L-Università ta' Malta
Faculty of Information &
Communication Technology

Project Report

Manwel Bugeja

March 2, 2021

Contents

1 Pay-to-Public-Key-Hash (P2PK) 2

1.1 Overview 2

1.2 Problems 2

2 Pay-to-Public-Key-Hash (P2PKH) 2

3 Bitcoin Script Example 1 3

4 Bitcoin Script Example 2 3

1 Pay-to-Public-Key-Hash (P2PK)

1.1 Overview

When using this script, the public key is stored as plaintext within the locking script. This is mostly used in coinbase transactions which was generated by mining software which is not up to date.

The following in an example of a P2PK locking script:

```
<Public Key A> OP_CHECKSIG
```

To unlock this, the following script must be submitted:

```
<Signature from Private Key A>
```

Together, they form the script which is validated transaction validation software. The script becomes:

```
<Signature from Private Key A> <Public Key A>
   OP_CHECKSIG
```

This script uses the CHECKSIG operater. This operator returns TRUE on the stack if the signature belongs to the correct key. [2]

1.2 Problems

The P2PK is longer than P2PKH discussed in section 2. This is because for people to make transactions, the would have to pass the whole publik key instead of the address, which is much longer. [3]

Apart from that, P2PK is also more unsafe than P2PKH. Bitcoin uses the Elliptic Curve Digital Signature Algorithm (ECDA) signature scheme. This means the the bitcoins that use P2PK are only safe until the ECDA is cracked. An algorithm already exists that cracks the ECDA and is called Shor's Algorithm. However, to work effectively, this algorithm needs a quantum computer.

2 Pay-to-Public-Key-Hash (P2PKH)

This script is used for most of the transactions processed on the bitcoin network. Within this script, the locking script encrypts the output with a public key hash. This is known as a bitcoin address. When an output is spent, the lock on the output by a P2PKH is removed. This unlocking is done by handing over a public key and a digital signature that were created by the corresponding private key.

An example of a locking script is:

```
OP_DUP OP_HASH160 <Cafe Public Key Hash> OP_EQUAL
   OP_CHECKSIG
```

The corresponding unlocking script looks like:

```
<Cafe Signature> <Cafe Public Key>
```

The validation scripts of the previous two scripts looks like:

```
<Cafe Signature> <Cafe Public Key> OP_DUP OP_HASH160
    <Cafe Public Key Hash> OP_EQUAL OP_CHECKSIG
```

If the unlocking script has a valid signature that corresponds to the publid key hash set as an encumbrance, the script returns TRUE on the stack. [2]

3 Bitcoin Script Example 1

Question If you were given, 767695935687, which is a binary encoding for a Bitcoin script ScriptPubKey, then what ScriptSig would you need to combine the ScriptPubSig with to execute or unlock the ScriptPubKey? Hint: 76 → OP_DUP

Solution Changing the assembly `767695935687` to bitcoin script we get `OP_DUP OP_DUP OP_MUL OP_ADD OP_6 OP_EQUAL`.

When we translate this to normal algebra we get $(x*x)+x = 6$. Making x subject of the formula:

$$x^2 + x - 6 = 0$$

$$(x+3)(x-2) = 0$$

$$x = -3, x = 2$$

Therefore there are two signatures that will release cash and these are: `OP_2` and `OP_3 OP_NEGATE`

4 Bitcoin Script Example 2

Question Figure out what this script is doing: `6e879169a77ca787`

Solution Translating the assembly to bitcoin script, we get: `OP_2DUP OP_EQUAL OP_NOT OP_VERIFY OP_SHA1 OP_SWAP OP_SHA1 OP_EQUAL`.

Word	**Description**
OP_2DUP	Duplicates the top two stack items.
OP_EQUAL	Returns 1 if the inputs are exactly equal, 0 otherwise.
OP_NOT	If the input is 0 or 1, it is flipped. Otherwise the output will be 0.
OP_VERIFY	Marks transaction as invalid if top stack value is not true. The top stack value is removed.
OP_SHA1	The input is hashed using SHA-1.
OP_SWAP	The top two items on the stack are swapped.
OP_SHA1	The input is hashed using SHA-1.
OP_EQUAL	Returns 1 if the inputs are exactly equal, 0 otherwise.

This script returns TRUE if a hash collision is found and FALSE otherwise. [1]

References

[1] Bitcoin wiki script. https://en.bitcoin.it/wiki/Script, Last accessed on 2021-03-02.

[2] A. M. Antonopoulos. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies.* O'Reilly Media, 2014.

[3] Walker, Greg. P2pk. https://learnmeabitcoin.com/technical/p2pk, Last accessed on 2021-03-0.