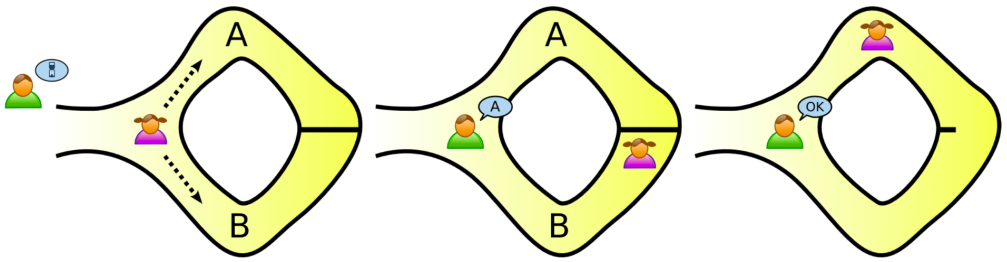
Zerocash Summary

Tim Sigl

zk-SNARKs are used in zero-knowledge cryptography. When it comes to blockchain zk-SNARKs can be used to store transactions encrypted on the ledger while still being able to verify the transactions. zero-knowledge Succinct Non-interactive ARguments of Knowledge (zk-SNARKs) allow to prove the possession of certain information without revealing the information itself. Succinct means that the proof can be verified with milliseconds. Non-interactive means that there is only one message from the prover to the verifier needed. A random public reference string is needed for the algorithm to work. The generation of this reference is done by a trusted party. It is important that the random parameters used in the generation are destroyed and never published. Otherwise malicious users will be able to create counterfeit coins. A transaction has to be converted to an “arithmetic circuit” which is a directed acyclic graph of basic mathematical operations like multiplication and addition. After the arithmetic circuit is created a Rank 1 Constraint System is built to check if values travel correctly through the circuit. To compress the arithmetic circuit all of the constraints can be converted to a single representation called Quadratic Arithmetic Program. The result is a polynomial which has to be checked at random points to verify it. To make the polynomial “zero-knowledge” it can just be randomly shifted. A commitment in Zcash is the equivalent to a UTXO in Bitcoin. The generation of the proof is more computationally expensive than the verification which can be done in milliseconds.
A proof is complete if an honest verifier will always be convinced of a true statement by an honest prover. A proof is sound if a cheating prover can convince an honest verifier that some false statement is true with only a very small probability. A very simple example for a zero knowledge proof is the following. A cave has a secret door deep in the cave that just opens after saying a secret word. Peggy wants to sell the secret to Victor but Victor wants to be certain that Peggy knows the word before paying her. The game works as follows, Peggy chooses one of the two paths into the cave either A or B at random. Then Victor tells a path he wants Peggy to return on. Without knowing the secret Peggy would only be able to take the path back she went into the cave. Otherwise if she knows the secret key she can return on the other path back because she can open the secret door with her key. So if Victor asks Peggy enough times to return on different paths he can be certain that Peggy knows the key without him knowing the key.



A magical cave [7]

Bitcoin blockchains can easily be de-anonymized with different analyzing tools. Transaction histories in Bitcoin can be obfuscated via mixes. Mixes are risky for users because mix

providers can steal their funds. Zerocoin was a predecessor of Zerocash providing a decentralized mix. Zerocash improves on reduction of transaction to under 1 kB, reduction of verification time to under 6 ms, allowing for transaction with variable amounts, hiding transaction amounts and values held by users. zk-SNARKs employ three different base algorithms. KeyGen generates a proving key (pk) and a verification key (vk) which both are public and used for proving and verification. Prove(pk, x, a) → pi, with a := input, pk := proving key, x := statement and pi := proof, the prover proves outputs a in a non-interactive proof pi. Verify(vk, x, pi) → b, with vk := verification key, x := statement, := proof, b := boolean indicating the validity. The Zerocash protocol can be applied on top of a ledger-based base currency such as Bitcoin. Zerocash adds two new transactions. Mint transaction record that a coin with a coin commitment and value has been minted. Pour transactions record the usage of two inputs coin_1 and coin_2 identified by their serial numbers and creates two new output coins with respective coin commitments. Zerocash employs 6 new algorithms. 1. System setup to generate public parameter. 2. Creating payment addresses for each user. 3. Mint coins to create new coins. 4. Pour coins to transfer value from input coins to new coins. 5. Verify transaction to check the validity of transaction (both mint and pour) and last 6. receive coins scans the ledger and retrieves unspent coins. The pour algorithm has 4.1 mio gates in the arithmetic circuit. Zerocash can either be added as a replacement to the base unit of a currency or as a hybrid currency where both exist and can be converted. On a single threaded machine the KeyGen takes close to 8 mins creating a proving key of close to 900 MiBs and a verification key of only 750 bytes. The setup and Pour algorithms are the only expensive algorithms.