Mimblewimble: Private, Massively-Prunable Blockchains

At about 04:30 UTC on the morning of August 2nd, 2016, an anonymous person using the name Tom Elvis Jedusor signed onto a Bitcoin research IRC channel, dropped a document hosted on a Tor hidden service, then signed out. The document, titled Mimblewimble, described a blockchain with a radically different approach to transaction construction from Bitcoin, supporting noninteractive merging and cut-through of transactions, confidential transactions, and full verification of the current chainstate without requiring new users to verify the full history of any coins.

Mimblewimble's security model is based on the premise that nobody should be able to reverse a transaction without rewriting the block it appeared in, but otherwise no specific information about the transaction needs to be retained forever. In particular, if a transaction's outputs are spent in a later block, it should not be necessary for later validators to know about those outputs, only that they (whatever they were) were not stolen, i.e., they were not spent without knowledge of their associated key. Therefore "full verification" of a Mimblewimble chain allows one to prove that all coins followed a valid path from coinbase to unspent output, without knowing the path itself.

Mimblewimble accomplishes this, at the lowest level, by replacing the digital signatures that Bitcoin uses with "blinded key exchanges" in which holders of secret keys can transfer Bitcoins to the holders of other secret keys. Unlike Bitcoin, while the individual holders of keys can assure themselves that this exchange happens honestly, the network itself cannot validate this, so it is possible to use Mimblewimble in unsafe ways. This freedom of usage translates into the algebraic freedom to do massive blockchain pruning without compromising the ability to validate. Unfortunately, this pruning is incompatible with Bitcoin script, so Mimblewimble supports far less functionality than Bitcoin.

In particular, in Mimblewimble fully-validating new users must download the chain of blockheaders, all unspent outputs, Confidential Transaction rangeproofs for these outputs, and a collection of "excess" data totalling 96 bytes per historic transaction. A Mimblewimble chain with Bitcoin's transaction history would need about 120Gb for this data; contrast Bitcoin with Confidential Transactions which would need about 1.1Tb. Fully-validating users must keep on disk only the list of unspent outputs, without rangeproofs, which would total around 1.2Gb, versus 1.6Gb for Bitcoin. (Unfortunately *these* savings are entirely due to Mimblewimble not supporting Script.)

The space savings for new users comes from Mimblewimble's ability to delete old transactions, saving only their effect on the blockchain state and a 96-byte "excess". This means the bulk of the blockchain state scales with the number of unspent outputs rather than the size of all historic transactions. This gives a simple parameter to control blocksize scaling: the UTXO set could be limited to some constant multiple of the block height, and the fee market would react accordingly to shrink this size.

Extensions and future research include the following: multisignatures and locktimed transactions, which are sufficient to create payment channels on top of Mimblewimble; aggregation of these 96 byte "excess" values within blocks, so that their total data scales with the length of the blockchain rather than the total number of transactions (for a system with Bitcoin's usage, this alone would reduces the excess data from 14.4Gb to 128Mb); aggregation *across* blocks of these values, which when combined with Compact SPV as described in Appendix B of the sidechains

whitepaper, would further reduce this to log-cubed scaling in the blockchain length; and further compression of the rangeproofs required on all unspent outputs.

In this talk we discuss the Mimblewimble system as proposed by Tom Jedusor, described above. We describe the system, its drawbacks – primarily its lack of script support, then look at extensions and future research.