COMPARISON OF TRANSACTIONS IN BITCOIN AND GRIN

1. INTRODUCTION

This paper aims to provide a high-level description of transactions in cryptocurrencies and compare their building process and verification techniques. The currencies taken into consideration are Bitcoin and Grin. Bitcoin is considered the pioneer of cryptocurrencies, being the first to be introduced in 2009. [1] It lies on the Bitcoin blockchain where everything is registered on a public ledger, and everyone can create an address generated from a public key without any approval. [2] Grin, on the other hand, is a cryptocurrency that lays on the Mimblewimble protocol where the concept of addresses does not exist, and its transactions are confidential. [3]

2. BITCOIN TRANSACTIONS

2.1. Transactions

Transactions in Bitcoin are data structures that encode the transfer of value between participants in the Bitcoin system. We can consider transactions as lines in a double entry bookkeeping ledger, where inputs stay on the left and act as debits while outputs stay on the right and act as credits. [2]

A transaction uses as input another previous transaction's unspent output. The sender provides the signature that is able to unlock the previous transactions, thus proving to the network the ownership of the funds.

2.2. Transaction Components

Transactions in Bitcoin are composed of two elements: transaction outputs and transaction inputs. Transaction outputs in Bitcoin contain two elements. The first element is the amount of satoshis (the smallest Bitcoin unit) transferred. The second element is the scriptPubKey or the locking script that serves a cryptographic puzzle which specifies the conditions required to spend the output. Usually, it contains the Bitcoin address of the recipient, which consists of the 160-bit hash of the public key. It needs to be specified that each output in a transaction has a different locking script.

Transaction inputs identify by reference which of the unspent transactions controlled by the wallet will be consumed and provide proof of ownership through an unlocking script. An input is constructed of four elements. The first element the transaction ID (txid) references the previous transaction that holds the unspent output, while the second element output index (vout) identifies the output being spent. The third element is the unlocking script (scriptSig) which satisfies the conditions specified in the unspent output by the cryptographic puzzle. Usually, it contains the public key and a digital signature produced by the user's private key. The last element is the sequence number which in the original Bitcoin implementation was intended to provide a type of high frequency trade functionality but has very limited uses nowadays.

Fig 1. Transaction in Bitcoin

Source: www.blockchain.com/explorer

2.3. Transaction Process

The transaction process in Bitcoin passes through several steps. At the beginning the sender starts by choosing the address of the recipient, then identifies the unspent transaction output from which it is going to retrieve the funds. After unlocking the script of the previous output, proving the ownership of the funds, he populates the transaction output part (amount and locking script) signing the entire transaction or parts of it with his private key ends the process with signing the transaction.

2.4. Verification

To verify that the inputs are authorized to collect the funds inside the unspent outputs, Bitcoin uses the scripting system. The language used in the scripts uses stacks as data structures. Stack is a very simple data structure, and its operations are always performed on the item placed on top of the stack.

The locking script for a transaction output comes in the following stack structure with the recipient address and stack operations.

OP_DUP OP_HASH160 <Public Key Hash> OP_EQUALVERIFY OP_CHECKSIG

These operations are described as follows:

- OP_DUP operation adds a duplicate of the topmost item to the stack.
- OP_HASH160 hashes the last element.
- OP_EQUALVERIFY verifies if the last two items are equal and if true pops them out from the stack.
- Op_CHECKSIG checks if the signature matches the public key and pushes TRUE to the top of the stack.

Meanwhile, the unlocking script begins with the signature and public key of the user, which now is considered as the sender since the input is part of a new transaction, and then executes all the operations specified in the locking script. If the result of the unlocking script is TRUE and there are no other elements left in the stack, the input satisfies the conditions specified in the unspent output and, therefore, can spend it.

<Signature> <Public Key> OP_DUP OP_HASH160 <Public Key Hash> OP_EQUALVERIFY OP_CHECKSIG

2.5. Transaction Fees

Another element in a transaction that is not declared but deducted from the difference between the input and output values is the transaction fee. The fee is collected by the mining node which adds the transaction to the blockchain. The fee mechanism serves not only as an incentive for the miner to include the transaction in the block but also as a disincentive for abusers who might want to overload the system with transactions. However, transaction fees are not mandatory and eventually transactions without fees might be processed, even though the chances that this happens are quite low. The fee, however, is not calculated based on the value of the transaction in Bitcoin but on the size of the transaction in kilobytes.

2.6 Common Transaction Forms

There are three common transaction forms in Bitcoin. The most common one is a simple payment from the sender to the recipient, that includes a change output returned to the original owner. In this case the transaction will contain outputs that are addressed to a new owner and an output that is addressed to the same or another address from the owner's wallet. Usually this is called change address. Another form of transaction aggregates several small inputs into a single output. This is the equivalent of paying with pocket change in the real world. Last, the third most common form of transaction disputes one input to multiple outputs, sent to different recipients.

3. GRIN TRANSACTIONS

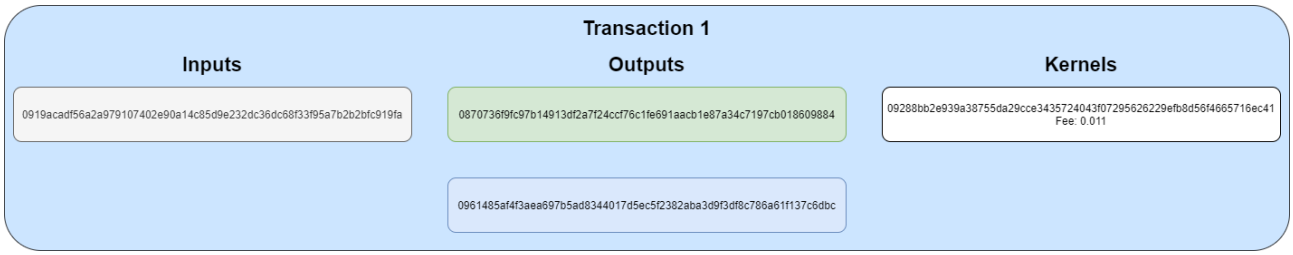
3.1. Transactions

The main goals of Mimblewimble protocol are privacy, lightweightness and strong cryptography. First, the blockchain manages to store anonymity by making transactions indistinguishable. Second, it is lighter compared to other protocols since it follows a cut through and pruning technique, removing from the blockchain all spent outputs leaving only the kernel of the transaction which is roughly 100 kb. For this reason, the blockchain has a rate of growth four time less than the rate of Bitcoin. Finally, it relies on a strong and proven cryptography such as the Elliptic Curve Cryptography. Transactions in Grin are considered interactive, meaning both parties need some communication to interact with each other and exchange the necessary data to create it. The transaction data that is exchanged until the signature is complete is called a slate. [4]

3.2. Transaction Components

Transactions in Grin, same as in Bitcoin are composed by a set of outputs and a set of inputs, that reference and spend a set of UTXOs. In Grin, the form of the output is a Pedersen Commitment and includes a blinding factor, the output value, and proof that the value is positive. Moreover, the transaction includes a third element which is called a transaction kernel. It consists of a kernel excess (the public key corresponding to the excess value), a transaction signature signed by the excess value, and a transaction fee written in cleartext. The transaction kernel proves the authenticity of the transaction, meaning that both parts agreed with it. Furthermore, proves that the difference between the sum of outputs and inputs is null, asserting that no coins were created.

Fig 2. Transaction in Grin



Source: phyro.github.io/what-is-Grin/interactive_txs.html

3.3. Transaction Process

A transaction in Grin starts with the recipient providing a Slatepack address. It should be noted that this type of address is not a normal address. This means that it is not part of the chain, it is only used for supporting peer-to-peer interaction. Then, the sender starts building the transaction slate, encrypts it with the recipient's address (public key), and passes it over to the recipient. Subsequently, the recipient adds his own data and partial signature to the slate and delivers it back to the sender. Finally, the sender can finalize the transaction by adding to the slate the final data and his own partial signature. At this point he can post it to the chain, completing the transaction process. [5]

3.4. Verification

Since Mimblewimble transactions are confidential and do not involve addresses, the only requirement for spending them is the ability to unlock the Pedersen Commitment. To do that the spender should know both the blinding factor and the amount of Grin present in the output. Because it is impossible to deconstruct a commitment and therefore deduce these elements, it is impossible for a stranger to spend it. Not even the sender could spend it, because even though he knows the amount, he does not know the blinding factor which is only known by the recipient.

3.5. Transaction Fees

The transaction fee in Grin is the only element that is written in clear text. Unlike Bitcoin, they are declared and not deducted from the difference between the input and the output. This fee is collected by the mining node which adds the transaction to the blockchain. The fee mechanism serves as an incentive for the miner and as a disincentive for the abusers. Same as in Bitcoin, the fee does not depend on the value of the transaction but on the size and the complexity of it.

3.6 Common Transaction Forms

As in Bitcoin, there are three common transaction forms. The most common form is a simple payment from the sender to the recipient that includes a change output which returns to the owner. Other forms include multiple inputs into a single output and a single input into multiple outputs.

4. CONCLUSIONS

4.1 Conclusions

In conclusion, we can state that both protocols use the unspent transaction output model, meaning that the transaction is a state transition that transfers value from previous outputs. Each transaction has an input and an output. In addition, in Grin they contain a transaction kernel.

The transaction process in Bitcoin is simpler since everything is done by the sender. Meanwhile, in Grin an interaction is necessary between the parties to conclude the process, meaning that they both need to sign the transaction (add their partial signatures). In a way this can be considered an advantage, since sending funds by mistake becomes impossible. Furthermore, it gives the chance to the recipient to refuse a transaction avoiding dust attacks where abusers send small amounts in an attempt to try and trace transactions. [5]

Grin transactions are considered private and indistinguishable from one another. This happens because the concept of addresses in Grin does not exist and every transaction is completely confidential, since the amounts and the owner are not shown. Furthermore, in case of 2 transactions where one spends the other all intermediary information is removed because they can be merged in a block.

Finally, while the output in Bitcoin gets locked with several stack operations, in Grin it is in the form of a Pedersen Commitments. To unlock it there is no need of the signature and the public key. It can be unlocked by the amount and the blinding factor which is known only by the recipient of the previous transaction.

References

1. *Davis, Joshua (10 October 2011). "The Crypto-Currency: Bitcoin and its mysterious inventor". The New Yorker. Archived from the original on 1 November 2014. Retrieved 31 October 2014.*
2. *Antonopoulos, A., 2018. Mastering bitcoin. Beijing: O'Reilly.*
3. *Scalingbitcoin.org. 2022. [online] Available at: <https://scalingbitcoin.org/papers/mimblewimble.txt> [Accessed 01 June 2022].*
4. *Developers, G., 2022. Mimblewimble - Grin Documentation. [online] Docs.grin.mw. Available at: <https://docs.grin.mw/wiki/introduction/mimblewimble/mimblewimble/> [Accessed 03 June 2022].*
5. *what-is-grin. 2022. Transaction building process. [online] Available at: <https://phyro.github.io/what-is-grin/interactive_txs.html> [Accessed 03 June 2022].*