Consensus in Bitcoin and Bitcoin Script

1. The timestamp and difficulty fields are part of the header of a Bitcoin block. How are these values related?

> **Solution:**
> The timestamp is one of the values required to govern Bitcoins mining difficulty. Timestamps allow the monitoring of block times, which indicate whether difficulty is currently too low or high.

2. What does probabilistic consensus mean? Can a transaction be reverted?

> **Solution:**
> A probabilistic consensus means that, if a network agrees on a certain set of transactions to be executed, it is not 100% certain to be valid in the long term. Bitcoin and other blockchains which have a probabilistic consensus have no settlement finality directive. This means, it is possible that transactions can be reverted, however with a very low (and decreasing) probability. The reason for such reversal could be a 51%-attack.

3. Name two functions that are fulfilled by the *coinbase* transaction, the first transaction in a Bitcoin block.

> **Solution:**
>
> - It mints new coins providing mining incentive.
> - It can be used for data storage.

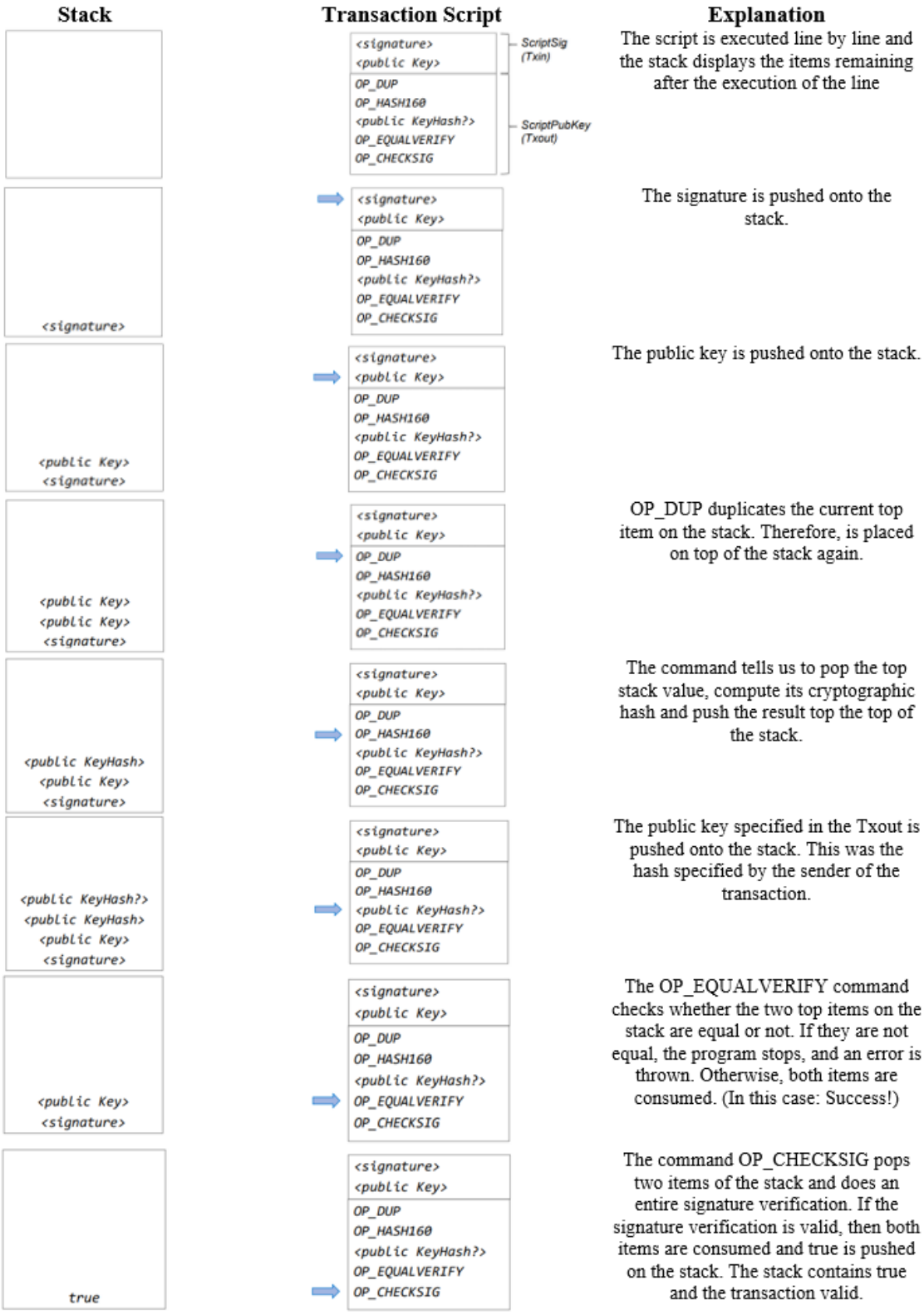
Introduction to Bitcoin Scripts

UTXOs are locked using Bitcoin script, making sure only the intended recipient gets to spend them. The simplest type of script is pay-to-public-key (P2PK). In this case, the receiver must provide the sender with his/her public key. The successor to P2PK is Pay-to-Public-Key-Hash (P2PKH) where the identity is not a public key, but a hash of a public key. A person to redeem the UTXO needs to provide a public key that hashes to the P2PKH and a signature which belongs to this public key.

How does the script work?

- The scriptSig is concatenated with the scriptPubKey and then executed.
- The script runs sequentially on a stack machine. There are no registers and no external memory.
- The script is executed and if the result is true, the UTXO can be spent, otherwise not.

The below figure shows how a script is executed. Refer to this introduction to solve questions 5 and 6. For additional information on Opcodes and Bitcoin script execution, your are kindly referred to the following Bitcoin wiki.


Stack
Transaction Script
Explanation
<signature>
<public Key>
ScriptSig (Txin)
OP_DUP
OP_HASH160
<public KeyHash?>
OP_EQUALVERIFY
OP_CHECKSIG
ScriptPubKey (Txout)
The script is executed line by line and the stack displays the items remaining after the execution of the line
<signature>
The signature is pushed onto the stack.
<public Key>
<signature>
The public key is pushed onto the stack.
<public Key>
<public Key>
<signature>
OP_DUP duplicates the current top item on the stack. Therefore, is placed on top of the stack again.
<public KeyHash>
<public Key>
<signature>
The command tells us to pop the top stack value, compute its cryptographic hash and push the result top the top of the stack.
<public KeyHash?>
<public KeyHash>
<public Key>
<signature>
The public key specified in the Txout is pushed onto the stack. This was the hash specified by the sender of the transaction.
<public Key>
<signature>
The OP_EQUALVERIFY command checks whether the two top items on the stack are equal or not. If they are not equal, the program stops, and an error is thrown. Otherwise, both items are consumed. (In this case: Success!)
true
The command OP_CHECKSIG pops two items of the stack and does an entire signature verification. If the signature verification is valid, then both items are consumed and true is pushed on the stack. The stack contains true and the transaction valid.

4. The following transaction output is provided:

```
OP_DUP OP_HASH160 8a014218a5a42e2c6fc5d573ab54a91ff555d1de OP_EQUALVERIFY OP_CHECKSIG
```

(a) Can you tell which entity has created this transaction output?

> **Solution:** No. As the we only see the transaction output, we only know the receiver of the transaction.

(b) Can you tell if this transaction output is spent?

> **Solution:** No. We do not know if transactions exist which spend the output.

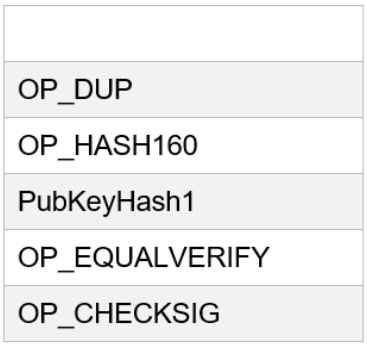
(c) Can you tell which entity is allowed to spend this transaction output?

> **Solution:** The owner of the private key which corresponds to the public key which corresponds to the hash 8a014...1de.

(d) What specific data is required to spend the transaction output?

> **Solution:** The public key of the hash 8a014...1de and the corresponding signature (therefore the private key).

5. Bitcoin script allows to set rules for the spending of Bitcoins. Following script represents a standard Pay-to-public-key-hash (P2PKH) script.



The TxOut-script.

As an input, you would provide the corresponding signature and public key. Out of simplicity and reduced computational effort, Bob removes following codes:

```
OP_DUP OP_HASH160
```

The entity which wants to spend this TxOut-script needs to provide the hash of the public key additionally to the signature and public key. Explain how you would attack this script and steal the funds.

The TxOut-script.

Solution: In this case, the provided public key does not have to correspond to the public key hash. The attacker therefore just could provide the public key hash (the equalverify checks if they are identical) and an arbitrary public key alongside the signature, stealing the funds.