충남대학교
CHUNGNAM NATIONAL UNIVERSITY

전자서명

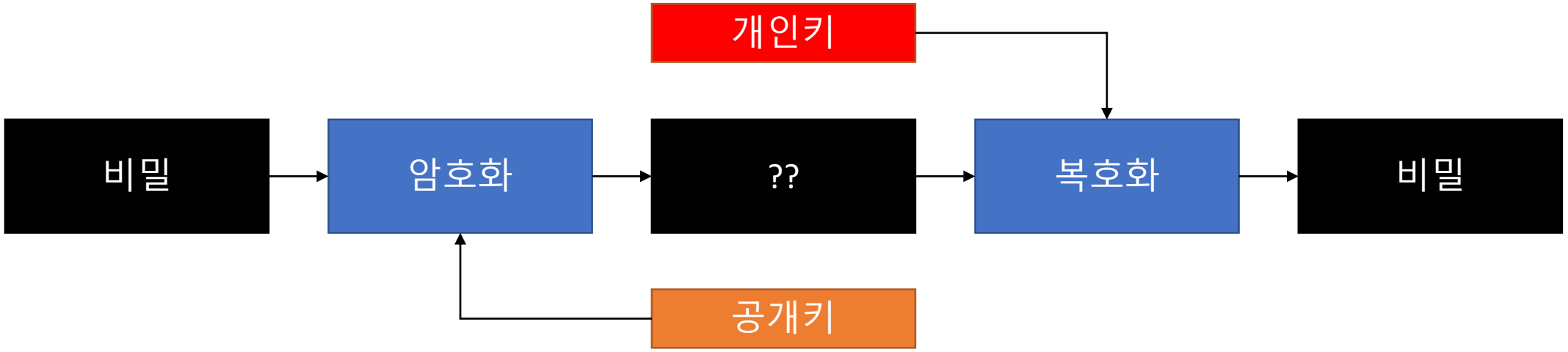
2022년 11월 23일 수요일

정보보호

충남대학교 정보보호연구실 허강준

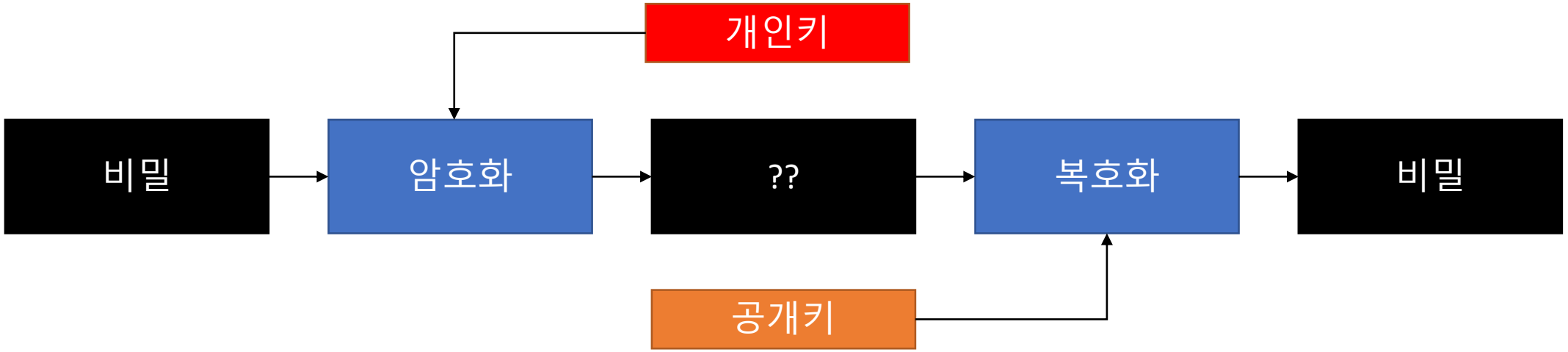
공개키 암호

- **공개**키암호 → 공개키로 암호화, 비밀키로 복호화



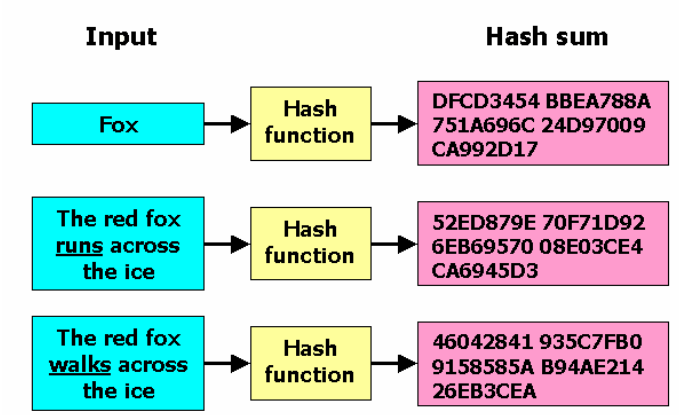
공개키 암호

- 개인키 → 외부에 노출되어선 안됨
- 개인키로 암호화할 경우 공개키로만 복호화 가능… 누구나 복호화 가능?



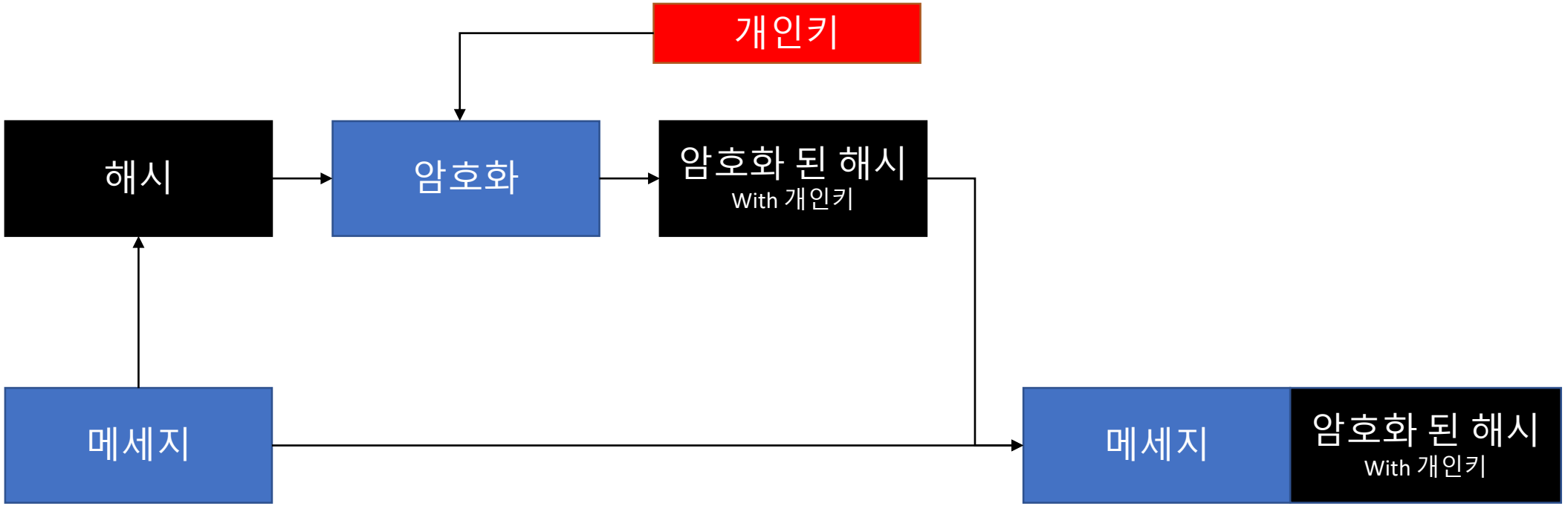
해시

- “디지털 지문” : 데이터의 무결성 보장을 위한 방법
- 해시가 같다 → 매우 높은 확률로 정확히 같은 데이터임



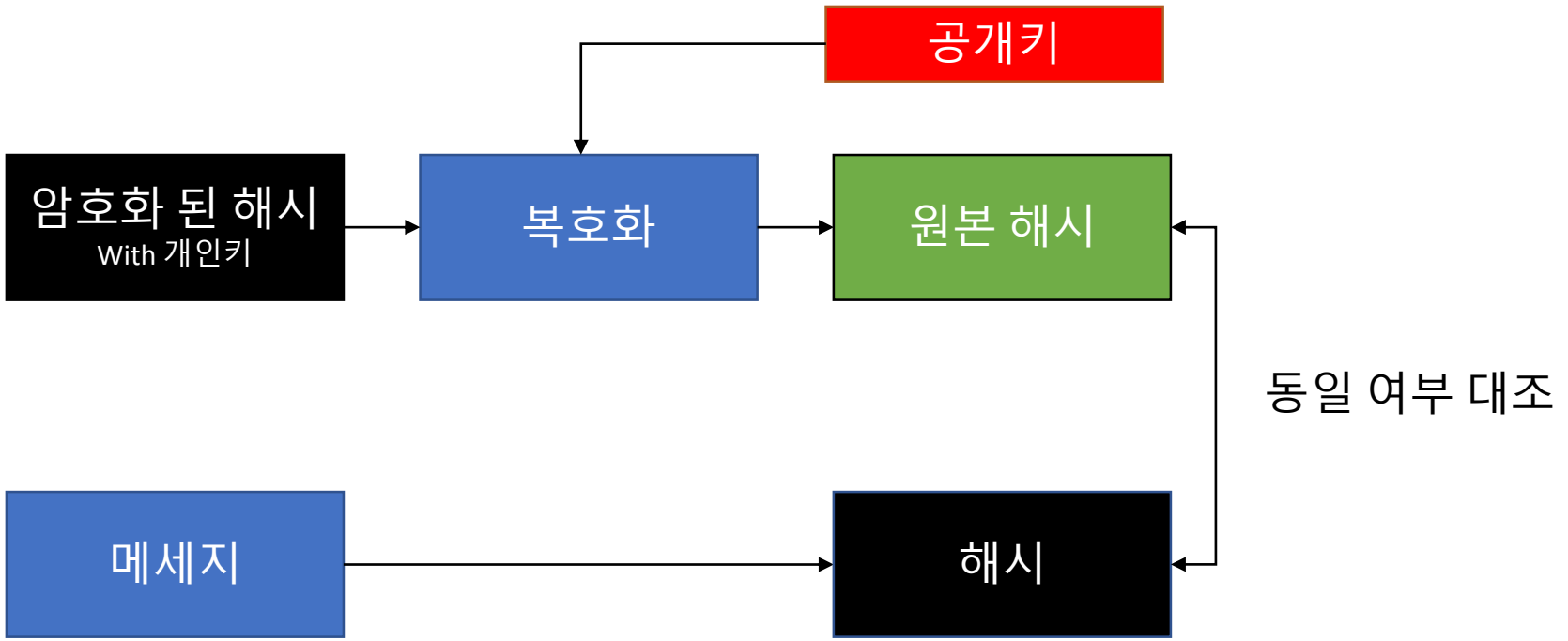
공개키 암호

- 어떤 데이터에 대한 해시를 개인키로 암호화



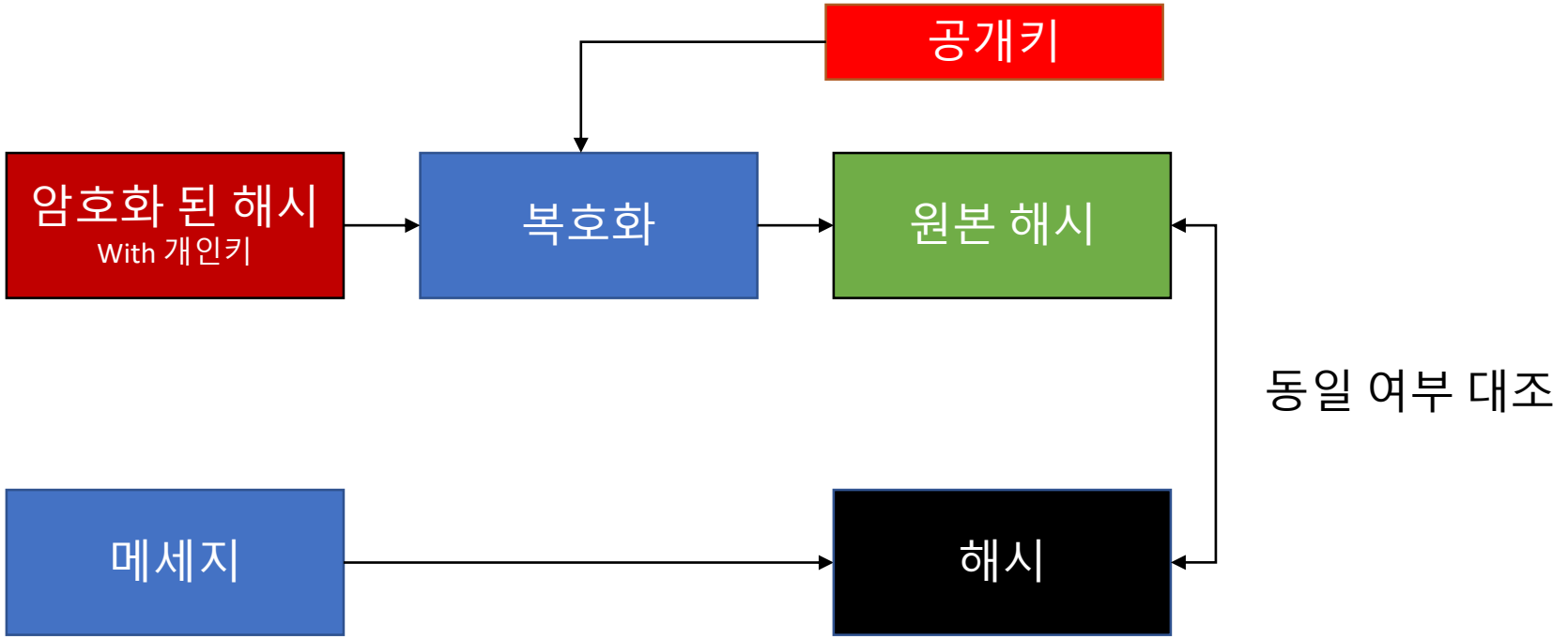
공개키 암호

- 암호화 된 해시를 공개키로 복호화 한다면?



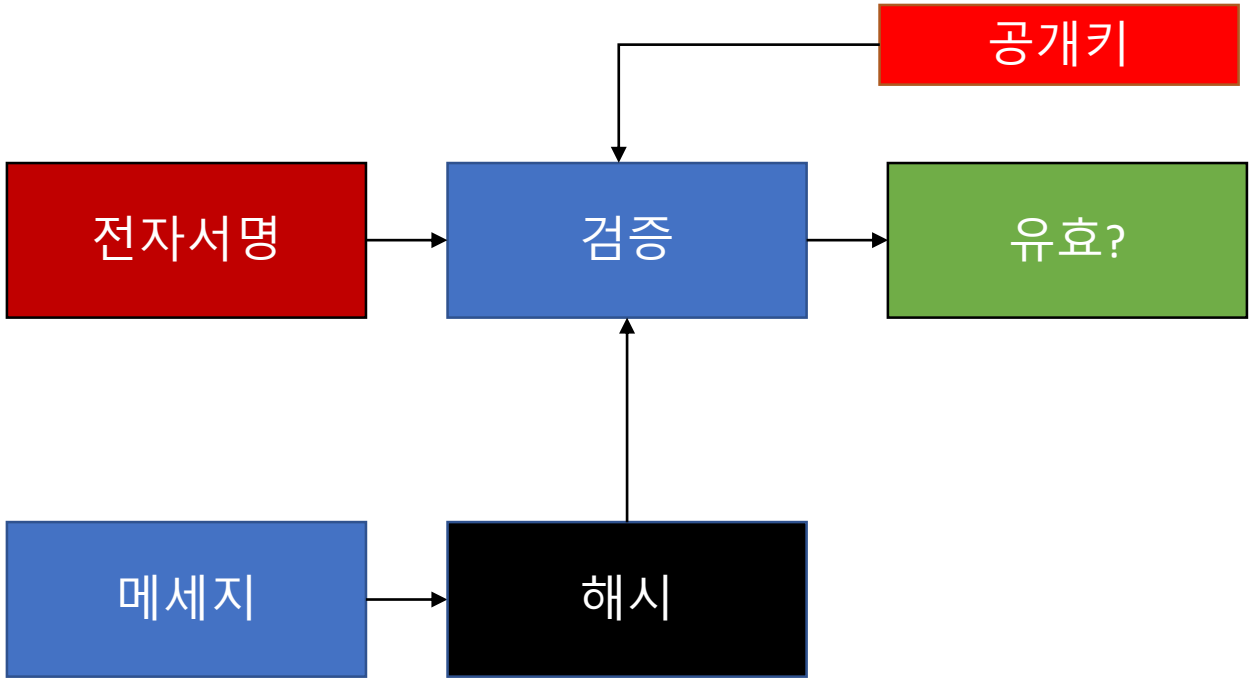
전자서명

- 개인키로 암호화 할 수 있는 주체는 한정적
- “부인방지” : 누가 데이터를 생성했는지 검증할 수 있음

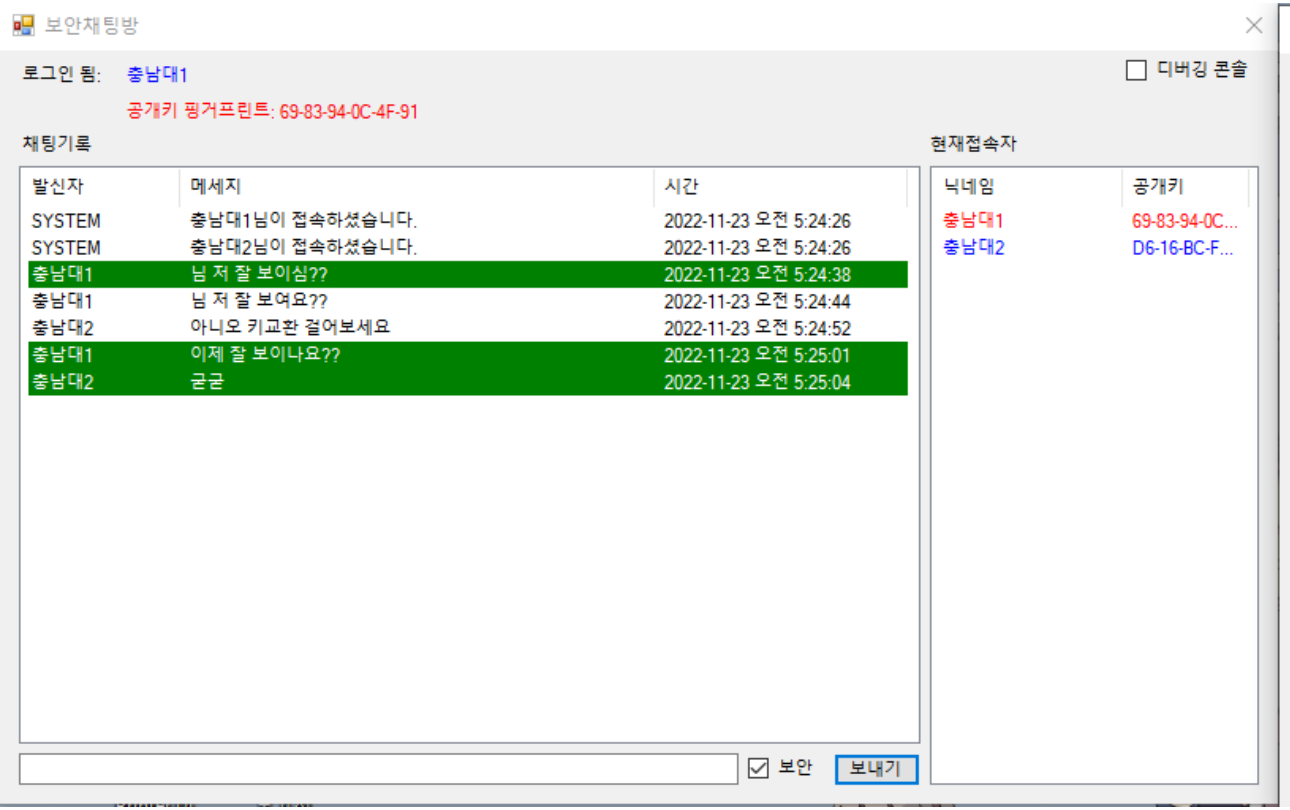


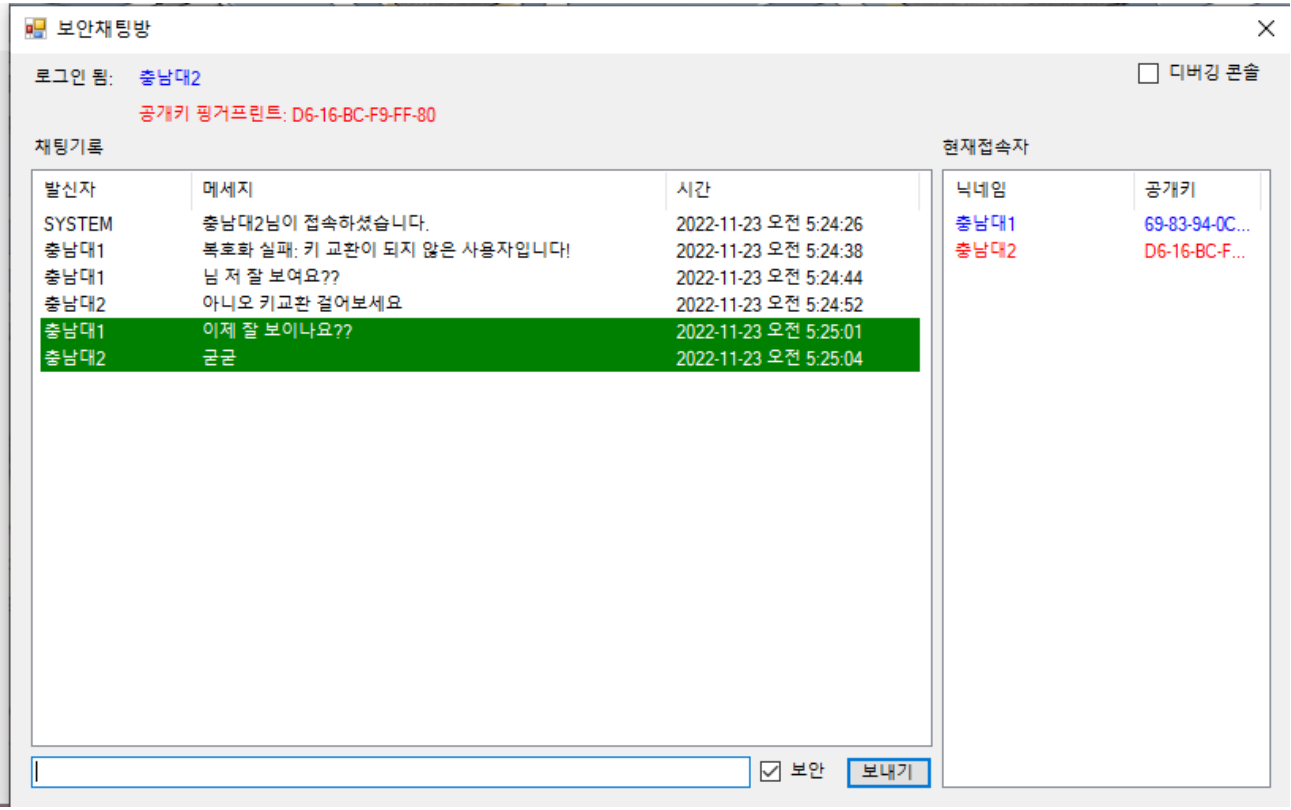
전자서명

- 개인키로 암호화 할 수 있는 주체는 한정적
- “부인방지” : 누가 데이터를 생성했는지 검증할 수 있음

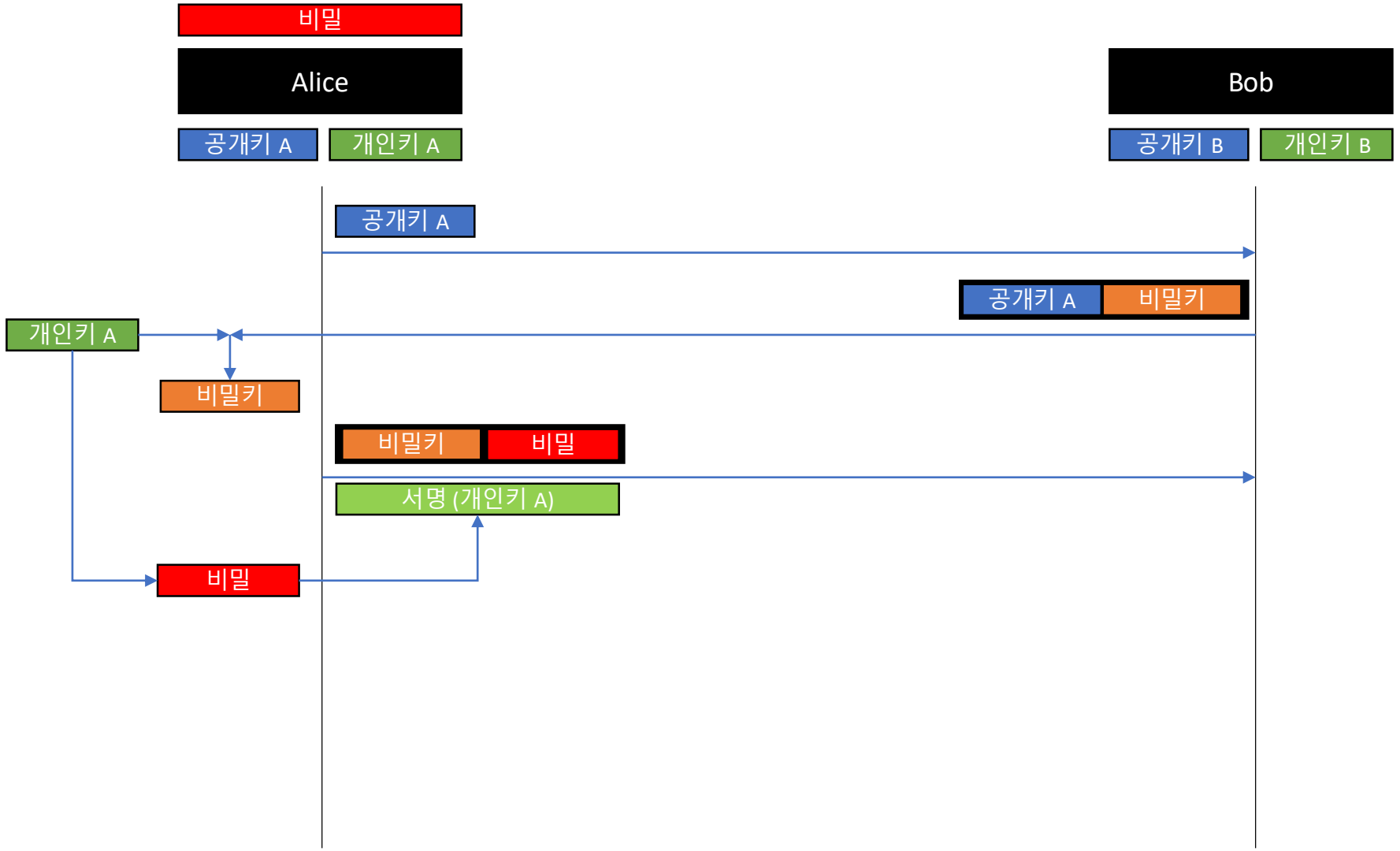


암호채팅 프로그램…


보안채팅방
로그인 됨: 충남대1
디버깅 콘솔
공개키 핑거프린트: 69-83-94-0C-4F-91
채팅기록
현재접속자
발신자 메세지 시간
SYSTEM 충남대1님이 접속하셨습니다. 2022-11-23 오전 5:24:26
SYSTEM 충남대2님이 접속하셨습니다. 2022-11-23 오전 5:24:26
충남대1 님 저 잘 보이심?? 2022-11-23 오전 5:24:38
충남대1 님 저 잘 보여요?? 2022-11-23 오전 5:24:44
충남대2 아니오 키교환 걸어보세요 2022-11-23 오전 5:24:52
충남대1 이제 잘 보이나요?? 2022-11-23 오전 5:25:01
충남대2 굳굳 2022-11-23 오전 5:25:04
닉네임 공개키
충남대1 69-83-94-0C...
충남대2 D6-16-BC-F...
보안
보내기


보안채팅방
로그인 됨: 충남대2
디버깅 콘솔
공개키 핑거프린트: D6-16-BC-F9-FF-80
채팅기록
현재접속자
발신자 메세지 시간
SYSTEM 충남대2님이 접속하셨습니다. 2022-11-23 오전 5:24:26
충남대1 복호화 실패: 키 교환이 되지 않은 사용자입니다! 2022-11-23 오전 5:24:38
충남대1 님 저 잘 보여요?? 2022-11-23 오전 5:24:44
충남대2 아니오 키교환 걸어보세요 2022-11-23 오전 5:24:52
충남대1 이제 잘 보이나요?? 2022-11-23 오전 5:25:01
충남대2 굳굳 2022-11-23 오전 5:25:04
닉네임 공개키
충남대1 69-83-94-0C...
충남대2 D6-16-BC-F...
보안
보내기

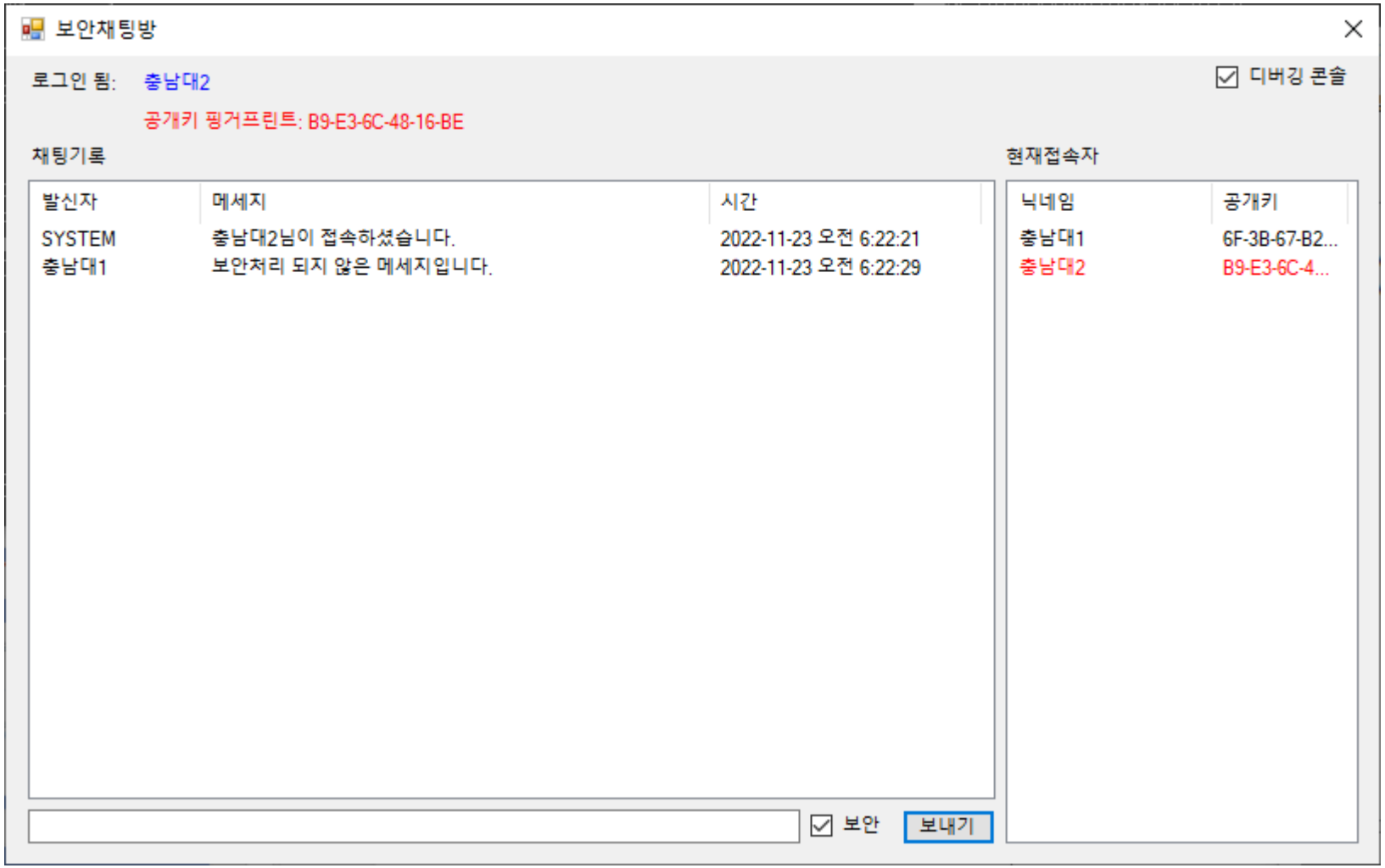
암호채팅 프로그램…

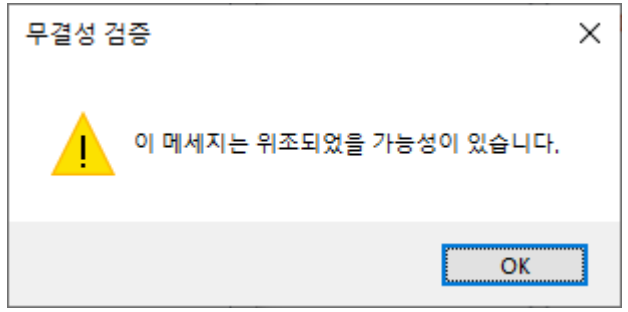

비밀
Alice
Bob
공개키 A
개인키 A
공개키 B
개인키 B
공개키 A
공개키 A
비밀키
개인키 A
비밀키
비밀키
비밀
서명 (개인키 A)
비밀

암호채팅 프로그램…


비밀
Alice
Bob
공개키 A
개인키 A
공개키 B
개인키 B
공개키 A
공개키 A
비밀키
공개키 A
개인키 A
비밀키
서명 (개인키 A)
비밀키
비밀
비밀키
서명 (개인키 A)
비밀

암호채팅 프로그램…

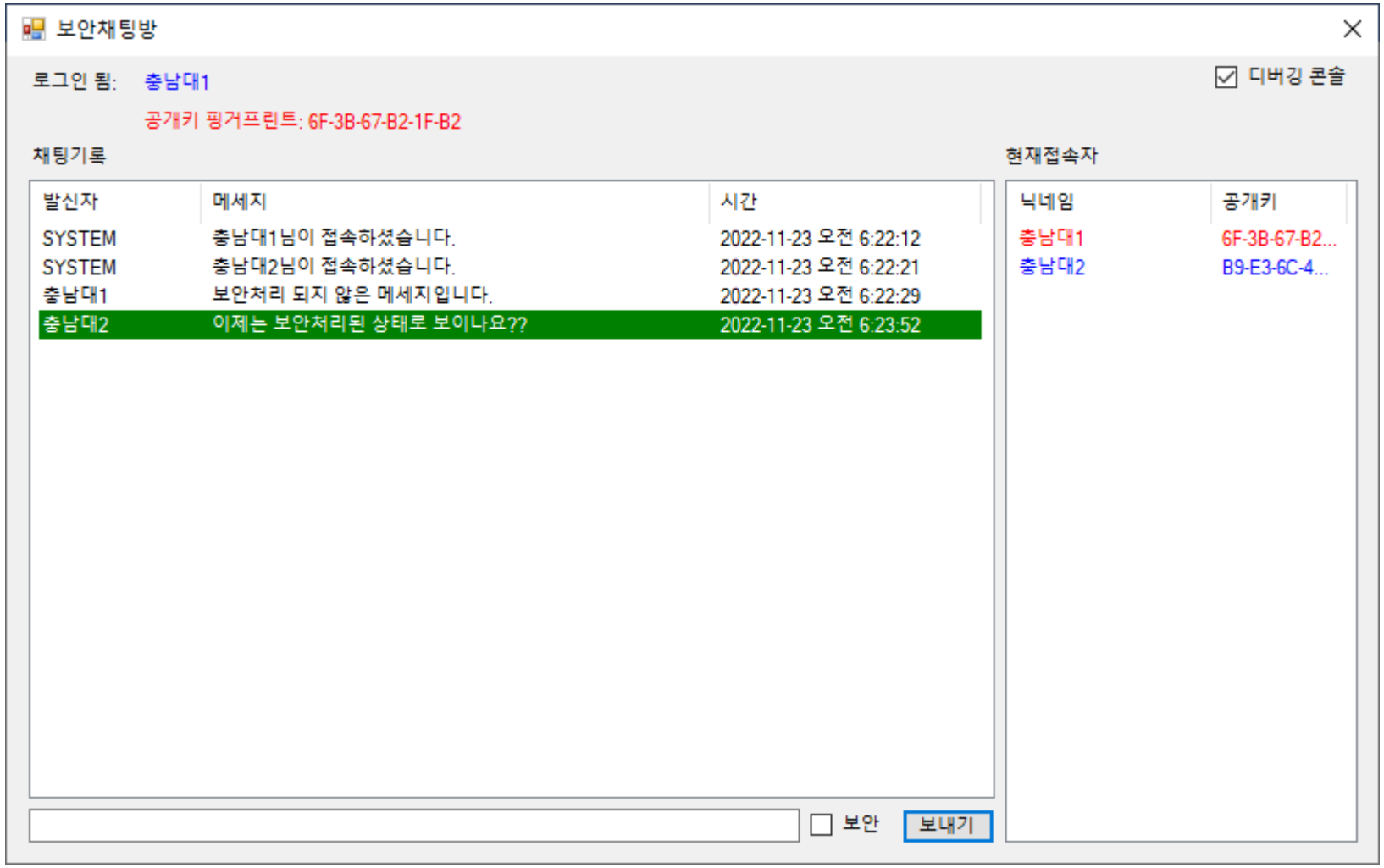

보안채팅방
로그인 됨: 충남대2
디버깅 콘솔
공개키 핑거프린트: B9-E3-6C-48-16-BE
채팅기록
현재접속자
발신자
메세지
시간
닉네임
공개키
SYSTEM
충남대2님이 접속하셨습니다.
2022-11-23 오전 6:22:21
충남대1
6F-3B-67-B2...
충남대1
보안처리 되지 않은 메세지입니다.
2022-11-23 오전 6:22:29
충남대2
B9-E3-6C-4...
보안
보내기

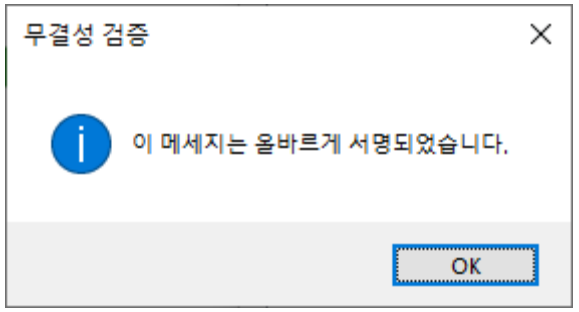

무결성 검증
이 메세지는 위조되었을 가능성이 있습니다.
OK

암호채팅 프로그램…

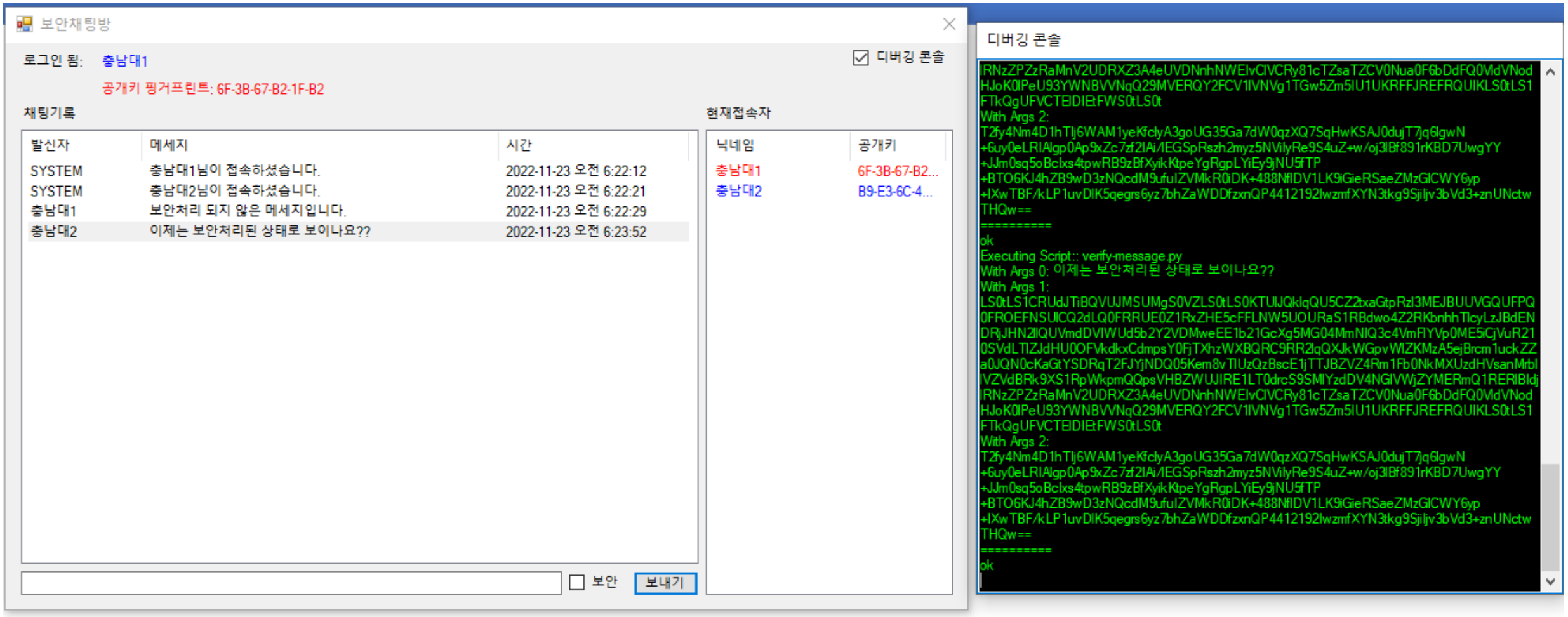

현재접속자
닉네임
공개키
6:22:21
6:22:29
충남대1
6F-3B-67-B2...
키 교환 요청
현재접속자
닉네임
공개키
충남대1
6F-3B-67-B2...
충남대2
B9-E3-6C-4...

암호채팅 프로그램…


보안채팅방
로그인 됨: 충남대1
디버깅 콘솔
공개키 핑거프린트: 6F-3B-67-B2-1F-B2
채팅기록
현재접속자
발신자 메세지 시간
SYSTEM 충남대1님이 접속하셨습니다. 2022-11-23 오전 6:22:12
SYSTEM 충남대2님이 접속하셨습니다. 2022-11-23 오전 6:22:21
충남대1 보안처리 되지 않은 메세지입니다. 2022-11-23 오전 6:22:29
충남대2 이제는 보안처리된 상태로 보이나요?? 2022-11-23 오전 6:23:52
닉네임 공개키
충남대1 6F-3B-67-B2...
충남대2 B9-E3-6C-4...
보안
보내기


무결성 검증
이 메세지는 올바르게 서명되었습니다.
OK

암호채팅 프로그램…

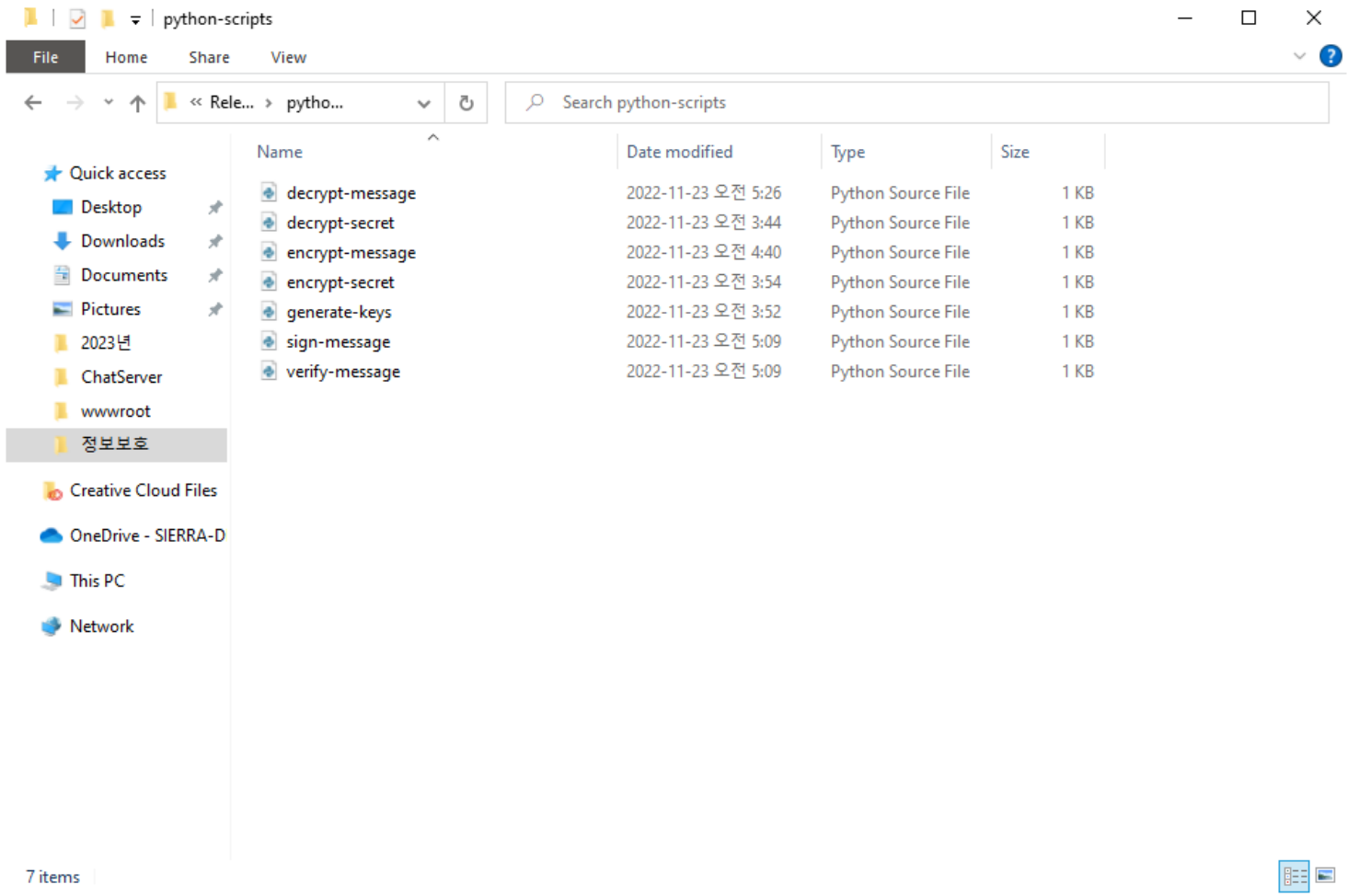

보안채팅방
로그인 됨: 충남대1
디버깅 콘솔
공개키 핑거프린트: 6F-3B-67-B2-1F-B2
채팅기록
현재접속자
발신자 메세지 시간
SYSTEM 충남대1님이 접속하셨습니다. 2022-11-23 오전 6:22:12
SYSTEM 충남대2님이 접속하셨습니다. 2022-11-23 오전 6:22:21
충남대1 보안처리 되지 않은 메세지입니다. 2022-11-23 오전 6:22:29
충남대2 이제는 보안처리된 상태로 보이나요?? 2022-11-23 오전 6:23:52
닉네임 공개키
충남대1 6F-3B-67-B2...
충남대2 B9-E3-6C-4...
보안
보내기
디버깅 콘솔
IRNzZPZzRaMnV2UDRXZ3A4eUVDNnhNWElvClVCRy81cTZsaTZCV0Nua0F6bDdFQ0VldVNod
HJoK0lPeU93YWNBVVNqQ29MVERQY2FCV1IVNVg1TGw5Zm5IU1UKRFFJREFRQUIKLS0tLS1
FTkQgUFVCTElDIEtFWS0tLS0t
With Args 2:
T2fy4Nm4D1hTlj6WAM1yeKfclyA3goUG35Ga7dW0qzXQ7SqHwKSAJ0dujT7jq6lgwN
+6uy0eLRIAlgp0Ap9xZc7zf2lAi/lEGSpRszh2myz5NVilyRe9S4uZ+w/oj3lBf891rKBD7UwgYY
+JJm0sq5oBclxs4tpwRB9zBfXyikKtpeYgRgpLYiEy9jNU5fTP
+BTO6KJ4hZB9wD3zNQcdM9ufuIZVMkR0iDK+488NfIDV1LK9iGieRSaeZMzGlCWY6yp
+IXwTBF/kLP1uvDIK5qegrs6yz7bhZaWDDfzxnQP4412192IwzmfXYN3tkg9Sjiljv3bVd3+znUNctw
THQw==
==========
ok
Executing Script:: verify-message.py
With Args 0: 이제는 보안처리된 상태로 보이나요??
With Args 1:
LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlJQklqQU5CZ2txaGtpRzl3MEJBUUVGQUFPQ
0FROEFNSUlCQ2dLQ0FRRUE0Z1RxZHE5cFFLNW5UOURaS1RBdwo4Z2RKbnhhTlcyLzJBdEN
DRjJHN2lIQUVmdDVIWUd5b2Y2VDMweEE1b21GcXg5MG04MmNIQ3c4VmFlYVp0ME5iCjVuR21
0SVdLTlZJdHU0OFVkdkxCdmpsY0FjTXhzWXBQRC9RR2lqQXJkWGpvWlZKMzA5ejBrcm1uckZZ
a0JQN0cKaGtYSDRqT2FJYjNDQ05Kem8vTlUzQzBscE1jTTJBZVZ4Rm1Fb0NkMXUzdHVsanMrbl
lVZVdBRk9XS1RpWkpmQQpsVHBZWUJIRE1LT0drcS9SMlYzdDV4NGlVWjZYMERmQ1RERlBldj
IRNzZPZzRaMnV2UDRXZ3A4eUVDNnhNWElvClVCRy81cTZsaTZCV0Nua0F6bDdFQ0VldVNod
HJoK0lPeU93YWNBVVNqQ29MVERQY2FCV1IVNVg1TGw5Zm5IU1UKRFFJREFRQUIKLS0tLS1
FTkQgUFVCTElDIEtFWS0tLS0t
With Args 2:
T2fy4Nm4D1hTlj6WAM1yeKfclyA3goUG35Ga7dW0qzXQ7SqHwKSAJ0dujT7jq6lgwN
+6uy0eLRIAlgp0Ap9xZc7zf2lAi/lEGSpRszh2myz5NVilyRe9S4uZ+w/oj3lBf891rKBD7UwgYY
+JJm0sq5oBclxs4tpwRB9zBfXyikKtpeYgRgpLYiEy9jNU5fTP
+BTO6KJ4hZB9wD3zNQcdM9ufuIZVMkR0iDK+488NfIDV1LK9iGieRSaeZMzGlCWY6yp
+IXwTBF/kLP1uvDIK5qegrs6yz7bhZaWDDfzxnQP4412192IwzmfXYN3tkg9Sjiljv3bVd3+znUNctw
THQw==
==========
ok

암호채팅 프로그램…

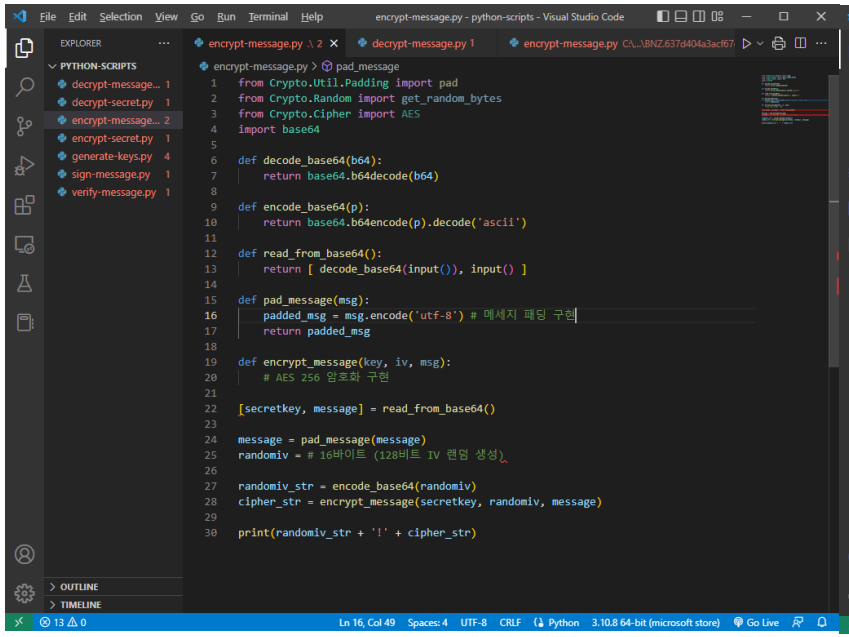
- 13주차까지 완성 목표
- 채팅 프로그램은 소스코드가 공개되어 있음
  - 바로 실행 가능하도록 실행파일도 배포됨

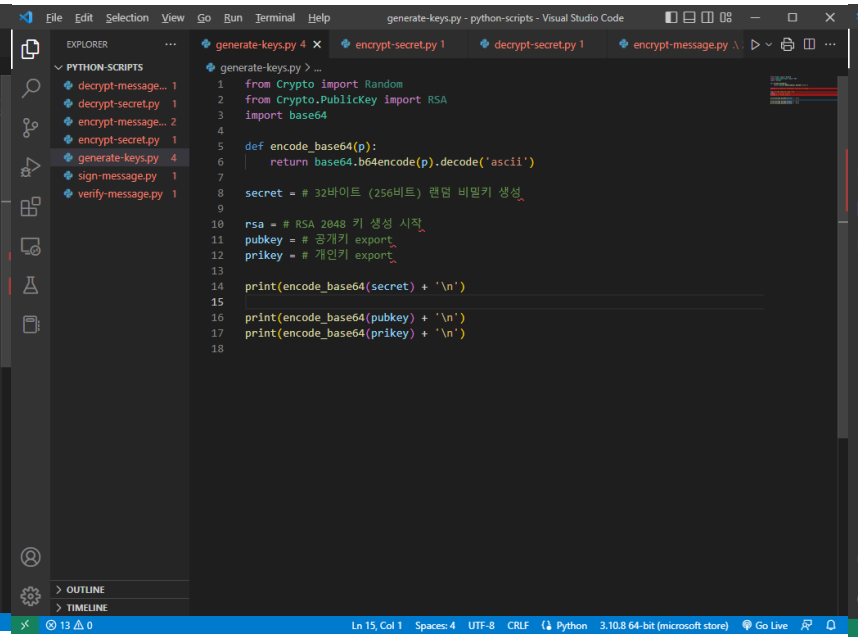
- 매주 구현해야 할 기능 및 스크립트 이름은 레포에 명시
- 이번주:
  - 공개키/개인키, 비밀키 생성
  - 공개키 암호를 이용한 비밀키 암호화 및 복호화
  - 대칭키 암호를 이용한 평문 암호화 및 복호화
  - 전자서명을 이용한 메세지 송신자 검증

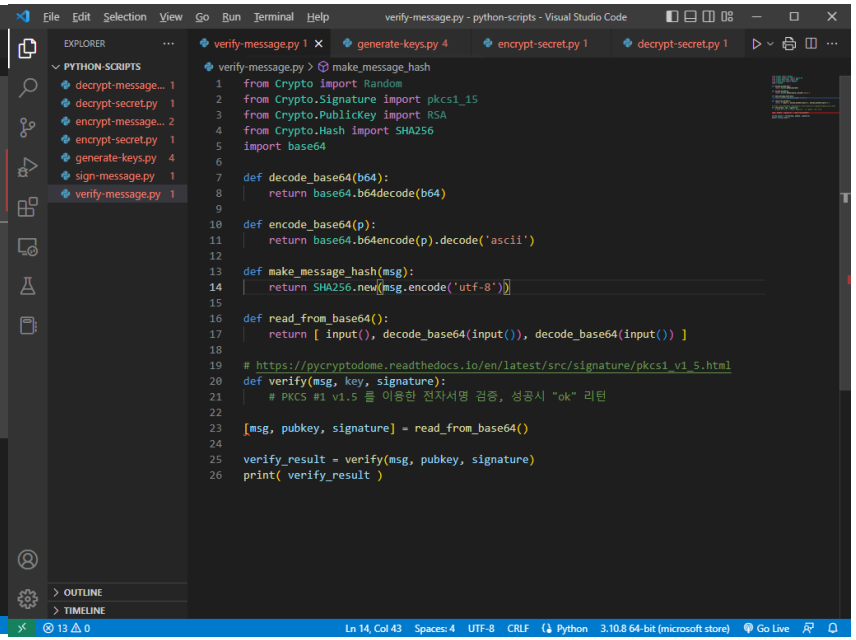
암호채팅 프로그램…


python-scripts
File
Home
Share
View
« Rele... › pytho...
Search python-scripts
Quick access
Desktop
Downloads
Documents
Pictures
2023년
ChatServer
wwwroot
정보보호
Creative Cloud Files
OneDrive - SIERRA-D
This PC
Network
Name
Date modified
Type
Size
decrypt-message 2022-11-23 오전 5:26 Python Source File 1 KB
decrypt-secret 2022-11-23 오전 3:44 Python Source File 1 KB
encrypt-message 2022-11-23 오전 4:40 Python Source File 1 KB
encrypt-secret 2022-11-23 오전 3:54 Python Source File 1 KB
generate-keys 2022-11-23 오전 3:52 Python Source File 1 KB
sign-message 2022-11-23 오전 5:09 Python Source File 1 KB
verify-message 2022-11-23 오전 5:09 Python Source File 1 KB
7 items

암호채팅 프로그램…

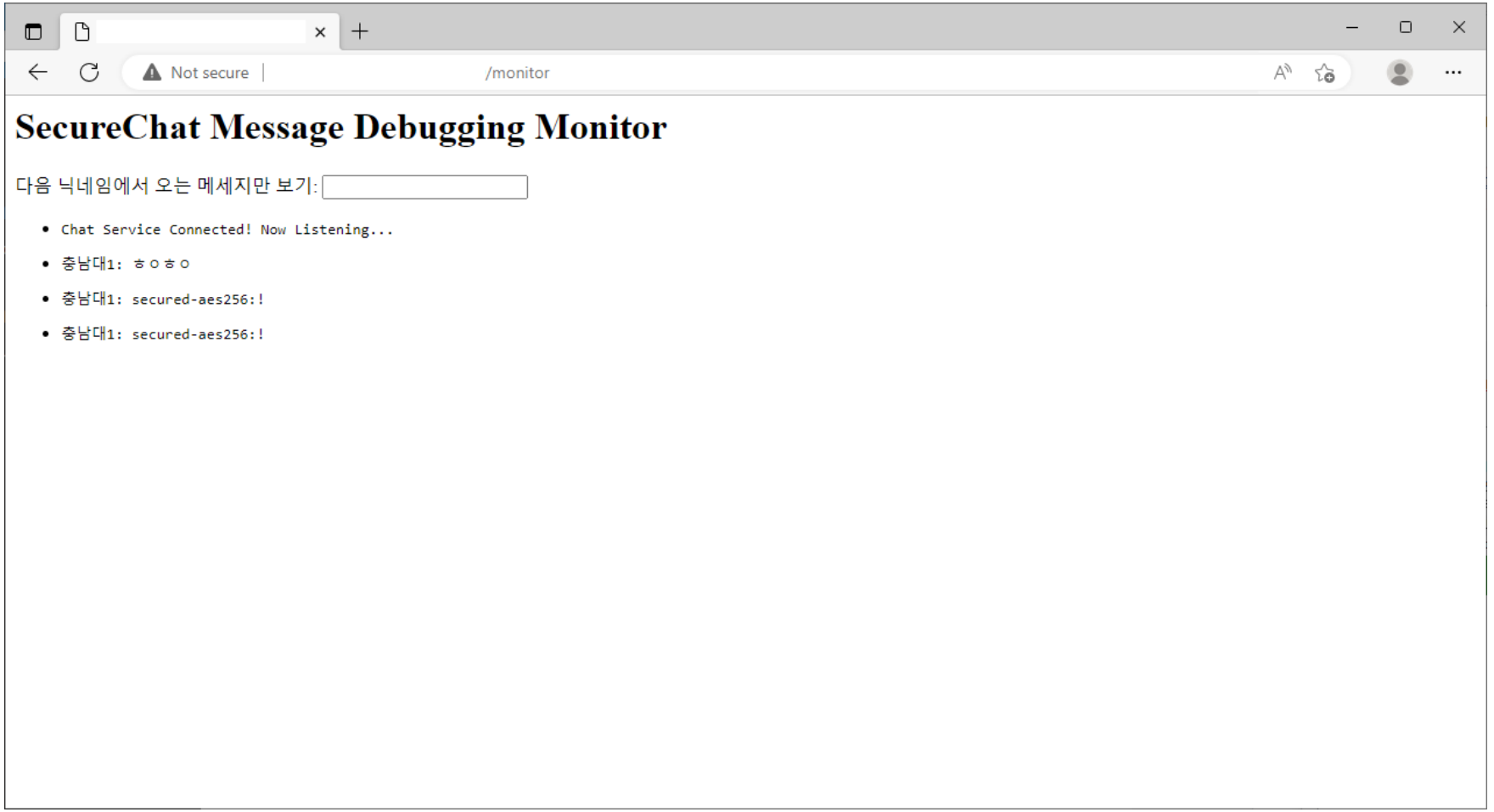

encrypt-message.py - python-scripts - Visual Studio Code
EXPLORER
PYTHON-SCRIPTS
decrypt-message... 1
decrypt-secret.py 1
encrypt-message... 2
encrypt-secret.py 1
generate-keys.py 4
sign-message.py 1
verify-message.py 1
encrypt-message.py \ 2
decrypt-message.py 1
encrypt-message.py
encrypt-message.py > pad_message
from Crypto.Util.Padding import pad
from Crypto.Random import get_random_bytes
from Crypto.Cipher import AES
import base64
def decode_base64(b64):
return base64.b64decode(b64)
def encode_base64(p):
return base64.b64encode(p).decode('ascii')
def read_from_base64():
return [decode_base64(input()), input()]
def pad_message(msg):
padded_msg = msg.encode('utf-8') # 메세지 패딩 구현
return padded_msg
def encrypt_message(key, iv, msg):
# AES 256 암호화 구현
[secretkey, message] = read_from_base64()
message = pad_message(message)
randomiv = # 16바이트 (128비트 IV 랜덤 생성)
randomiv_str = encode_base64(randomiv)
cipher_str = encrypt_message(secretkey, randomiv, message)
print(randomiv_str + '!' + cipher_str)
OUTLINE
TIMELINE
Ln 16, Col 49 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit (microsoft store) Go Live


generate-keys.py - python-scripts - Visual Studio Code
EXPLORER
PYTHON-SCRIPTS
decrypt-message... 1
decrypt-secret.py 1
encrypt-message... 2
encrypt-secret.py 1
generate-keys.py 4
sign-message.py 1
verify-message.py 1
generate-keys.py 4
encrypt-secret.py 1
decrypt-secret.py 1
encrypt-message.py
generate-keys.py > ...
from Crypto import Random
from Crypto.PublicKey import RSA
import base64
def encode_base64(p):
return base64.b64encode(p).decode('ascii')
secret = # 32바이트 (256비트) 랜덤 비밀키 생성
rsa = # RSA 2048 키 생성 시작
pubkey = # 공개키 export
prikey = # 개인키 export
print(encode_base64(secret) + '\n')
print(encode_base64(pubkey) + '\n')
print(encode_base64(prikey) + '\n')
OUTLINE
TIMELINE
Ln 15, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit (microsoft store) Go Live


verify-message.py - python-scripts - Visual Studio Code
EXPLORER
PYTHON-SCRIPTS
decrypt-message... 1
decrypt-secret.py 1
encrypt-message... 2
encrypt-secret.py 1
generate-keys.py 4
sign-message.py 1
verify-message.py 1
verify-message.py 1
generate-keys.py 4
encrypt-secret.py 1
decrypt-secret.py 1
verify-message.py > make_message_hash
from Crypto import Random
from Crypto.Signature import pkcs1_15
from Crypto.PublicKey import RSA
from Crypto.Hash import SHA256
import base64
def decode_base64(b64):
return base64.b64decode(b64)
def encode_base64(p):
return base64.b64encode(p).decode('ascii')
def make_message_hash(msg):
return SHA256.new(msg.encode('utf-8'))
def read_from_base64():
return [input(), decode_base64(input()), decode_base64(input())]
# https://pycryptodome.readthedocs.io/en/latest/src/signature/pkcs1_v1_5.html
def verify(msg, key, signature):
# PKCS #1 v1.5 를 이용한 전자서명 검증, 성공시 "ok" 리턴
[msg, pubkey, signature] = read_from_base64()
verify_result = verify(msg, pubkey, signature)
print(verify_result)
OUTLINE
TIMELINE
Ln 14, Col 43 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit (microsoft store) Go Live

암호채팅 프로그램…

- (채팅 서버 주소)/monitor → 실시간 메세지 송수신 조회
  - 채팅 서버 접속 주소는 수업시간 중 및 SMS 통해 전달



이후 일정

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~~	~~사설인증서 생성 및 프로그램 코드 서명~~	~~11/9~~
11	하이브리드 암호	하이브리드 암호 기반 암호 통신기 (시작)	11/16
12	난수와 디지털서명	하이브리드 암호 기반 암호 통신기 (2)	11/23
13	메세지 인증	하이브리드 암호 기반 암호 통신기 (3)	11/30
14	통신기 검증 & TLS와 PGP(GPG)	GPG를 이용하여 암호 메일 보내기	12/7
15	기 말 고 사 (12/12)		종강

질문?

- 없으면 자리에서 일어나셔도 좋습니다 :)
- 대학원 입학 문의는 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)