Mimblewimble: Private, Massively-Prunable Blockchains

Andrew Poelstra

grindelwald@wpsoftware.net

November 21, 2016

History

- 04:30 UTC, August 2nd, 2016: "Tom Elvis Jedusor" posts a .onion link to a text file on IRC, titled MIMBLEWIMBLE and dated July 19.

History

- 04:30 UTC, August 2nd, 2016: "Tom Elvis Jedusor" posts a .onion link to a text file on IRC, titled MIMBLEWIMBLE and dated July 19.
- Next morning: myself and Bryan Bishop verify it's actually just text and rehost it.

History

- 04:30 UTC, August 2nd, 2016: “Tom Elvis Jedusor” posts a .onion link to a text file on IRC, titled MIMBLEWIMBLE and dated July 19.
- Next morning: myself and Bryan Bishop verify it’s actually just text and rehost it.
- Following week: discussion on Reddit with Greg Sanders and others leads to understanding Mimblewimble’s trust model, and hints that the new crypto has merit.

History

- 04:30 UTC, August 2nd, 2016: "Tom Elvis Jedusor" posts a .onion link to a text file on IRC, titled MIMBLEWIMBLE and dated July 19.
- Next morning: myself and Bryan Bishop verify it's actually just text and rehost it.
- Following week: discussion on Reddit with Greg Sanders and others leads to understanding Mimblewimble's trust model, and hints that the new crypto has merit.
- September: myself and Avi Kulkarni develop an extension, "sinking signatures", to greatly improve its scaling properties.

History

- 04:30 UTC, August 2nd, 2016: "Tom Elvis Jedusor" posts a .onion link to a text file on IRC, titled MIMBLEWIMBLE and dated July 19.
- Next morning: myself and Bryan Bishop verify it's actually just text and rehost it.
- Following week: discussion on Reddit with Greg Sanders and others leads to understanding Mimblewimble's trust model, and hints that the new crypto has merit.
- September: myself and Avi Kulkarni develop an extension, "sinking signatures", to greatly improve its scaling properties.
- October 8th: released a paper showing Avi's and my work for Scaling Bitcoin Milan

History

- At 23:47 UTC, October 20, “Ignotus Peverell” appeared on IRC announcing a project to implement MimbleWimble.

History

- At 23:47 UTC, October 20, “Ignotus Peverell” appeared on IRC announcing a project to implement MimbleWimble.
- A few minutes later, Bryan Bishop called me to tell me to join the conversation. I pointed out that aggregate signatures give space savings on top of the Voldemort scheme, even without new crypto.

History

- At 23:47 UTC, October 20, “Ignotus Peverell” appeared on IRC announcing a project to implement MimbleWimble.
- A few minutes later, Bryan Bishop called me to tell me to join the conversation. I pointed out that aggregate signatures give space savings on top of the Voldemort scheme, even without new crypto.
- Other Harry Potter characters arrived over the next few weeks; the project continues to move forward. Though I’ve been involved with the project, I have not contributed any code.

History

- At 23:47 UTC, October 20, “Ignotus Peverell” appeared on IRC announcing a project to implement MimbleWimble.
- A few minutes later, Bryan Bishop called me to tell me to join the conversation. I pointed out that aggregate signatures give space savings on top of the Voldemort scheme, even without new crypto.
- Other Harry Potter characters arrived over the next few weeks; the project continues to move forward. Though I’ve been involved with the project, I have not contributed any code.
- I am not Ignotus Peverell.

What is Mimblewimble?

- Mimblewimble is a design for a blockchain-based ledger that is very different from Bitcoin.

What is Mimblewimble?

- Mimblewimble is a design for a blockchain-based ledger that is very different from Bitcoin.
- It can be implemented as a sidechain, or softforked into Bitcoin (as an extension block).

What is Mimblewimble?

- Mimblewimble is a design for a blockchain-based ledger that is very different from Bitcoin.
- It can be implemented as a sidechain, or softforked into Bitcoin (as an extension block).
- In Bitcoin transactions, old outputs sign new outputs; outputs have “script pubkeys” that are independent of each other. In Mimblewimble transactions, outputs have only EC pubkeys, and the difference between new outputs’ keys and old ones’ is multisigned by all transacting parties.

What is Mimblewimble?

- Mimblewimble is a design for a blockchain-based ledger that is very different from Bitcoin.
- It can be implemented as a sidechain, or softforked into Bitcoin (as an extension block).
- In Bitcoin transactions, old outputs sign new outputs; outputs have "script pubkeys" that are independent of each other. In Mimblewimble transactions, outputs have only EC pubkeys, and the difference between new outputs' keys and old ones' is multisigned by all transacting parties.
- Mimblewimble transactions are inherently scriptless.

Mimblewimble Transactions

A Mimblewimble transaction is the following data:

- Inputs (references to old outputs).

Mimblewimble Transactions

A Mimblewimble transaction is the following data:

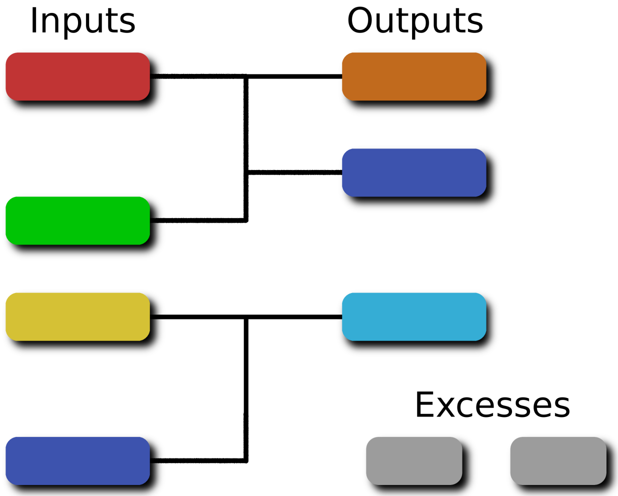
- Inputs (references to old outputs).
- Outputs: confidential transaction outputs (group elements, which blind and commit to amounts), plus rangeproofs.

Mimblewimble Transactions

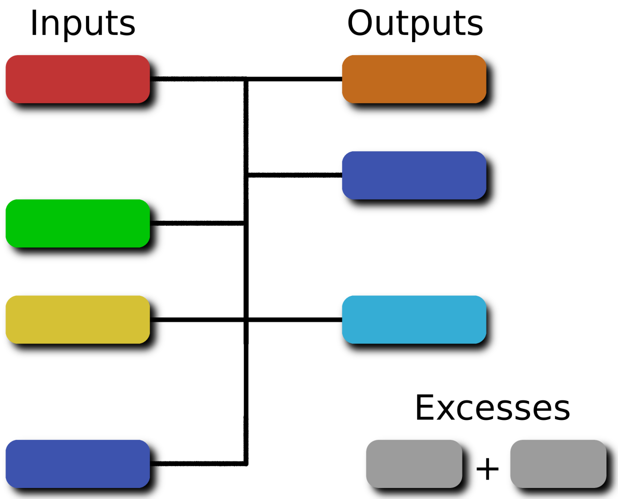
A Mimblewimble transaction is the following data:

- Inputs (references to old outputs).
- Outputs: confidential transaction outputs (group elements, which blind and commit to amounts), plus rangeproofs.
- Excess: difference between outputs and inputs (group element), plus signature (for authentication and to prove non-inflation)

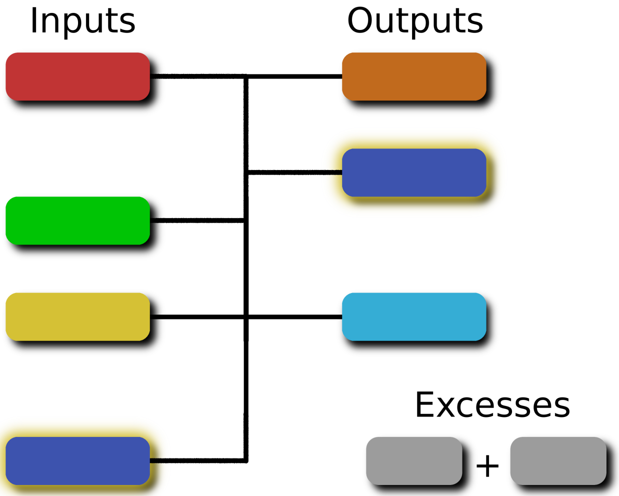
Mimblewimble Transactions



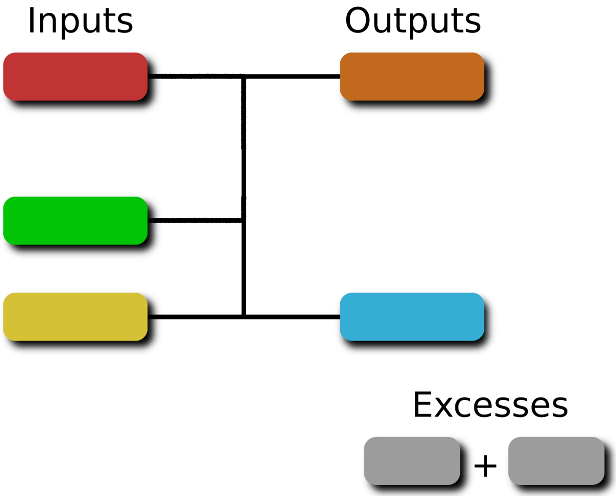
Mimblewimble Transactions



Mimblewimble Transactions



Mimblewimble Transactions



Mimblewimble Blocks

Blocks consist of:

- A merkle tree of transaction inputs.

Mimblewimble Blocks

Blocks consist of:

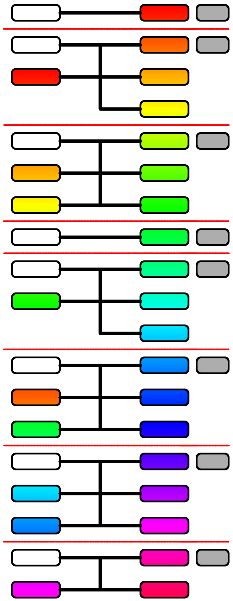
- A merkle tree of transaction inputs.
- A merkle tree of transaction outputs and rangeproofs.

Mimblewimble Blocks

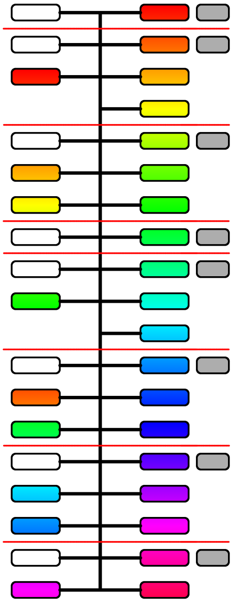
Blocks consist of:

- A merkle tree of transaction inputs.
- A merkle tree of transaction outputs and rangeproofs.
- A list of excess value(s) and signature(s)

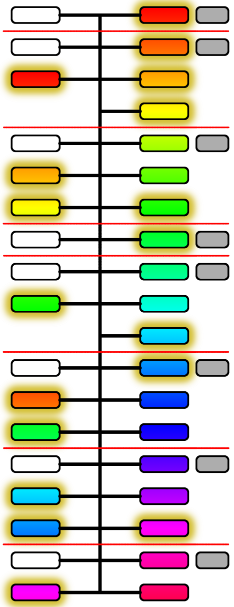
Mimblewimble Transactions



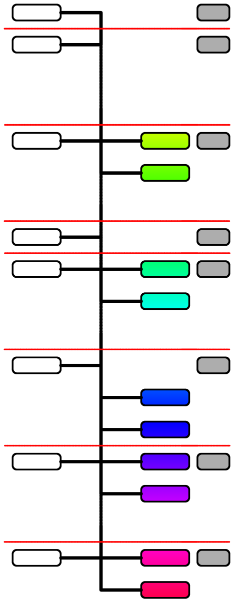
Mimblewimble Transactions



Mimblewimble Transactions



Mimblewimble Transactions



Trust Model: Transactions

A transaction is valid if:

- It is non-inflationary (total input amount equals total output amount)

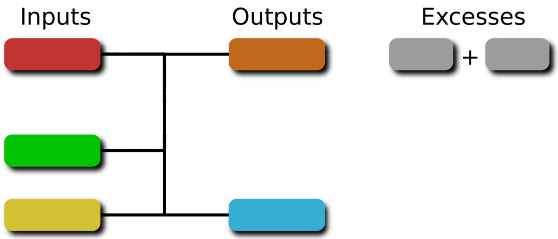
Trust Model: Transactions

A transaction is valid if:

- It is non-inflationary (total input amount equals total output amount)
- The owner of the input(s) has signed off on it.

Trust Model: Transactions

A transaction is valid if:

- It is non-inflationary (total input amount equals total output amount)
- The owner of the input(s) has signed off on it.
- 

Trust Model: Blockchain

It should be verifiable that

- A transaction, once committed to a block, cannot be reversed without doing enough work to rewrite the block (and all its descendants).

Trust Model: Blockchain

It should be verifiable that

- A transaction, once committed to a block, cannot be reversed without doing enough work to rewrite the block (and all its descendants).
- The current state of all coins reflects zero net theft and inflation.

Trust Model: Blockchain

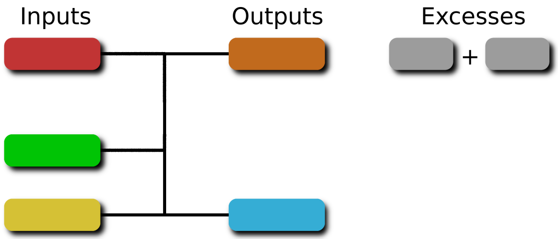
It should be verifiable that

- A transaction, once committed to a block, cannot be reversed without doing enough work to rewrite the block (and all its descendants).
- The current state of all coins reflects zero net theft and inflation.
- *The exact historical sequence of transactions does not need to be publicly verifable.*

Trust Model: Blockchain

It should be verifiable that

- A transaction, once committed to a block, cannot be reversed without doing enough work to rewrite the block (and all its descendants).
- The current state of all coins reflects zero net theft and inflation.
- *The exact historical sequence of transactions does not need to be publicly verifable.*
-



Trust Model: Block Verification

It is possible to verify the blockchain with only the following data:

- Block headers

Trust Model: Block Verification

It is possible to verify the blockchain with only the following data:

- Block headers
- Unspent outputs from each block

Trust Model: Block Verification

It is possible to verify the blockchain with only the following data:

- Block headers
- Unspent outputs from each block
- Excess values and signatures.

Trust Model: Block Verification

It is possible to verify the blockchain with only the following data:

- Block headers
- Unspent outputs from each block
- Excess values and signatures.
- Rangeproofs for the above (witness data)

Trust Model: Block Verification

It is possible to verify the blockchain with only the following data:

- Block headers
- Unspent outputs from each block
- Excess values and signatures.
- Rangeproofs for the above (witness data)
- Full blocks near the tip should be kept to handle reorgs

Trust Model: Block Verification

It is possible to verify the blockchain with only the following data:

- Block headers
- Unspent outputs from each block
- Excess values and signatures.
- Rangeproofs for the above (witness data)
- Full blocks near the tip should be kept to handle reorgs
- In Bitcoin there are 150 million transactions and 40 million unsigned transaction outputs: 21.6Gb of historic data, 2Gb of UTXOs and 100Gb of UTXO rangeproofs.

Next Steps

- Development, development, development!

Next Steps

- Development, development, development!
- Nail down chain parameters

Next Steps

- Development, development, development!
- Nail down chain parameters
- Sidechain / asset support

Next Steps

- Development, development, development!
- Nail down chain parameters
- Sidechain / asset support
- More crypto ;)

Open Problems

- Unconditionally sound commitments and rangeproofs

Open Problems

- Unconditionally sound commitments and rangeproofs
- Smaller rangeproofs? Aggregation of rangeproofs?

Open Problems

- Unconditionally sound commitments and rangeproofs
- Smaller rangeproofs? Aggregation of rangeproofs?
- Peer-to-peer protocol that can handle transaction merging

Open Problems

- Unconditionally sound commitments and rangeproofs
- Smaller rangeproofs? Aggregation of rangeproofs?
- Peer-to-peer protocol that can handle transaction merging
- Quantum resistance

Thank You

Andrew Poelstra `<grindelwald@wpsoftware.net>`