ECE8930 Blockchain

Homework 1

Biyang Fu - biyangf@clemson.edu

June 2020

1. What security guarantees can the blockchain provide? Explain how the blockchain data structure enforces these guarantees.

1) Hash uniqueness
In blockchain, each block corresponds to a hash, and each hash is calculated by sha256 from the block header. Because the block header contains the hash of the current block body and the hash of the previous block, if the content of the current block changes or the hash of the previous block changes, it will definitely cause the hash of the current block to change. If someone modifies a block, the hash of the block changes. In order for the following blocks to be connected to it, the person must modify all the following blocks at the same time, otherwise the changed blocks will leave the blockchain. Due to the high computational power requirement of block calculation, it is almost impossible to modify multiple blocks at the same time.
Due to this linkage mechanism, the blockchain guarantees its own reliability, and once data is written, it cannot be tampered with. It's like history, what happened is what happened, and can't be changed since then, ensuring the uniqueness of the data.
2) Cryptographic security
Taking Bitcoin as an example, digital currency uses asymmetric encryption, all data storage and records have digital signatures as credentials, and asymmetric encryption guarantees the reliability of payment.
3) Authentication
In the process of digital currency transactions, data transfer from one address to another address will verify it:
-Hash of the last transaction (verification of origin of currency)
-The addresses of both parties in this transaction
-The public key of the payer
-Digital signature generated by the private key of the payment method
The following steps are required to verify the success of the transaction:
-Find the source of the last transaction confirmation currency
-Calculate the fingerprint of the other party's public key and compare it with its address to ensure the authenticity of the public key
-Use the public key to unlock the digital signature to ensure the authenticity of the private key

4) Decentralized distributed design
For the blockchain, the account book data is all or part of the public, emphasizing that there are multiple copies of the account book data, there can be no risk of data loss, the current solution adopted by the blockchain is fully distributed storage, there are many in the network full nodes, synchronize all ledger data (some synchronization parts, of course, there are enough copies of each data storage), so that there are enough copies in the network to meet the requirements of high availability, and the risk of losing data will be much lower. Therefore, it is recommended that when deploying a blockchain network, all nodes are dispersed as much as possible, scattered in different geographic locations, different basic service providers, and different stakeholders.
5) Transmission security
During the transmission process, the data has not been persisted. This part of the air data will be processed using HTTP+SSL (also using websocket+websocketS) to ensure that the data is tamper-proof and encrypted during network transmission.

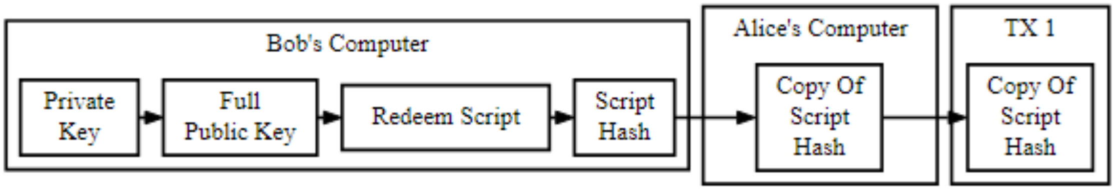
2. Data on the blockchain is secure and private. Explain what is true in this statement. Explain what is not true in this statement.

Blockchain is a distributed ledger technology with the characteristics of decentralization, security, credibility, tamper resistance and programmable. The transparency of the blockchain system makes user transaction privacy and account privacy seriously threatened, and blockchain based on encryption protocols, consensus mechanisms, mixed currencies, zero-knowledge proofs and other technologies are commonly used solutions in the field of data privacy protection. Keys, consensus proofs, mixed currency protocols, etc. encrypt and protect user account data and transaction data to ensure the security and privacy of user data.
Since the credit of the other party is not required, transactions can be made anonymously, and the transaction data can also be encrypted, so the security and privacy of the transaction data can be guaranteed. Only 51% of the nodes of the entire network can be tampered with to alter the ledger. The cost of fraud is extremely high and requires a lot of computing power. However, it still has its security limitations. A large number of nodes need to be introduced. If there are too few nodes, 51% of the nodes are easy to attack. The private key is stored locally on the user terminal, and if the private key is stolen, there will be a loss of funds. Whether the consensus mechanism of PoW and PoS is truly safe, there is a lack of rigorous proofs and tests.

3. We explained BTC P2PKH transactions. Research Pay to Script Hash P2SH transactions. Explain how they work. Give some positive and negative aspects of P2SH.

P2SH (Pay to Script Hash)
An important feature of P2SH is that it can compile script hashes into an address. The P2SH address is a script with a hash of 20 bytes based on Base58 encoding, just as the

bitcoin address is a 20-byte public key based on Base58 encoding.
Since P2SH addresses are prefixed with 5, this leads to addresses based on Base58 encoding starting with "3". The P2SH address hides all the complexity, so people who use it to make payments will not see the script.



In the example, Bob generated a redemption script, hashed the redemption script to generate a redemption script hash, and then provided the hash to Alice. Alice can then create a P2SH-style output that contains the hash of Bob's redemption script.
Advantages:

- In the transaction output, the complex script is replaced by a short electronic fingerprint, which makes the transaction code shorter.
- The script can be compiled into an address, and the Bitcoin wallet of the issuer of the payment instruction and the payer can execute P2SH without complicated processes.
- P2SH shifts the burden of building scripts to the receiver, not the sender.
- P2SH shifts the burden of long script data storage from the output side (stored in the UTXO set, affecting memory) to the input side (stored in the blockchain).
- P2SH shifts the burden of long script data storage from the current (when paying) to the future (when spending).
- P2SH transfers the transaction fee cost of the long script from the sender to the receiver. The receiver must include the redemption script when using the funds.

Disadvantages:
P2SH is the most basic script structure of Bitcoin. It shows the content of the entire script, including the information of the public key and private key signatures. The node uses this information to verify the transaction. But at the same time, the node and anyone can get all the details of the transaction.
So, we say that Bitcoin is transparent. Although it can isolate the physical entity's individual from the bitcoin account and achieve the anonymity of the physical individual, but from the perspective of the bitcoin account, P2SH does not provide privacy for the account, everyone can know which fund is / Which accounts are used in what way.

4. Explain how P2PKH works. Be specific about how P2PKH guarantees that payment goes only to the intended recipient.

PKH, the public key hash value. Most transactions on the Bitcoin network are P2PKH transactions. Such transactions contain a locking script that uses a public key hash to prevent the output function. The public key hash is the well-known bitcoin address. The output locked by the P2PKH script can be unlocked by typing the public key and the digital signature created by the corresponding private key.
Examples of P2PKH:
In one transaction, Bob paid Alice 0.15 BTC. Since there is no concept of an account in Bitcoin, the output of this transaction does not include Alice’s name or Alice’s public key,

but the hash value of Alice's public key. In this way, the privacy of the user is further ensured. Alice wants to spend 0.15 BTC. How should she prove that she owns this UTXO, and that no one else can fake Alice to spend this UTXO?

The answer is that the output created by Bitcoin's transaction is not a simple public key address, but a script. In this transaction where Bob pays Alice 0.15 BTC, the output script created by Bob is similar:

OP_DUP OP_HASH160 <Alice Public Key Hash> OP_EQUAL OP_CHECKSIG

The meaning of this script is that whoever can provide a signature and a public key and let this script run through, who can spend 0.15 BTC of this transaction. Since the signature can only be created using Alice's private key, signatures created by non-Alice's private key will not pass the verification of this script, so no one else can impersonate Alice to spend the transaction.

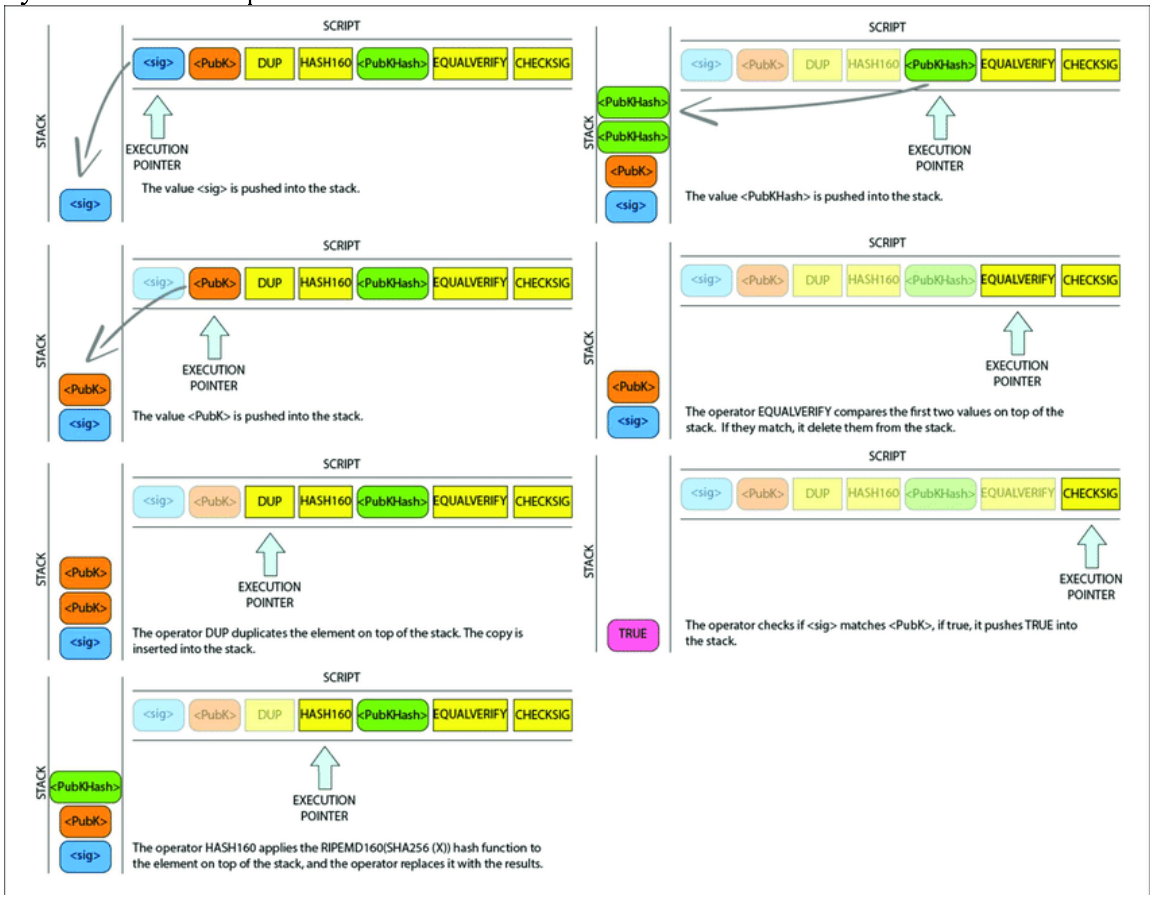
The unlock script of the lock script is:

<Alice Signature> <Alice Public Key>

Combining the two scripts can form the following effective combination script:

<Alice Signature> <Alice Public Key> OP_DUP OP_HASH160 <Alice Public Key Hash> OP_EQUAL OP_CHECKSIG

The figure below shows the step-by-step process of verifying the validity of a transaction by a combined script.



The value <sig> is pushed into the stack.

The value <PubK> is pushed into the stack.

The operator DUP duplicates the element on top of the stack. The copy is inserted into the stack.

The operator HASH160 applies the RIPEMD160(SHA256 (X)) hash function to the element on top of the stack, and the operator replaces it with the results.

The value <PubKHash> is pushed into the stack.

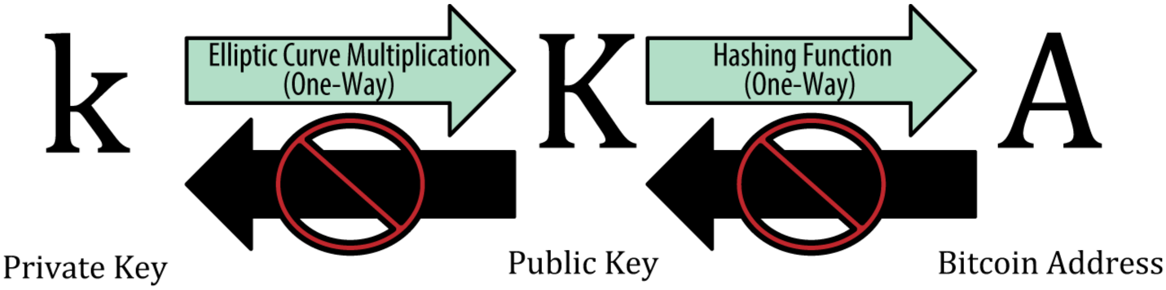
The operator EQUALVERIFY compares the first two values on top of the stack. If they match, it delete them from the stack.

The operator checks if <sig> matches <PubK>, if true, it pushes TRUE into the stack.

Only when the set conditions of the unlocked version script and the locked version script match, the result will be displayed as true when the combined effective script is executed. That is, only when the unlocking script obtains Alice's valid signature, the transaction execution result will be passed (the result is true). The valid signature is obtained from Alice's private key that matches the public key hash.

5. I set up a BTC account by generating an EC public key and private key pair. Assuming that the software has no bugs, how likely is it that someone else could redeem my coins? Compare that likelihood to the likelihood of that person winning the powerball lottery, winning the powerball lottery twice, being hit by lightning, and being hit by lightning twice. Is that risk acceptable? Why or why not?

This probability is small enough to be 0.
There is a classic diagram in the book "Mastering Bitcoin" to illustrate the relationship between the private key, public key and bitcoin address. The private key can generate a public key, and the public key can generate a bitcoin address, and vice versa is not feasible.



In the blockchain world, the bitcoin we hold is just a string of private keys and a string of 256-bit binary numbers. If you toss a coin, write 0 on the front and 1 on the reverse, toss it 256 times in a row, record it, and then convert this string of binary values to hexadecimal numbers, all your possessions are on this string of private keys too. Creating a private key is essentially "taking a number between 1 and 2 to the 256th power".
2 to the 256th power is an unimaginable large number, expressed in decimal, it is about 10 to the 77th power, and the visible part of the universe, its composition is about 10 to the 80th power. Therefore, the probability of randomly hitting the same private key can be small enough to be ignored. The possibility that any two randomly generated addresses are exactly the same is equivalent to 2 people picking up the same grain of sand from the earth, and seeing this grain of sand as the earth, then 2 people picking up the same grain of sand again from the earth.
In wallets such as Bitcoin Core, the random number generation algorithm is already very professional, and the probability of coincidence is 0, which can be used with confidence.