MimbleWimble

Few ideas within crypto have garnered as much attention as the MimbleWimble proposal. MimbleWimble is a novel protocol that works to improve privacy and scalability for its users. MimbleWimble takes the main architecture of Bitcoin, removes script and adds Confidential Transactions. MimbleWimble was created as an idea for improving scalability and privacy within Bitcoin. Due to the level of modification and tradeoffs involved, it's not currently politically or technologically feasible to include on Bitcoin proper. However, it may be possible to implement as a sidechain in the future. A sidechain is a separate blockchain that utilizes its own consensus, but not its own token. Instead it uses the bitcoin via a two-way peg. Cryptography, while generally trusted, is usually only empirically correct. Most cryptography is not **proven** to be correct. Confidence is gained by lasting a long time without being broken by security researchers. Cryptography relies on assumptions that certain computations are difficult enough that they are practically impossible. Bitcoin depends on the discrete log problem and cryptographic hashing. The beauty of Bitcoin is that it involves only relatively simple cryptographic assumptions that have decades of research. Those primitives being the discrete log problem and cryptographic hashing. If the discrete log problem and hashing assumptions hold, then Bitcoin and MimbleWimble are secure. MimbleWimble combines these relatively simple cryptographic primitives in very sophisticated ways and the result is a system that is quite complex.

Discrete Log Problem

Cryptography is built on the idea that certain operations are easy to compute in one direction, and near impossible to compute in the other direction. The discrete log problem is an important example of this and is one of the most fundamental assumptions. It allows innovations like public key cryptography and signatures to work and be highly efficient. The discrete log problem comes from discrete mathematics, a branch of math that deals with a limited set of values. Like Bitcoin, MimbleWimble relies on elliptic curve cryptography (ECC). In ECC, math operations are defined over a range that is the set of points that satisfy a specifically chosen elliptic curve. Points can be added, subtracted and multiplied easily with standard computers. However, division is very difficult, and brute force is the only currently known way. The fact that multiplication is easy but division is difficult is the property we need to build powerful cryptography. In Bitcoin and MimbleWimble, public keys are derived from private keys. The protocol chooses a point on the elliptic curve and typically labels it H or G, called a generator point. The private key is actually just a whole number chosen at random from a very large set (on the order of 2^{128} or greater). The generator point is multiplied by the chosen scalar to give the public key. The fact that this multiplication is considered extremely difficult to reverse is what allows these systems to operate. Reversing multiplication is also known as taking a logarithm, hence the name discrete log problem.

Cryptographic Hashing

A cryptographic hash function takes an arbitrarily large amount of data called the input and "digests" it into a fixed length string of data called a hash. For a hash function to be a cryptographic hash function, the output must be completely unpredictable and unrelated to the input. A small deviation in the input should result in a completely different hash. This unpredictability is what allows these functions to be secure. Most importantly, it should not be possible for an attacker to find an input that corresponds to a hash from another input. Bitcoin uses the RIPEMD-160 function to generate an address from a public key and SHA-256 for its proof-of-work function. In MimbleWimble, hashing is used to create outputs which in fact are cryptographic commitments. These commitments do not reveal a destination address on the chain, but are only spendable when a user has possession of a private key. Both implementations of the MimbleWimble protocol use hashing as the proof-of-work consensus mechanism.

Homomorphic Encryption

Homomorphic encryption is a very cool technique that allows mathematic operations on encrypted numbers. It turns out multiplication is rather hard to get right in a homomorphic encryption scheme. Additively homomorphic meaning that addition and subtraction are preserved over encrypted values. The ability to check whether two separate summations result in equal values turns out to be tremendously powerful. As we'll see, MimbleWimble leans heavily on this additively homomorphic property to continuously verify that the sum of the inputs equals the sum of the outputs without needing to know the values themselves.

Confidential Transactions

Confidential Transactions (CT) uses a Pedersen Commitment scheme which replaces plaintext unspent transaction outputs (UTXOs) values with cryptographic commitments. UTXOs represent individual piles of unspent money on the Bitcoin blockchain, an alternative approach to account balances. A cryptographic commitment binds the user to a chosen value without revealing what that value is. This means if and when the time comes for the user to reveal the chosen value, they cannot change their mind about the value as only the original value will satisfy the mathematics involved. The really cool part is that only recipients of a CT need to learn what the value actually is. Pedersen commitments follow the additive homomorphic property and therefore allow us to check that the sum of the inputs equals the sum of the outputs within a transaction. Transactions can be validated without knowing the amount transacted — a big win for privacy. MimbleWimble uses the CTs scheme for bookkeeping on its blockchain. There are no observable values on MimbleWimble, only cryptographic commitments and range proofs. The homomorphic additive property ensures that the total money supply in the system can be continuously checked without having amounts be visible.

Cut Through

As just mentioned, MimbleWimble collapses all transactions within a block into a single block-wide transaction. The structure and transaction boundaries are removed. If a transaction is spending a very fresh (unconfirmed) input, then it is possible to completely remove the intermediary outputs without affecting the validity of the chain.

To conclude

The MimbleWimble protocol combines the above into a specification for a blockchain suitable for simple payments. It uses a modified version of CTs so that the balances are stored as cryptographic commitments rather than publicly visible amounts. Transaction structure is removed within each block, and the blocks are validated as a whole. Interestingly, the system does away with addresses, and instead outputs are actually commitments that can only be spent by people with

knowledge of a particular parameter used to create the commitment. This parameter is known as a blinding factor and was originally included in CTs purely for privacy. In a clever modification, MimbleWimble uses the blinding factor as the private key that authorizes the spending of an output. These blinding factors are now fundamental to authentication, and must not be shared.

MimbleWimble is a stripped down blockchain protocol suitable for simple payments. Since it removes addresses, senders and receivers must cooperate via a secure and private medium to create transactions before broadcasting a transaction to the network. This is significantly different from address based systems where it is easy to receive money while offline and without a private communication channel.

Privacy

Amounts are obscured and it is difficult for third parties to decipher what is happening without extensive outside knowledge. The consolidation of transactions within a block helps privacy as well. However if a spy node receives transactions individually, they can begin to compile a forensics database that associates inputs with outputs, possibly linking them to IP addresses. This information could later be used to possibly deanonymize parts of the chain later with learned information. Both implementations of MimbleWimble utilize a proposal called Dandelion a network routing proposal originally created for Bitcoin that creates plausible deniability. It passes transactions around via several hops and, in MimbleWimble, later aggregates them randomly before they are sent to miners for inclusion into a block. This will make it much harder for spy nodes to learn about what's happening. Transactions reliably obscures the amounts, but the transaction graph is only hidden as well as the user can find simultaneous transactions to combine theirs with.

Scalability

MimbleWimble does not feature significant improvements in tx/s over existing cryptocurrencies. CTs offers privacy benefits, but requires significant resources. Combining individual transactions at the block level removes a small amount of bandwidth overhead. The biggest benefit is for new nodes joining the network. Recall that the chain is validated by continuously checking whether the total input and output sides of the equation balance. Because of this, it is possible to prune matching inputs and outputs and still check that the chain validates. This means that when new nodes want to join the network, they may be able to download just the relevant subset of historical inputs and outputs. Existing nodes are also able to reclaim a bit of disk space.

Grin vs Beam

Grin and Beam are two separate implementations of the MimbleWimble protocol. Beam is structured as a product of a company, while Grin is an open source community effort. Both projects have chosen a similar block time. The other main difference between the projects lies in their monetary policy. Some Grin supporters believe that its true purpose is to be a Bitcoin testnet. Beam has a hardcoded supply cap and a team incentivized with Beam tokens, so they have made choices more appropriate for an increasing token value.