.0:24000
[*] Accepted a connection from ('127.0.0.1', 6239)
[*] Accepted a connection from ('127.0.0.1', 6240)
('127.0.0.1', 6239): b'\xf8\xca\xdb\xba\xef\xc3\xc5\xa3\xa7Tb\x9fg\xfb\x1d%\xafb*\x85H\xa9V\x0e\xba\xa1\xca"\xc2\xa7b\xcew3\x9ei\x9b\xc1\x90\xc6\xa4U{'\x1f\xcdj\x10\xcb\xa1\xec\x97A)\xe9\xa5\x8c\xe4o\x1b'
Broadcast_send: ('127.0.0.1', 6240)
('127.0.0.1', 6240): b'l\xef\x84\xc3\x179\x8d\t\xcd\x02o\x88E\x3\xff\x02\xfb\xa8?_ \xc7M\x0b\x98<\x7f\x8c\xa7k\xcf\xc5\xee{g\rU0b\x0b\xda\x94\x8f\xfb\x84\x0bw\xb8\x1f'
Broadcast_send: ('127.0.0.1', 6239)
```

```
PS C:\Users\patche\Documents\GitHub\CipherCommunicator> python client.py
[*] connected to 127.0.0.1:24000, Receiving an encryption key...
[*] Key received: b'\xfc\xad\xdb\x73|9\x9f\xab\x93\xae\xee'
[*] Now a chatting session is starting...
Message: 이건 비밀인데 서버 주인은 절대 모를걸 ㅋㅋ
Me: 이건 비밀인데 서버 주인은 절대 모를걸 ㅋㅋ
Message:
Received: 뭔데? 나한테만 말해보셈 🤔
```

```
Windows PowerShell
PS C:\Users\patche\Documents\GitHub\CipherCommunicator> python client.py
[*] connected to 127.0.0.1:24000, Receiving an encryption key...
[*] Key received: b'\xfc\xad\xdb\x73|9\x9f\xab\x93\xae\xee'
[*] Now a chatting session is starting...
Message:
Received: 이건 비밀인데 서버 주인은 절대 모를걸 ㅋㅋ
뭔데? 나한테만 말해보셈 💎
Me: 뭔데? 나한테만 말해보셈 🤔
Message: |
```

DES - Data Encryption Standard

• 의문점

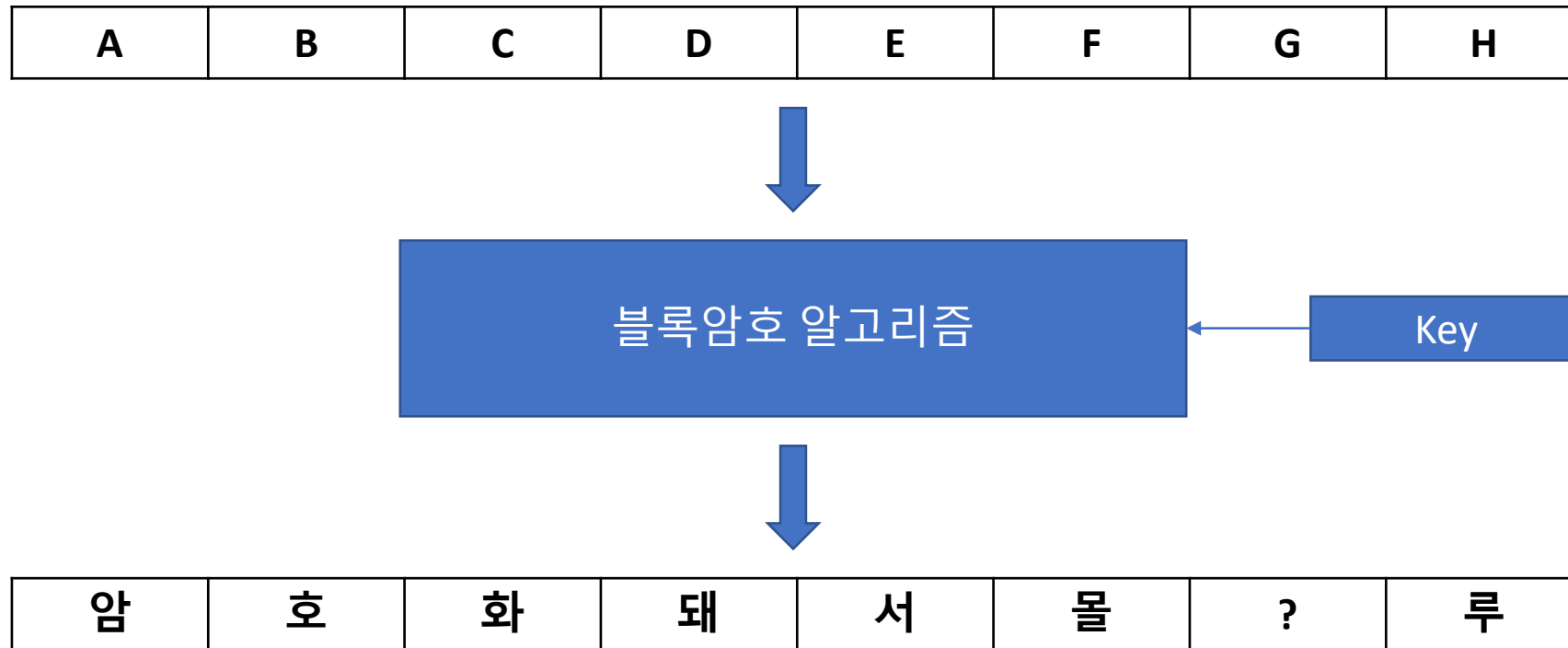
- DES나 AES는 길이 상관 없이 전부 암호화 가능
- 그런데 왜 알고리즘에서는 블록 크기 만큼만 암호화 하냐?
 - 블록 크기보다 작거나 큰 데이터는 어떻게 암호화 하냐?

• 다음주 아론 시간 (or) 다다음주 설습 때 설명

- 선행학습을 위한 키워드: 운용 모드, 패딩

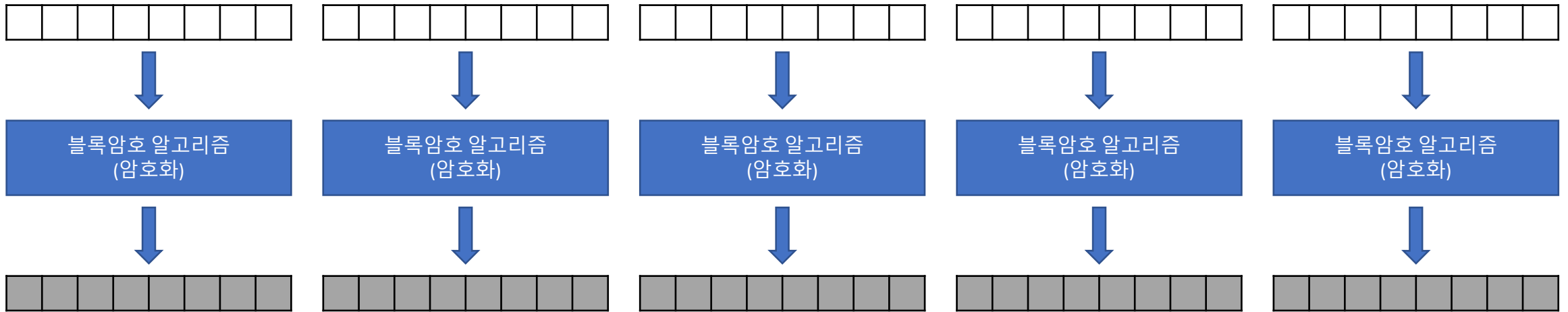
운용모드 (Operation Mode)

- 블록암호는 블록단위로 암호화
 - 블록 크기보다 큰 데이터를 암호화 하는 방법?



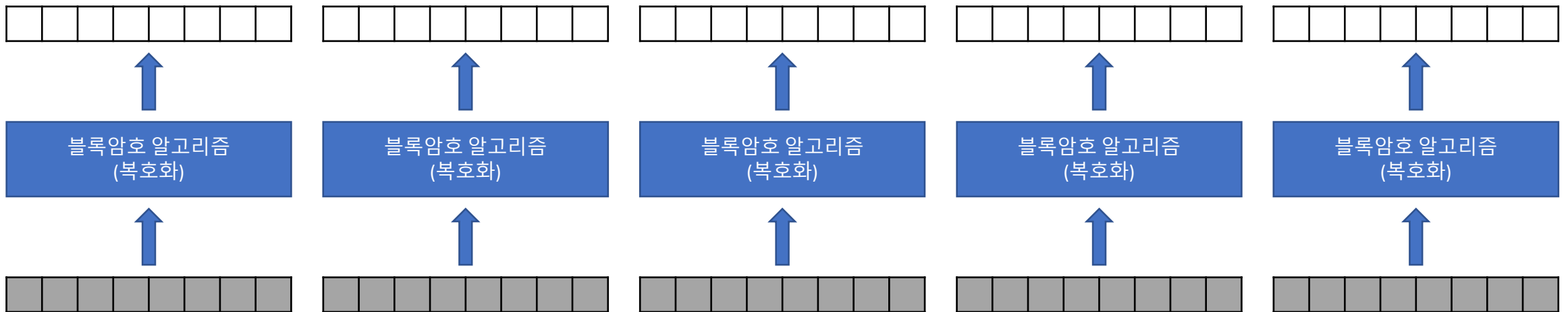
운용모드 (Operation Mode)

- 그냥 블록별로 *같은 키로* 암호화 한다음 합치기?
 - Electronic Code Book (ECB)



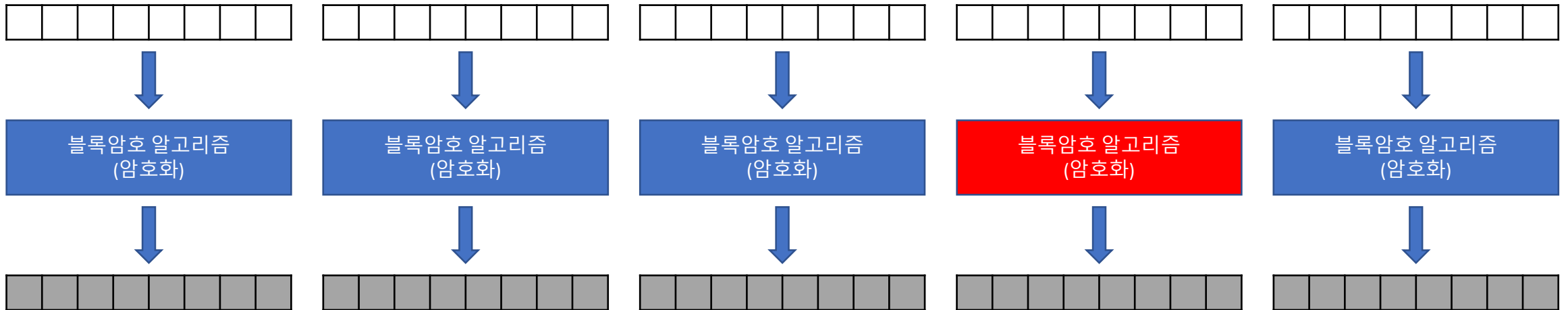
운용모드 (Operation Mode)

- 그냥 블록별로 *같은 키/로* 암호화 한다음 합치기?
 - Electronic Code Book (ECB)



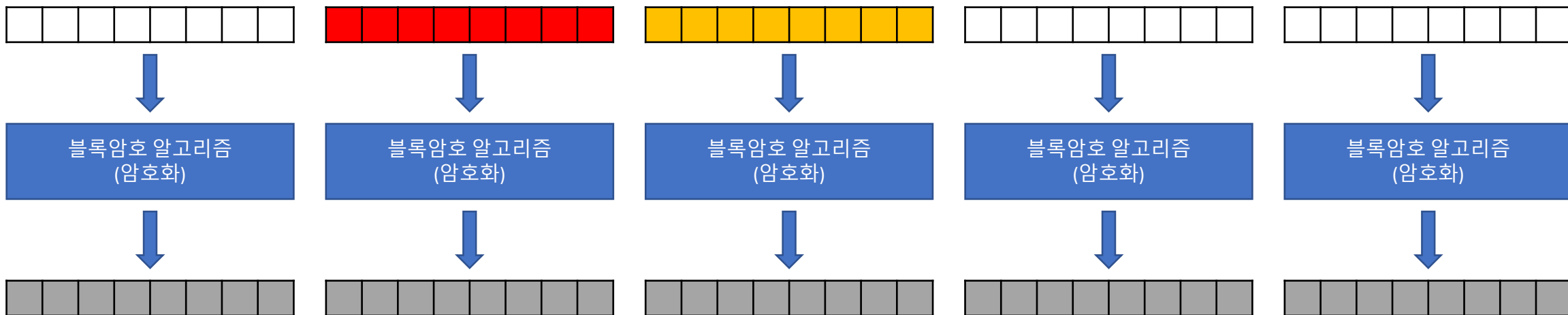
운용모드 (Operation Mode)

- 이러면 어떻게 하지?



운용모드 (Operation Mode)

- 이러면 어떻게 하지?



운용모드 (Operation Mode)

- One Time Pad : Revisited

- OTP를 그대로 이용하긴 힘들지만 장점을 활용할 수는 없을까?

XOR (Exclusive-OR)

- One Time Pad

- 평문의 길이와 키 길이가 같은 암호
- 철!태! 뚫는다!

평문: 10110011 11110000 00001111 11101110 11111111

암호키: 10000001 11111111 11110000 11001100 00001111

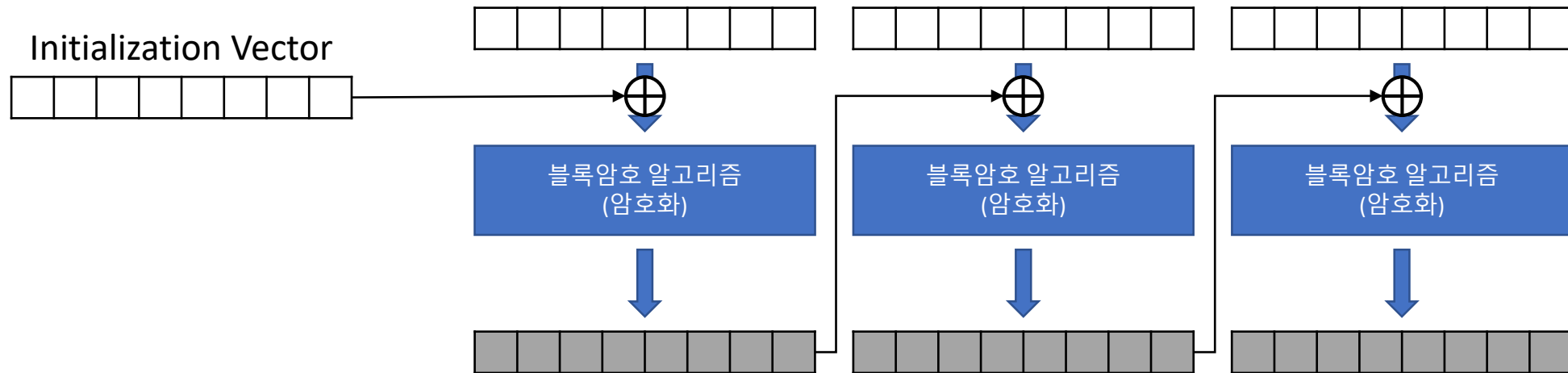
암호문: ~~00110010~~ ~~00001111~~ ~~11111111~~ ~~00100010~~ ~~11110000~~

암호키: 10000001 11111111 11110000 11001100 00001111

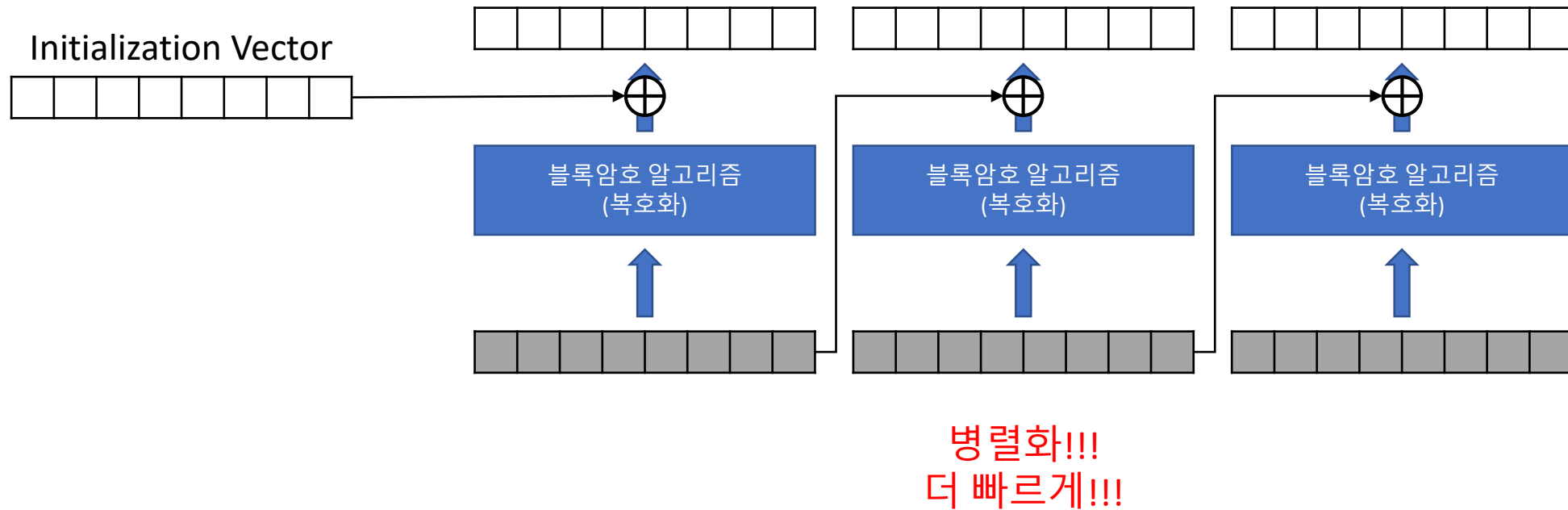
평문: 10110011 11110000 00001111 11101110 11111111

- 이렇게나 강력한데... 왜 대중화 되지 않았을까

- Cipher Block Chaining(CBC)

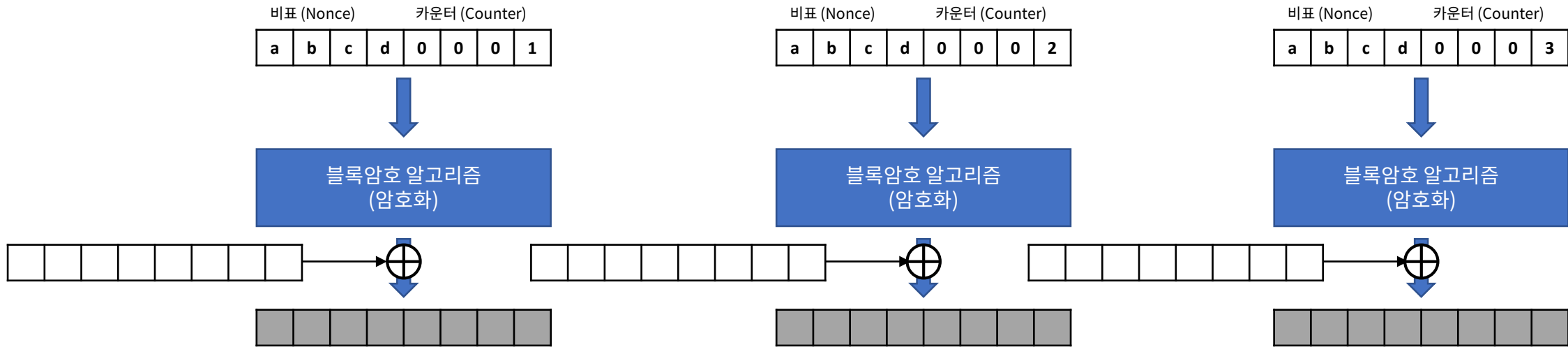


- Cipher Block Chaining(CBC)



운용모드 (Operation Mode)

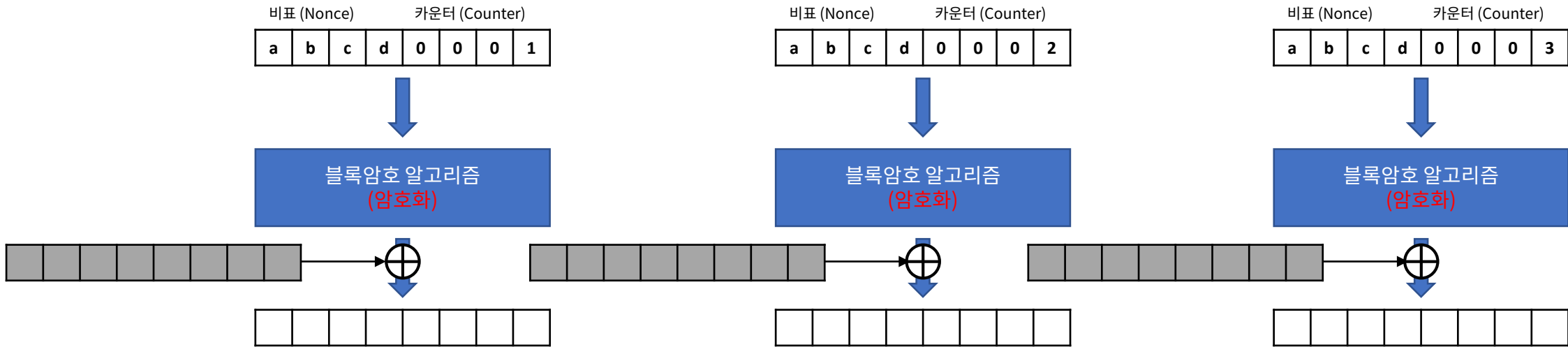
- CounTeR (CTR)



암호화도 병렬화!!!
훨씬 더 빠르게!!!

운용모드 (Operation Mode)

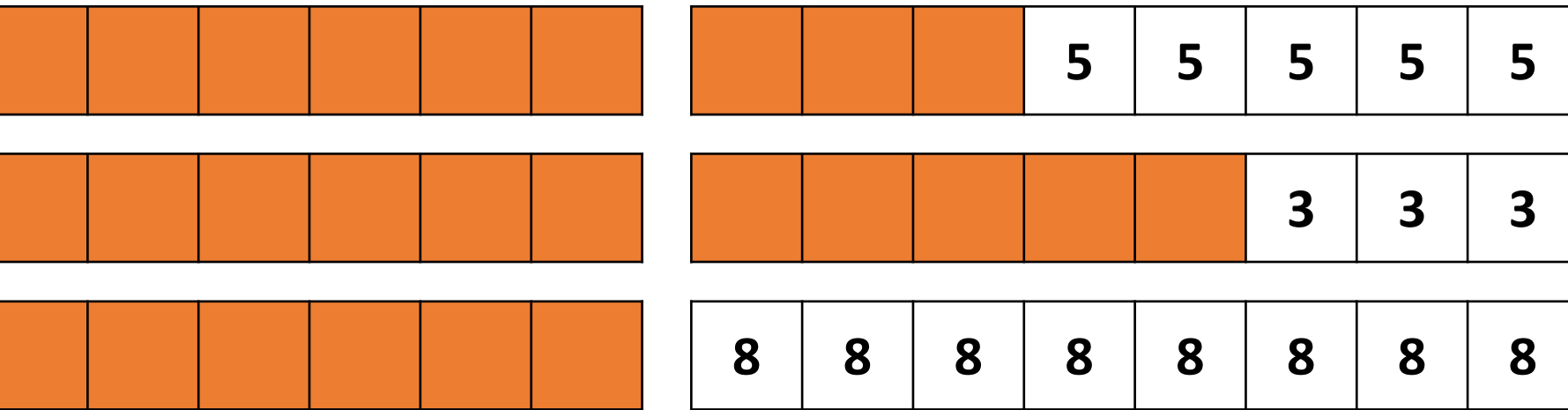
- CounTeR (CTR)



복호화도 병렬화!!!
훨씬 더더 빠르게!!!

패딩 (Padding)

- 평문이 항상 블록 크기의 배수만큼일까?
- DES -64bit, AES -128bit;
 - 항상 블록 크기의 배수만큼 평문이 주어지지 않음
- PKCS#5, PKCS#7: 비어있는 갯수만큼 숫자로 채우기 (표준)
 - 없으면 블록 하나 더 만들어 채움



과제!

- S-DES-ECB / S-DES-CBC 구현하기

```
def sdes_encrypt_ecb(text: bytearray, key: bytearray):  
    pass  
  
def sdes_decrypt_ecb(ciphertext: bytearray, key: bytearray):  
    pass  
  
def sdes_encrypt_cbc(text: bytearray, key: bytearray, iv: bytearray):  
    pass  
  
def sdes_decrypt_cbc(ciphertext: bytearray, key: bytearray, iv: bytearray):  
    pass
```

운용모드별로 암호화/복호화 구현
ECB, CBC

```
plaintext = input("[*] Input Plaintext in Binary: ")  
key = input("[*] Input Key in Binary (10bits): ")  
  
# Plaintext must be multiple of 8 and Key must be 10 bits.  
if len(plaintext) % 8 != 0 or len(key) != 10:  
    raise ArgumentError("Input Length Error!!!")
```

평문 입력 제한 변경
→ 8비트 (블록 크기)의 배수

과제!

- S-DES-ECB / S-DES-CBC 구현하기

```
result_encrypt = sdes_encrypt_ecb(bits_plaintext, bits_key)

print(f"Encrypted (ECB): {result_encrypt}")

result_decrypt = sdes_decrypt_ecb(result_encrypt, bits_key)

print(f"Decrypted (ECB): {result_decrypt}, Expected: {bits_plaintext}")

if result_decrypt != bits_plaintext:
    print(f"S-DES-ECB FAILED...")
else:
    print(f"S-DES-ECB SUCCESS!!!")

random_iv = bytearray(Random().randrange(0, 255))
print(f"IV will be random...{random_iv}")

result_encrypt = sdes_encrypt_cbc(bits_plaintext, bits_key, random_iv)

print(f"Encrypted (CBC): {result_encrypt}")

result_decrypt = sdes_decrypt_cbc(result_encrypt, bits_key, random_iv)

print(f"Decrypted (CBC): {result_decrypt}, Expected: {bits_plaintext}")

if result_decrypt != bits_plaintext:
    print(f"S-DES-CBC FAILED...")
else:
    print(f"S-DES-CBC SUCCESS!!!")
```

IV 자동 생성
→ CBC 모드에서 사용

주차	실습 주제	과제	날짜
1	오리엔테이션 & 썰풀기	과제를 위한 GitHub 설정	9/7
2	카이사르&비즈네르 암호	ENIGMA	9/14
3	XOR과 블록암호	Simplified DES 구현하기	9/21
4	여러가지 블록암호	블록암호를 이용하여 암호통신기 완성하기	9/28
5	블록암호 운용모드	S-DES-CBC, S-DES-ECB 구현하기	10/5
6	RSA	RSA 구현하기, 저강도 RSA 크랙하기	10/12
7	해시	암호통신기에 무결성 검증 기능 추가하기	10/19
8	중 간 고 사 (10/24)		공강
9	메세지 인증코드(MAC)	HMAC 구현하기	11/2
10	디지털 서명	사실인증서 생성 및 프로그램 코드 서명	11/9
11	하이브리드 암호	하이브리드 기반 암호 통신기	11/16
12	난수	시드값 추측을 이용한 암호문 크랙	11/23
13	블록체인과 머클 트리	머클트리 구현하기	11/30
14	TLS와 PGP(GPG)	GPG를 이용하여 암호 메일 보내기	12/7
15	기 말 고 사 (12/12)		종강

질문?

- 없으면 자리에서 일어나셔도 좋습니다 :)
- **대 학 원 입 학 문 의**는 언제나 환영
 - 블록체인, Web 3, 해킹 관심있거나 유경험자 우대

입학문의

- 류재철 교수님 (jcryou [at] cnu.ac.kr)
- 허강준 조교 (knowledge [at] o.cnu.ac.kr)