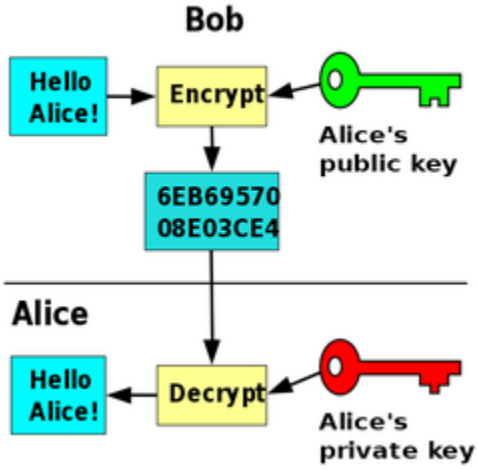
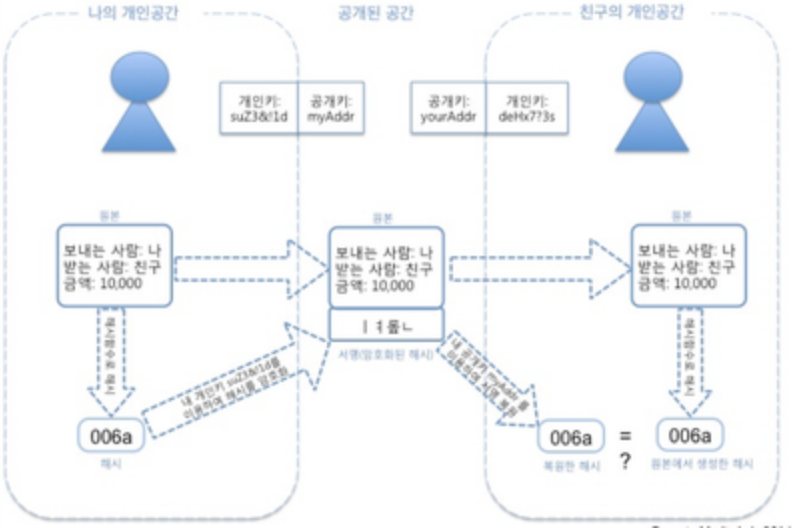
5. 거래

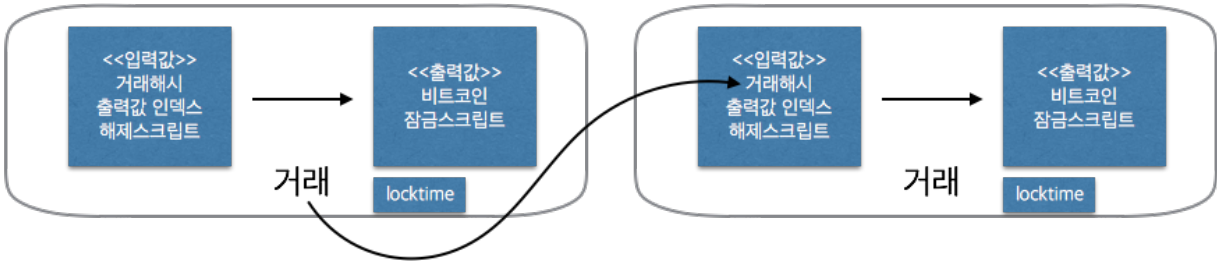
- 먼저 알아야 할 것들
    - 공개키 암호화



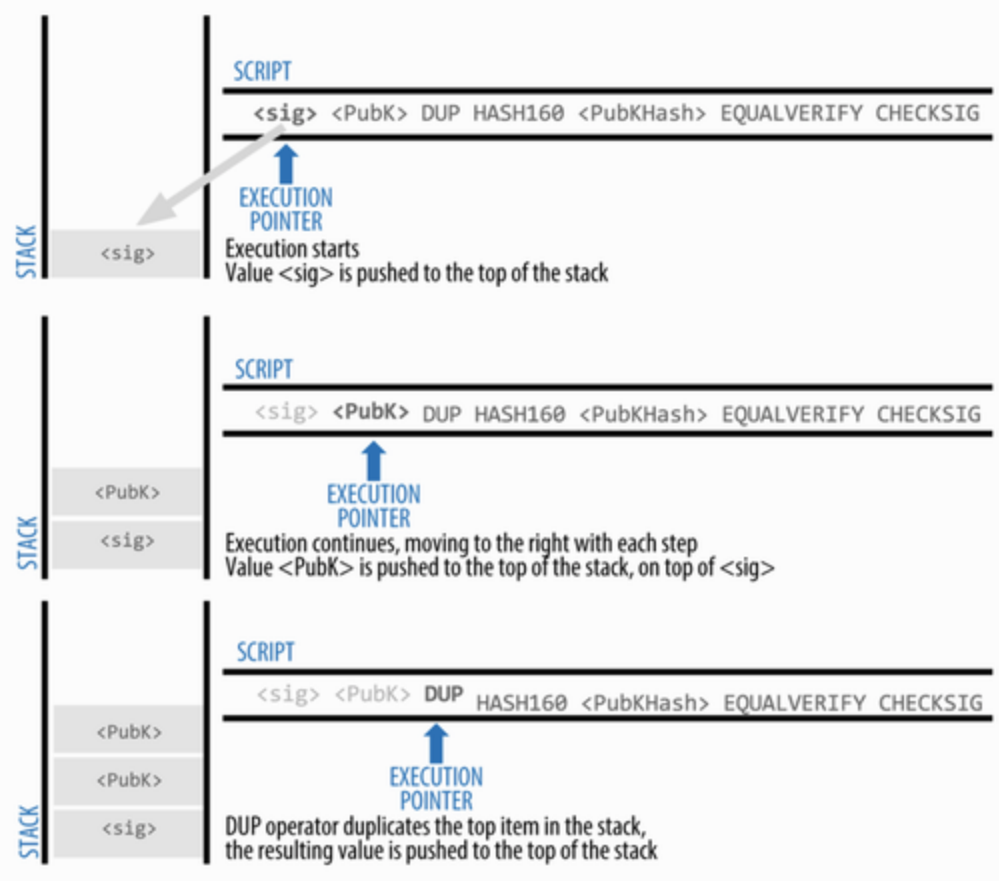
    -
    - Alice의 비밀키와 이에 대응되는 공개키가 있을 때, Alice는 비밀키를 본인만 가지고 있고, 공개키를 말 그대로 공개해서 누구든 볼 수 있게 한다.
    - Bob이 Alice에게 암호화된 문서를 보내고 싶다면, Alice의 공개키로 암호화 하고, Alice는 비밀키로 복호화 할 수 있다.
- 디지털 서명

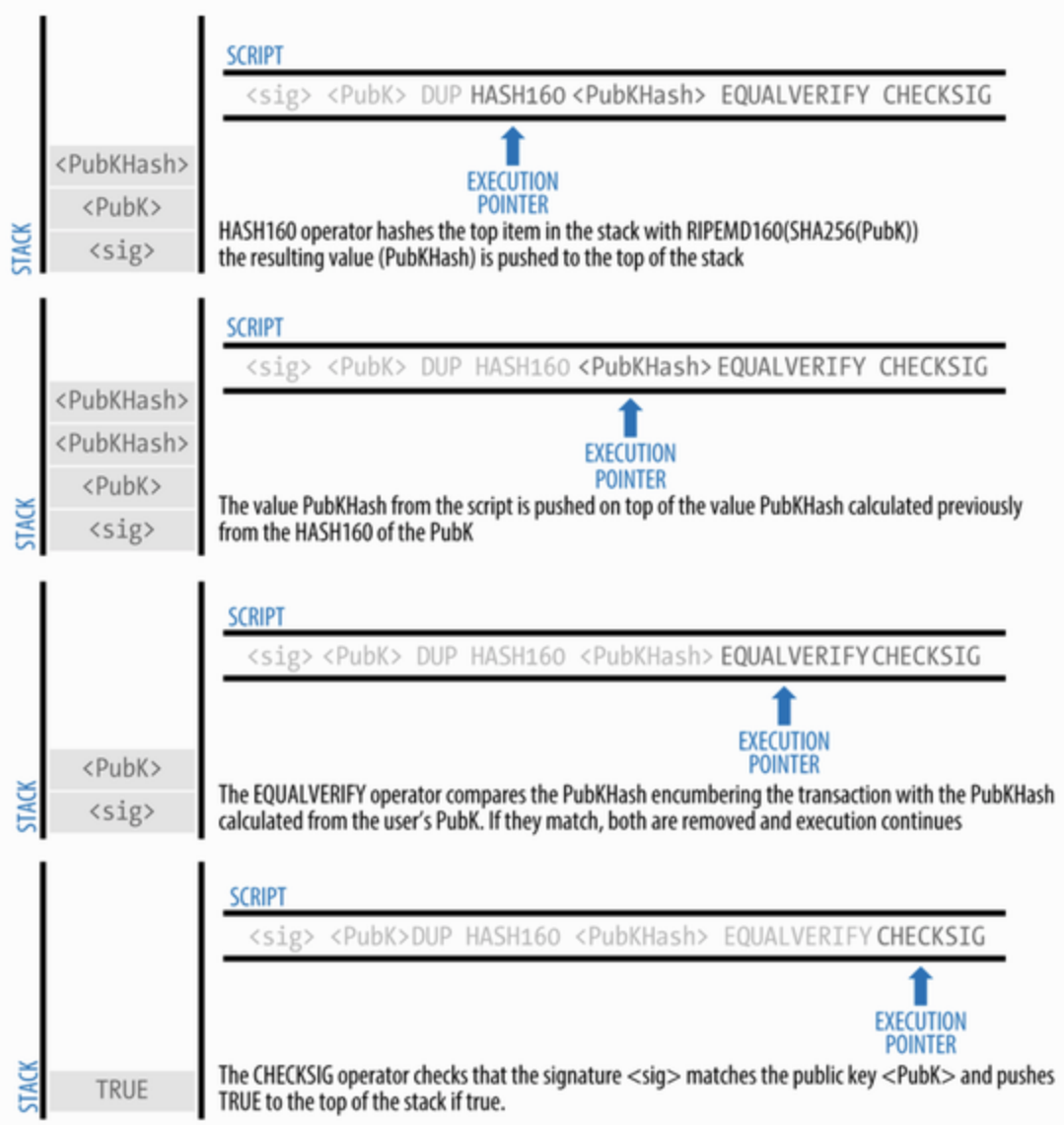


    -
    - 원본 데이터의 해시를 개인키로 암호화(서명) 하여 데이터와 함께 전송
    - 수신자는 송신자의 공개키로 서명을 복호화 한 후에 데이터의 해시 값과 비교하여 검증
- 거래 구조 - 입력값과 출력값



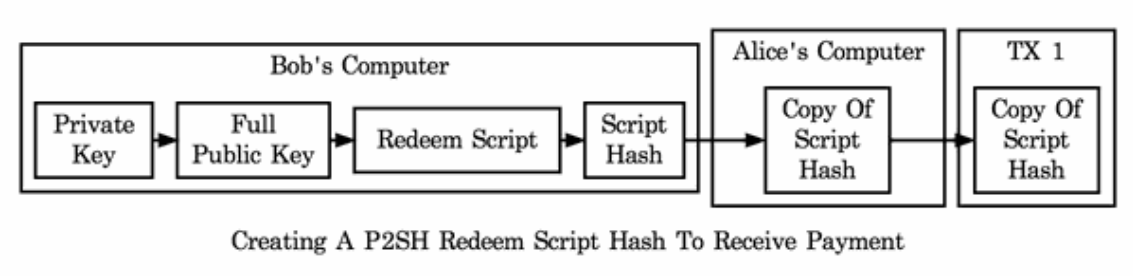
    -
    - 잠금 스크립트 - 수신자의 공개키 해시를 담고 있다.
    - 해제 스크립트 - 송신자의 개인키로 만든 디지털 서명과 송신자의 공개키를 담고 있다.
        - 디지털 서명 - 입력값 데이터(거래해시 + 출력값 인덱스 즉, UTXO)의 해시값 + 수신자의 공개키해시를 송신자의 개인키로 암호화
- 거래 사슬과 고아 거래
    - 거래는 이전 거래를 참조하는 사슬 구조로 이루어져 있다.
    - 부모거래보다 자식거래가 네트워크 상에서 먼저 도착한 경우를 '고아거래'라고 하며, 고아거래 풀에 저장된다.
    - 고아거래는 부모거래가 노드에 도착하면 유효화되고, 거래풀로 옮겨져서 블록에서 채굴될 준비를 마친다.

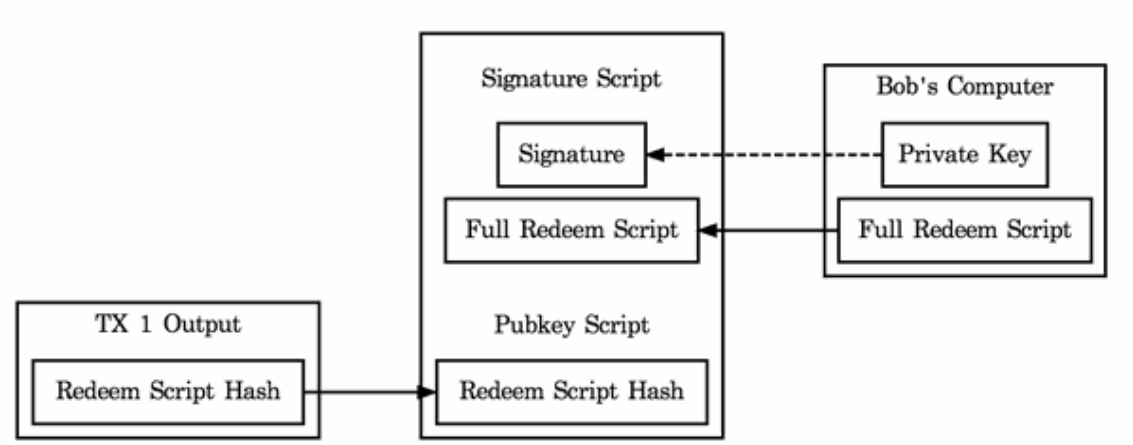
- 거래 스크립트
  - 상태가 없으며 루프를 만들 수 없는 튜링 불완전한 스택 기반 언어를 사용한다.
  - P2PKH(Pay-to-Public-Key-Hash)
    - 해제 스크립트 - <sig> <PubK>
    - 잠금 스크립트 - DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG
    - 



-
- 잠금 스크립트 만들기 (잠금스크립트 만들기와 해제 스크립트 만들기의 '수신자'의 의미는 다름. 앞의 수신자가 뒤의 송신자가 됨)
    - DUP HASH160 - 앞에 수신자의 공개키가 올걸로 가정하고, base58Check 인코딩되지 않은 비트코인 주소를 만들기 위한 코드
    - <PubkHash> - 수신자의 비트코인 주소에서 base58Check 디코딩하여 구함
    - EQUALVERIFY - 앞의 두 값이 같은지 확인하기 위한 코드
    - 위 코드들은 어느 '계좌'로 부터 입력값을 만드는지 검증하기 위한 코드
    - CHECKSIG - 앞에 <sig> <PubK> 가 올 것으로 가정하고, 해제 스크립트를 검증하기 위한 코드
- 해제 스크립트 만들기
    - <sig> - UTXO(거래 아이디, 출력값 인덱스)의 해시에 수신자의 공개키를 붙여서 송신자의 개인키를 이용하여 암호화
    - <PubK> - 송신자의 공개키
    - 위 둘은 잠금스크립트와 붙여져서, CHECKSIG 를 통해서 검증됨. 즉, <sig>를 PubK를 이용하여 복호화 해보면, UTXO의 해시와 수신자의 공개키를 얻을 수 있고, 이 값이 거래내역과 같은지 확인할 수 있음.

- P2PK(Pay-to-Public-Key)
    - 잠금 스크립트
        - <PubK> OP_CHECKSIG
    - 해제 스크립트
        - <sig>
    - P2PKH와 비교해보면, P2PKH는 입력값의 출처를 검증하기 위한 과정이 있는 반면에, P2PK는 이 과정은 생략하고 디지털 서명 과정만 구현함
- Multi-Signature
    - OP_0 <Signature B> <Signature C> 2 <Public Key A> <Public Key B> <Public Key C> 3 OP_CHECKMULTISIG
    - 해제 스크립트 잠금 스크립트
    - 3개 중에 2개의 서명이 있는가 검증
- OP_RETURN
    - OP_RETURN <data>
    - 블록체인 내에 데이터를 기록하기 위한 용도

    - 금액이 0인 출력값이며, UTXO 풀에 있을 필요가 없다.
    - 이 출력값이 입력값으로 사용되면 해당 거래는 유효하지 않다.
- P2SH(Pay-to-Script-Hash)
    - 멀티 시그니쳐 방식의 잠금 스크립트는 그 길이가 길고 복잡하여 사용이 용이하지 않다.
    - 이를 한번 더 해싱한 리딤 스크립트를 이용하여 잠금 스크립트로 만든다.
    - 잠금 스크립트 - OP_HASH160 <20-byte hash of redeem script> OP_EQUAL
    - 해제 스크립트 - sig1 sig2 <redeem script>
    - 리딤 스크립트 - 2 PubK1 PubK2 PubK3 PubK4 PubK5 5 OPCHECKMULTISIG
    - 수신자가 스크립트에 대한 부담을 가진다
    - 

Creating A P2SH Redeem Script Hash To Receive Payment
    - 

Spending A P2SH Output