By **Patrick** - August 20, 2018

scriptPubKey & scriptSig

Last Updated: 30th October 2018

scriptPubKey is a locking script placed on the output of a Bitcoin transaction that requires certain conditions to be met in order for a recipient to spend his/her bitcoins; scriptPubKey is also known as PubKey Script outside of the Bitcoin code. Conversely, scriptSig is the unlocking script that satisfies the conditions placed on the output by the scriptPubKey, and is what allows it to be spent; outside of code, scriptSig is also known as Signature scripts.

Both scriptPubKey and scriptSig are written in Script, the programming language used for constructing Bitcoin transactions. Script lacks many of the functionalities of present day programming languages, however, this also makes it inherently more secure due to the limited number of operations that it can perform.

Understanding Inputs and Outputs

To understand how scriptPubKey and scriptSig function within in a Bitcoin transaction, the manner in which transactions are constructed must first be understood.

Bitcoin transactions use unspent transaction outputs (UTXOs) from previous transactions as inputs in the construction of a new one. To illustrate, consider that Alice wants to send Bob 1 bitcoin, however, she knows that a transaction fee of 0.25 bitcoins is required. Using previous UTXOs as inputs for her transaction with bob, those inputs could be:

- Input 1 – 0.25 BTC
- Input 2 – 0.25 BTC
- Input 3 – 0.25 BTC
- Input 4 – 0.25 BTC
- Input 5 – 0.25 BTC

Taking into account the transaction fee of 0.25 BTC, the output of the transaction, i.e. the amount of bitcoins Bob would actually receive, would look like this:

- Output 1 – 0.25 BTC
- Output 2 – 0.25 BTC
- Output 3 – 0.25 BTC
- Output 4 – 0.25 BTC

Bob would therefore receive a single bitcoin at the end of the transaction.

However, in order for Bob to spend his bitcoin, each output that Bob receives will contain a locking script, scriptPubKey, which must first be satisfied by the unlocking script, scriptSig.

To illustrate, when Alice decides to initiate her transaction with Bob, the outputs that Bob receives contain an amount of bitcoins that can be spent only when the conditions laid out by the attached scriptPubKey are satisfied. When Bob decides to spend these outputs, he will create an input that includes an unlocking script, scriptSig, that must satisfy the conditions that Alice placed in the previous outputs before he can spend them.

Source: Mastering Bitcoin

As can be seen from the image, the unlocking script, scriptSig, contains a *sig* and *PubK,* or digital signature and public key, which must be provided in order for the locking script to be satisfied. Conversely, the locking script, scriptPubKey, contains a *PubKHash*, also known as a public key hash, or more simply, a Bitcoin address.

The process works such that, the scriptSig and scriptPubKey are combined and executed in sequence, with the unlocking script being executed first. For example, when Bob decides to spend the 1 bitcoin that he received from Alice, he must first **unlock** the outputs, which then become **locked** when the recipient receives the 1 bitcoin.

This method of executing the unlocking and locking scripts (scriptPubKey and scriptSig) in the Bitcoin core client was changed in 2010 due to a vulnerability that allowed the locking script to be corrupted by the unlocking script. The scripts now operate in such a way wherein they execute separately.

Conclusion

To conclude, scriptPubKey is a locking script used within the Bitcoin core client that places conditions on a transaction output which must be satisfied before it can be spent. The unlocking script, scriptSig, satisfies the conditions placed on the output by scriptPubKey.

Both scripts are written in scripting language, Script.

More information on how scriptPubKey and scriptSig work can be found here:

- Bitcoin Developers Guide – Transactions