Zcash was the first widespread application of zk-SNARKs, a novel form of zero-knowledge cryptography. The strong privacy guarantee of Zcash is derived from the fact that shielded transactions in Zcash can be fully encrypted on the blockchain, yet still be verified as valid under the network's consensus rules by using zk-SNARK proofs.

The acronym zk-SNARK stands for "Zero-Knowledge Succinct Non-Interactive Argument of Knowledge," and refers to a proof construction where one can prove possession of certain information, e.g. a secret key, without revealing that information, and without any interaction between the prover and verifier.

Zero-knowledge" proofs allow one party (the prover) to prove to another (the verifier) that a statement is true, without revealing any information beyond the validity of the statement itself. For example, given the hash of a random number, the prover could convince the verifier that there indeed exists a number with this hash value, without revealing what it is.

In a zero-knowledge “Proof of Knowledge” the prover can convince the verifier not only that the number exists, but that they in fact know such a number – again, without revealing any information about the number. The difference between “Proof” and “Argument” is quite technical and we don’t get into it here.

How zk-SNARKs are applied to create a shielded transaction

In Bitcoin, transactions are validated by linking the sender address, receiver address, and input and output values on the public blockchain. Zcash uses zk-SNARKs to prove that the conditions for a valid transaction have been satisfied without revealing any crucial information about the addresses or values involved. The sender of a shielded transaction constructs a proof to show that, with high probability:

- the input values sum to the output values for each shielded transfer.

- the sender proves that they have the private spending keys of the input notes, giving them the authority to spend.

- The private spending keys of the input notes are cryptographically linked to a signature over the whole transaction, in such a way that the transaction cannot be modified by a party who did not know these private keys.

In addition, shielded transactions must satisfy some other conditions that are described below.

Bitcoin tracks unspent transaction outputs (UTXOs) to determine what transactions are spendable. In Zcash, the shielded equivalent of a UTXO is called a “commitment”, and spending a commitment involves revealing a “nullifier”. Zcash nodes keep lists of all the commitments that have been created, and all the nullifiers that have been revealed. Commitments and nullifiers are stored as hashes, to avoid disclosing any information about the commitments, or which nullifiers relate to which commitments.

For each new note created by a shielded payment, a commitment is published which consists of a hash of: the address to which the note was sent, the amount being sent, a number “rho” which is unique to this note (later used to derive the nullifier), and a random nonce.

Commitment = HASH(recipient address, amount, rho, r)

When a shielded transaction is spent, the sender uses their spending key to publish a nullifier which is the hash of the secret unique number (“rho”) from an existing commitment that has not been spent, and provides a zero-knowledge proof demonstrating that they are authorized to spend it. This hash must not already be in the set of nullifiers tracking spent transactions kept by every node in the blockchain.

Nullifier = HASH(spending key, rho)

The zero-knowledge proof for a shielded

transaction verifies that, in addition to the conditions listed above, the following assertions are also true:

- For each input note, a revealed commitment exists.

- The nullifiers and note commitments are computed correctly.

- It is infeasible for the nullifier of an output note to collide with the nullifier of any other note.

In addition to the spending keys used to control addresses, Zcash uses a set of proving and verifying keys to create and check proofs. These keys are generated in the public parameter ceremony discussed above, and shared among all participants in the Zcash network. For each shielded transaction, the sender uses their proving key to generate a proof that their inputs are valid. Miners check that the shielded transaction follows consensus rules by checking the prover’s computation with the verifying key. The way that Zcash’s proof generation is designed requires the prover to do more work up-front, but it simplifies verifying, so that the major computational work is offloaded to the creator of the transaction (this is why creating a shielded Zcash transaction can take several seconds, while verifying that a transaction is valid only takes milliseconds).

The privacy of Zcash’s shielded transactions relies upon standard, tried-and-tested cryptography (hash functions and stream ciphers), but it’s the addition of zk-SNARKs, applied with the system of commitments and nullifiers, that allows senders and receivers of shielded transactions to prove that encrypted transactions are valid.