

0.1 Plasma, Optimistic Rollup and zkRollup

0.2 Data Availability

0.3 layer-2

0.4 Merkle Trees

0.5 mimc

0.6 Replay Attacks

A common security consideration to make in cryptographic protocols are replay attacks. A replay attack occurs when a valid cryptographic proof is maliciously resubmitted, and no security checks are in place to invalidate the resubmission. When making a transaction in a blockchain network, the transaction is signed with the user's private key. Miners that receive transactions check the signature's validity, which authorizes the transaction as the user has access to the address's private key. However, a malicious actor could store these signed transactions and resubmit them to the network. Since the signature is still valid, a mechanism is needed to invalidate transaction signatures, once they have been included in a block and thereby executed. In most blockchain systems, like Bitcoin or Ethereum, this is solved by tracking a nonce for each address. Miners extract an address's current nonce by indexing the entire blockchain, simply counting the number of confirmed transactions. Among other things, the nonce is part of the signed transaction. When a miner checks the signature, it is also checked if the nonce set in the signed transaction has been incremented by one, when comparing with the nonce that was extracted. This invalidates a resubmission of the signed transaction and prevents replay attacks.

Transaction Replay Attacks in zkSwap The way transactions are executed in this system is quite different, compared to the blockchain systems described above. As we will explore, most verifications and computations happen off-chain and are then verified on-chain with a single proof, that

When making a transaction, which can be a trade, a deposit or a withdrawal, a user signs the transaction details and sends them to our off-chain entity, the aggregator. The signature is then checked in a zkSNARK circuit, along with the nonce. The resulting proof can then be used to verify the correct execution of the zkSNARK circuit, as an effect verifying the correct. While the signature check happens off-chain, the mechanism of preventing replay attacks is comparable to most blockchain systems. By checking if the nonce signed in the transaction details has been incremented, we ensure the signed transaction is invalidated when resubmitted.

Proof Verification Replay Attacks Another consideration to make, is the submission and verification of the zkSNARK proof. It also could be

resubmitted, thereby breaking the protocol if no measures are in place to prevent this. When verifying a zkSNARK proof, we're essentially proving the execution of the circuit was done correctly, and that the resulting outputs were computed by running the circuit. In this system, the zkSNARK circuit is used to ensure correct state transitions tbc

0.7 zk-SNARKs