Mimblewimble Semi-Interactive Payments

Jakob Abfalter

September 2019

1 Introduction

Mimblewimble is a protocol for a payment system in which participants record transactions in a public ledger, which is updated by a distributed consensus protocol similar to Bitcoin. The protocol was initially introduced in 2016 by an anonymous user who called himself Tom Elvis Jedusor (the french name of the character Voldemort in Harry Potter). Similar to the name of the anonymous person, the protocols name, Mimblewimble, originates from a spell, which characters cast in Harry Potter to tongue-tie their enemies [2].
Later the protocol was worked out in more detail by Cryptographer Andrew Poelstra [3] and has so far seen two instantiations in Cryptocurrencies Beam[1] and Grin[2].
The protocols focus is to hide both the transferred value in a transaction, as well as the transaction history. Hiding values is achieved by using homomorphic commitments in combination with range proofs instead of the cleartext values. Furthermore, transactions histories are hidden by pruning spent coins from the ledger. By doing this, we lose the ability to validate the history of the Blockchain, which is the currently most common way to validate the ledger in a Cryptocurrency. Instead, in Mimblewimble ledger validity is verified with so-called Kernel Excess values, which represent the difference between input and output coins in a transaction. The excess value is signed by sender and receiver in a interactive protocol and proofs that a transaction is balanced (does not generate coins) as well as coin ownership.

2 Problem description

The concept of addresses, which are used for example in Bitcoin or Ethereum currently does not exist in Mimblewimble based cryptocurrencies. Instead ownership of a coin is given by the knowledge of its blinding factor and value. Together they are the opening of the homomorphic commitments stored in the ledger. As a result transaction building requires both the sender and receivers active participation.

[1] https://beam.mw/
[2] https://grin-tech.org/

Figure 1 shows how transaction building originally was envisioned. First the sender would create his change coin, calculate his part of the key used later to sign the excess value. He then will transmit this information through a secure channel to the receiver, who adds his receiving coin, finalizes and publishes the transaction. This interaction however turned out to be vulnerable, as the receiver would be able to reverse the transaction after it was published and send back the value to the original coin.
The vulnerability was avoided in the implementations of the protocol by using a multiparty protocol to generate the final signature. The process is depicted in figure 2, which is the implementation used in the Grin Cryptocurrency.[3] Fuchstauer et al. showed in their paper [1] an alternative way to construct transactions. Their suggested flow is shown in figure 3 and is a salvaged version of the original protocol. The sender will build a special coin in an already completed transaction and sends information to the receiver to spend this coin. The receiver will create a second transaction spending this special coin. He then merges the two transactions, which can be done easily in Mimblewimble, and publishes the merged result to the ledger. However, the protocol described by Fuchsbauer et al. has some potential drawbacks. For the duration in which the receiver has not yet spent the special coin transferred to him, both the sender and receiver have the ability to spend it. Additionally, if the secure channel gets compromised, a third party listening might be able to retrieve the keys for the coin and spend it, thereby stealing the coins. In the proposed protocol the final published transaction furthermore contains one additional kernel excess and range proof, making the final transaction slightly bigger than in the currently employed method.

It is of high interest if we could do better. Specifically, we would like to construct a transaction construction protocol which only needs one message exchange instead of two, such as the one described by Fuchsbauer et al., but without the outlined flaws. Furthermore, a protocol initiated by the receiver rather then the sender would be out of interest. This would allow merchants to provide payment options without actively having to have an open listening connection at all times. Grinbox[4], a service built to allow users to have Grin addresses like in Bitcoin, where senders can send coins to asynchronously, shows the interest in this way of transacting for Grin.

References

[1] Georg Fuchsbauer, Michele Orrù, and Yannick Seurin. Aggregate cash systems: a cryptographic investigation of mimblewimble. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 657–689. Springer, 2019.

[3] https://medium.com/@brandonarvanaghi/grin-transactions-explained-step-by-step-fdceb905a853

[4] https://grinbox.io/

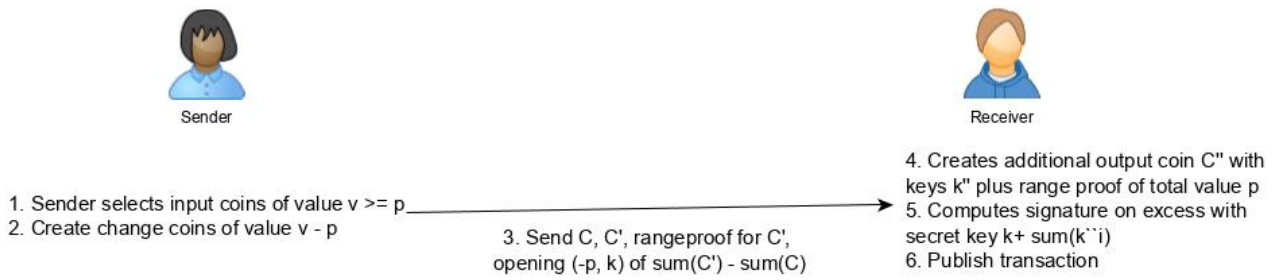

Figure 1: Original transaction flow

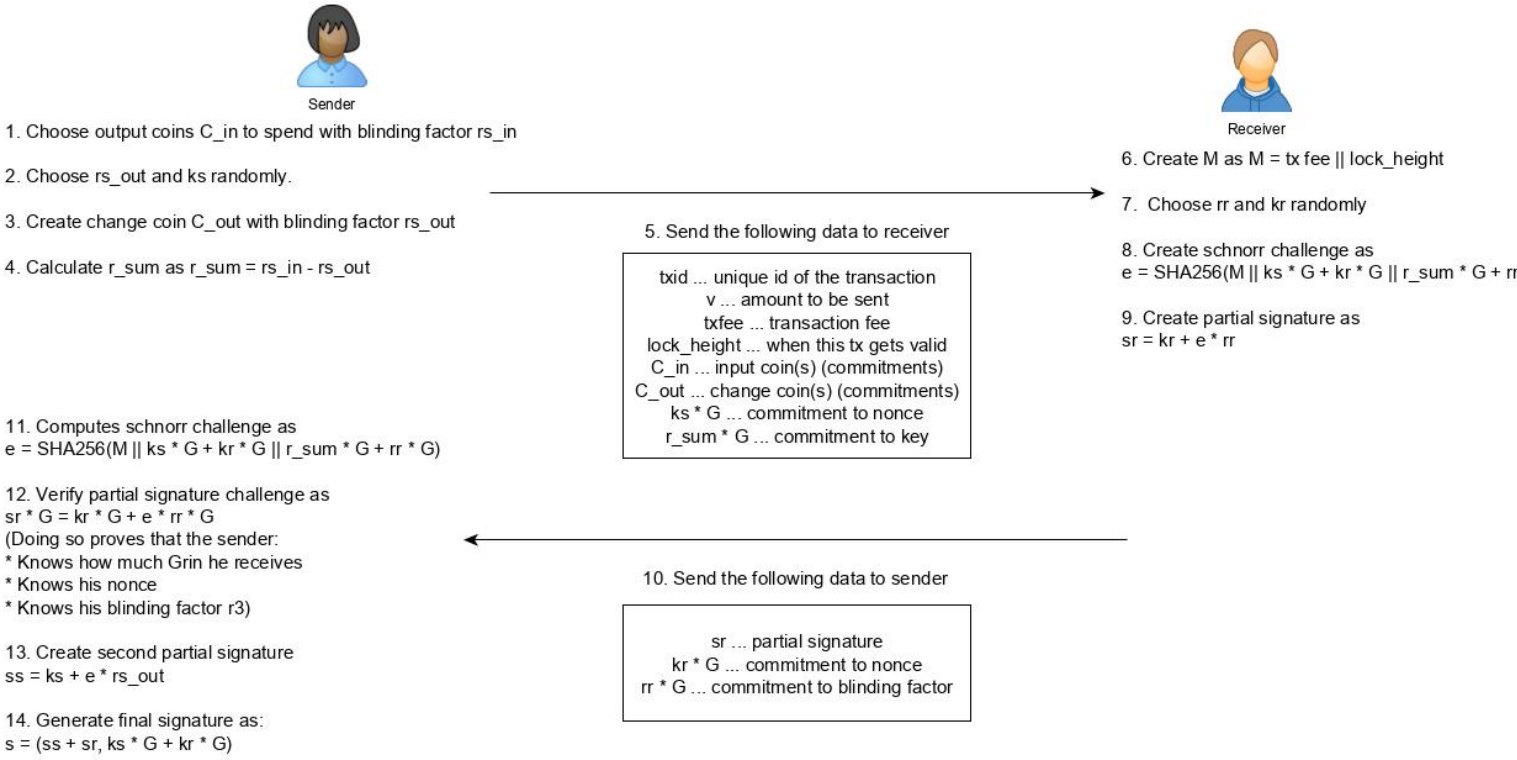

Figure 2: Grin transaction flow

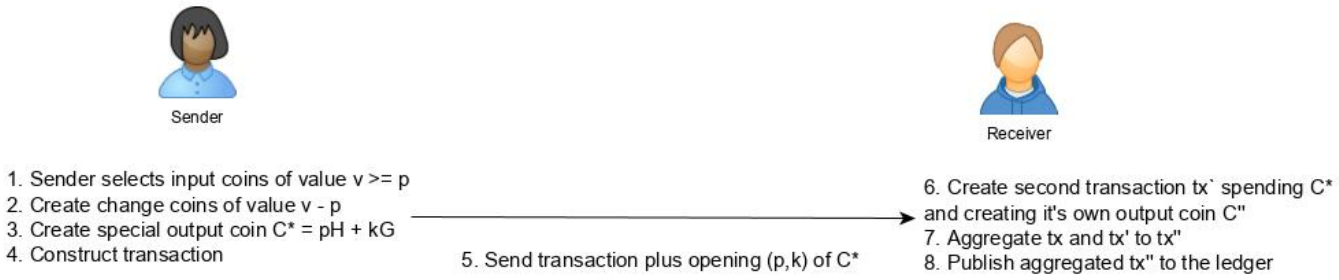

Figure 3: Salvaged transaction flow

[2] Tom Elvis Jedusor. Mimblewimble, 2016.

[3] Andrew Poelstra. Mimblewimble, 2016.