Assignment2: Bitcoin Transactions

컴퓨터공학과 2271104 최수희

1. 개요 (Introduction)

본 보고서는 DSA(Digital Signature Algorithm)를 활용하여 비트코인의 기본 거래 유형인 P2PK(Pay-to-PubKey)와 P2PKH(Pay-to-PubKey-Hash)의 스크립트 기반의 거래 로직을 구현하고 검증하는 과정을 다룹니다. 과제의 목표는 암호화 원리를 이해하고, ScriptSig와 ScriptPubKey의 구성 및 실행을 통해 거래의 유효성을 입증하는 것 입니다.

2. 프로그램 실행 요구 사항 (Execution Requirments)

2.1 개발 환경 및 종속성

- Python 버전 : python 3.8 이상 (3.10 또는 3.11)
- 주요 라이브러리 : `cryptography` (DSA 키 생성 및 디지털 서명 구현에 사용됩니다.)
- 추가 라이브러리 : `hashlib` (표준 SHA-1 해시 함수 사용을 위해 Python 기본 모듈을 사용합니다.)
- 패키지 설치 : 활성화된 가상 환경 내에서 다음 명령으로 `cryptography` 를 설치해야 합니다.
  - `pip install cryptography`

2.2. 실행 방법

프로그램은 Task 별로 기능이 분리된 3개의 Python 파일로 구성되며, Task1 파일은 나머지 파일에 키 객체를 제공하는 모듈 역할을 합니다.

a. 파일 구성 : 다음 3개의 `.py` 파일이 같은 디렉토리에 위치해야 한다.
   - `dsa_key_generation.py` : (Task1) 키 생성 및 객체 정의
   - `p2pk_implementation.py` : (Task2) P2PK 구현 및 실행
   - `p2pkh_implementation.py` : (Task3) P2PKH 구현 및 실행

b. 실행 명령어 : 가상 환경이 활성화된 터미널에서 Task2와 Task3 파일만 순차적으로 실행하면 됩니다. 각 파일은 `dsa_key_generation.py` 를 임포트하여 필요한 키 객체를 가져오고, 스크립트를 실행한다.

```
python p2pk_implementation.py
python p2pkh_implementation.py
```

3. 프로그램 예상 결과 및 검증 기준 (Expected Outcome)

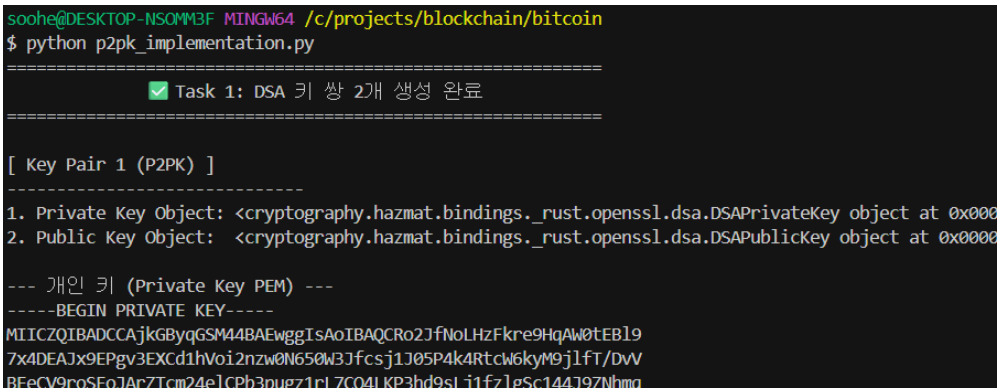
프로그램의 검증 기준은 모든 스크립트 실행의 최종 결과가 True를 반환하는 것 입니다. 이는 ScriptSig가 ScriptPubKey의 요구 조건을 성공적으로 만족시켜 거래가 유효하다는 것을 의미합니다.

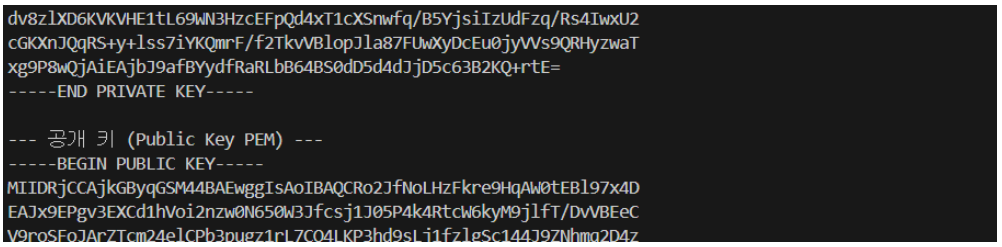
Task	목표	예상 출력	검증 기준
1. DSA 키 쌍 생성	P2PK 및 P2PKH에 사용할 DSA 키 쌍 2개 생성	두 쌍의 Private Key PEM 및 Public Key PEM 문자열	키 객체(`DSAPrivateKey` , `DSAPublicKey`)가 성공적으로 생성되었는지 확인
2. P2PK 구현	"Blockchain Application Q1" 메시지에 대한 서명 및 P2PK 스크립트 실행	`✅ OP_CHECKSIG 결과: TRUE (거래 성공!)`	최종 스크립트 실행 결과가 `True` 인지 확인
3. P2PKH 구현	"Blockchain Application Q2"메 시지에 대한 서명,	`✅ OP_EQUALVERIFY 성공: Public Key Hash 일치 확인.`	OP_EQUALVERIFY와 OP_CHECKSIG가 모두 성

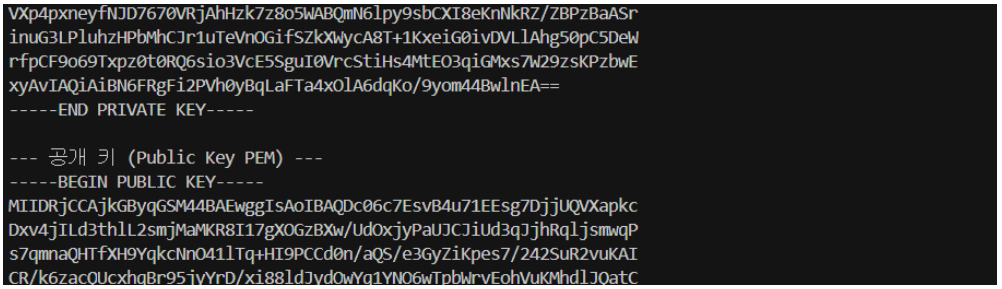
	SHA-1 해시 적용 및 P2PKH 스크립트 실행	및 ✅ OP_CHECKSIG 결과: TRUE (거래 성공!)	공하여 최종 결과가 True 인지 확인

Task2 실행

- 메시지를 SHA-256 알고리즘을 사용하여 개인 키로 서명하는 과정이 성공적으로 완료 (Task1)
- ScriptSig와 ScriptPubKey가 결합된 완전한 스크립트가 실행되어 서명 검증에 성공
- P2PK 거래의 유효성이 최종적으로 확인





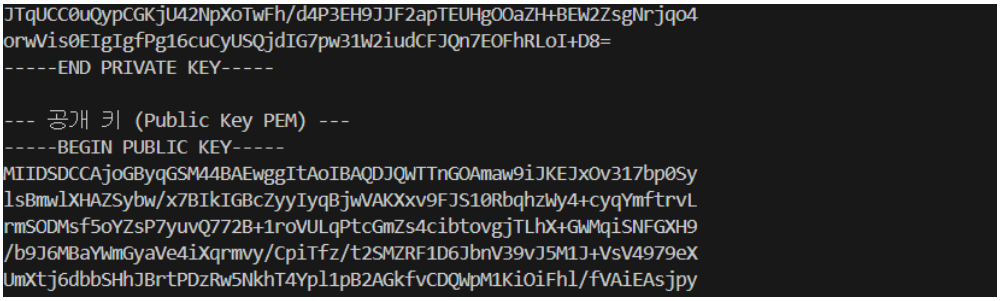


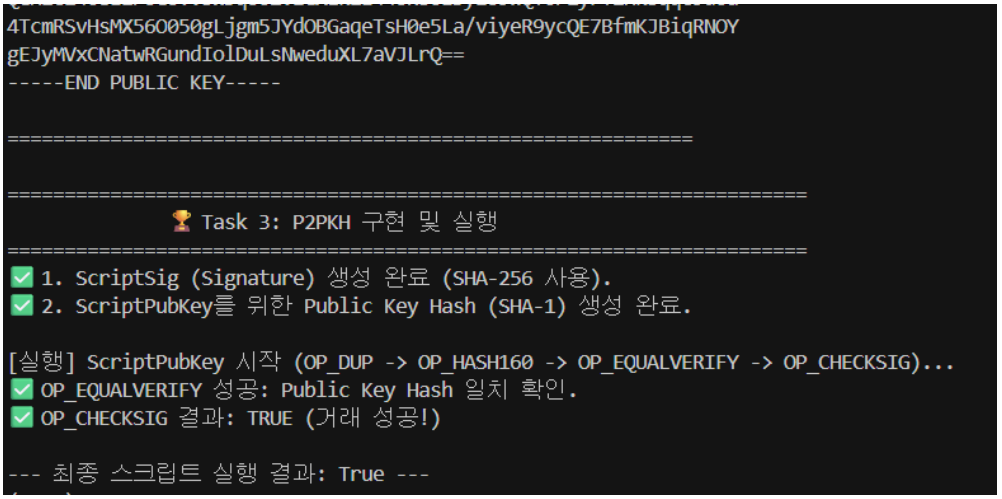
```
OrRp4PinG7NSU7jQQ2gogAAGfDqgKRyrcr8X5HjVTTnNxXzXEZ9xfciPHC8BPzpU
ztmu1juoltw/Zg+6+8xc5F2HbKDwi5SUx6riwJBaipbpWWYe1hOOfFMUkyP6vRpd
We7BNXB3fN2JiIzwJBncCCfBMJLHIhYI7t0=
-----END PUBLIC KEY-----

============================================================


============================================================
              🚀 Task 2: P2PK 구현 및 실행
============================================================
✅ 1. ScriptSig (Signature) 생성 완료 (SHA-256 사용).
✅ 2. ScriptSig 및 ScriptPubKey 구성 완료.

[실행] 완전한 스크립트 구성 및 실행 시작...
✅ OP_CHECKSIG 결과: TRUE (거래 성공!)

--- 최종 스크립트 실행 결과: True ---
```

Task3 실행

- 2가지 DSA 키 쌍 생성 (Task1)
- ScriptSig에서 제출한 공개 키를 SHA-1로 해시한 값이 ScriptPubKey에 미리 잠금 설정된 공개 키 해시 값과 일치함을 확인 (주소 일치 확인 완료)
- 서명과 공개키를 사용하여 서명이 유효함을 확인 (소유자 증명 완료)
- 위 2가지 조건이 모두 충족되어 거래가 유효함이 최종적으로 확인

```
soohe@DESKTOP-NSOMM3F MINGW64 /c/projects/blockchain/bitcoin
$ python p2pkh_implementation.py
============================================================
              ✅ Task 1: DSA 키 쌍 2개 생성 완료
============================================================

[ Key Pair 1 (P2PK) ]
------------------------------
1. Private Key Object: <cryptography.hazmat.bindings._rust.openssl.dsa.DSAPrivateKey object at 0x0000012BA9496
2. Public Key Object:  <cryptography.hazmat.bindings._rust.openssl.dsa.DSAPublicKey object at 0x0000012BA94901

--- 개인 키 (Private Key PEM) ---
-----BEGIN PRIVATE KEY-----
MIICZAIBADCCAjkGByqGSM44BAEwggIsAoIBAQCUokROcB3XCz+2DF718nmnsD6y
aG6rsLblKtz9Go4ViBLEAwtt12wq3WFlwFgcETXRfDZ9HNVBae1k52BP36vi1tUs
GwKJmQQiAiAC/AiHGRlRy9O9LApQTVBqJpDwKNl+QIN4qrT37sTgsA==
-----END PRIVATE KEY-----

--- 공개 키 (Public Key PEM) ---
-----BEGIN PUBLIC KEY-----
MIIDRjCCAjkGByqGSM44BAEwggIsAoIBAQCUokROcB3XCz+2DF718nmnsD6yaG6r
sLblKtz9Go4ViBLEAwtt12wq3WFlwFgcETXRfDZ9HNVBae1k52BP36vi1tUsdyoU
JmMZdPUziazqOv/UlCFBsm1zqNKI9orXZdCW0HQrdb+orzRKTROTJBiUKBs1QXXw
MjpKOEYjoRsze8PV3TPv+oJezQNWKbtMirVTT31+ifKxDeayfLMwKBJaSMfXxyTR
fKZXQdXE2jAf96w3gR1eU7Owin8zhWgrLZxDmGxEadnSHP23Mw8WuSLS3WgVzLqf
cmtCd3UQ/M48nlyK5mP2gPoArxzgiFbdpkkhpiE2kU6Lkrt3QuZyO5dpAiEAiSv8
------------------------------------------------------------

[ Key Pair 2 (P2PKH) ]
------------------------------
1. Private Key Object: <cryptography.hazmat.bindings._rust.openssl.dsa.DSAPrivateKey object at 0x0000012BA9490070
2. Public Key Object:  <cryptography.hazmat.bindings._rust.openssl.dsa.DSAPublicKey object at 0x0000012BA94900B0>

--- 개인 키 (Private Key PEM) ---
-----BEGIN PRIVATE KEY-----
MIICZQIBADCCAjoGByqGSM44BAEwggItAoIBAQDJQWTTnGOAmaw9iJKEJxOv317b
```





4. 결론 (Conclusion)

본 과제를 통해 DSA 알고리즘을 사용하여 개인 키와 공개 키 쌍을 생성하는 과정을 이해했습니다. 또한, 비트코인 거래의 가장 기본적인 두 가지 스크립트 유형인 P2PK와 P2PKH를 Python 코드로 구현하고, 디지털 서명(SHA-256)과 공개 키 해시(SHA-1)를 통한 스택 기반의 검증 로직을 성공적으로 시뮬레이션했습니다. 모든 Task의 스크립트가 유효성 검사를 통과하고 True를 반환함으로써, 비트코인 거래의 잠금(ScriptPubKey) 및 해제(ScriptSig) 매커니즘을 성공적으로 입증하였습니다.