TX MIXERS

By Simon CHEREL & Enguerrand DECLERCQ

Contents

Topics to tackle

1. The blockchain
2. Limitations to transparency
3. The rise of privacy coins and mixers
4. Zero-knowledge proof
5. Tornado.cash ZKP implementation
6. Sanctions
7. Limiting risks

The blockchain

A trustless public ledger

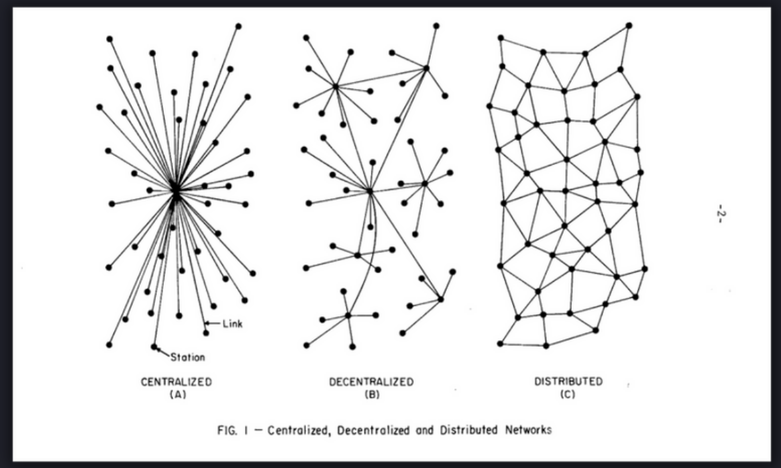


FIG. 1 — Centralized, Decentralized and Distributed Networks

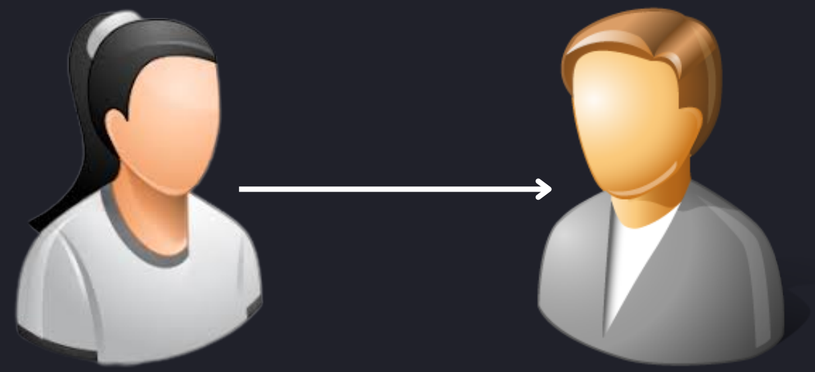
Networks possible configurations

- Decentralized network of validators (= servers)
- Agreement on the same state of the machine
- Content shared among nodes
- Tamperproof & fault resistant
- Turing-complete smart contracts
- No deletion or update rights
- Permissionless creation and read rights

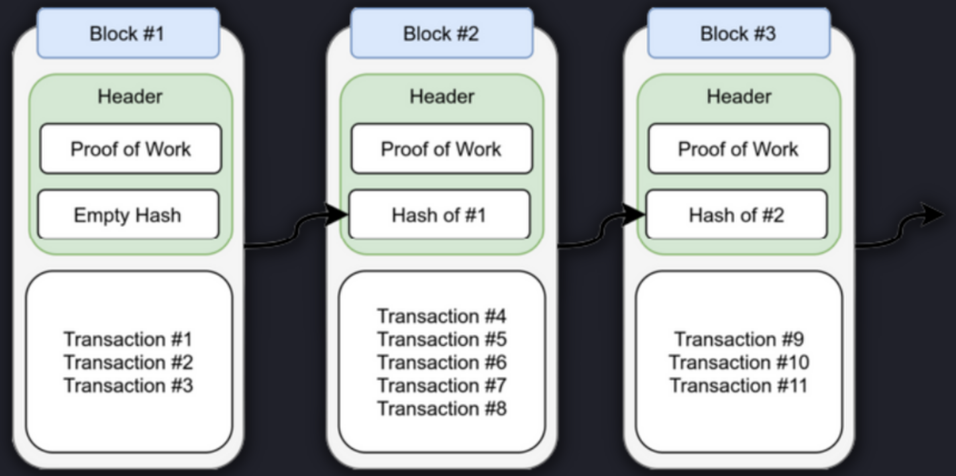
Transparency

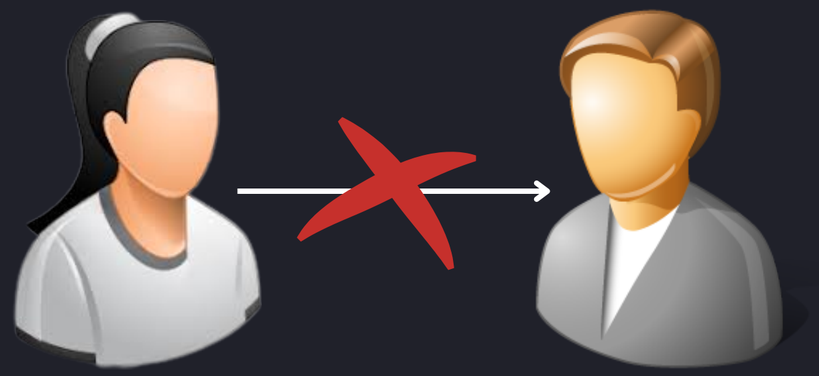
Limitations of a public ledger

- **Pseudonymous:** wallet owners can be doxxed;
- **No privacy:** each transaction traces back to a sender and a receiver;
- **No secrets:** everyone can read a smart contract's memory.



Alice Bob





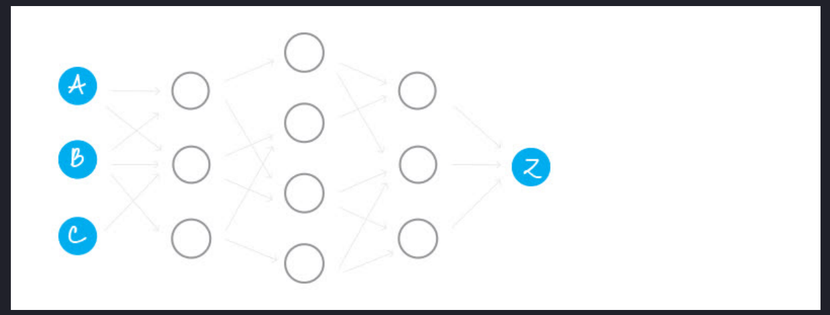
Alice

Bob

Privacy reborn

The rise of privacy coins & mixers

- **Privacy:** the bond between a sender and a receiver is at least obfuscated, ideally broken;
- **No secrets:** everyone can still read a smart contract's memory;
- Several privacy-enhancing technologies exist.



Privacy coins

The original mixers

- Obfuscation through transaction mixing;
- High throughput;
- Users need to buy the token (i.e. pass a KYC) to use the service

Monero

A privacy coin example

- The amount sent through each transaction can be hidden, thanks to their zero-knowledge proof algorithm;
- Senders' IP addresses are obscured to prevent external intel.

tornado CASH

Otter cash

Smart contracts

The so-called CoinJoins

- Obfuscation by gathering coins together;
- Slow & low throughput on purpose;
- Relies on ZKP;
- Strict memory & computation limitations.

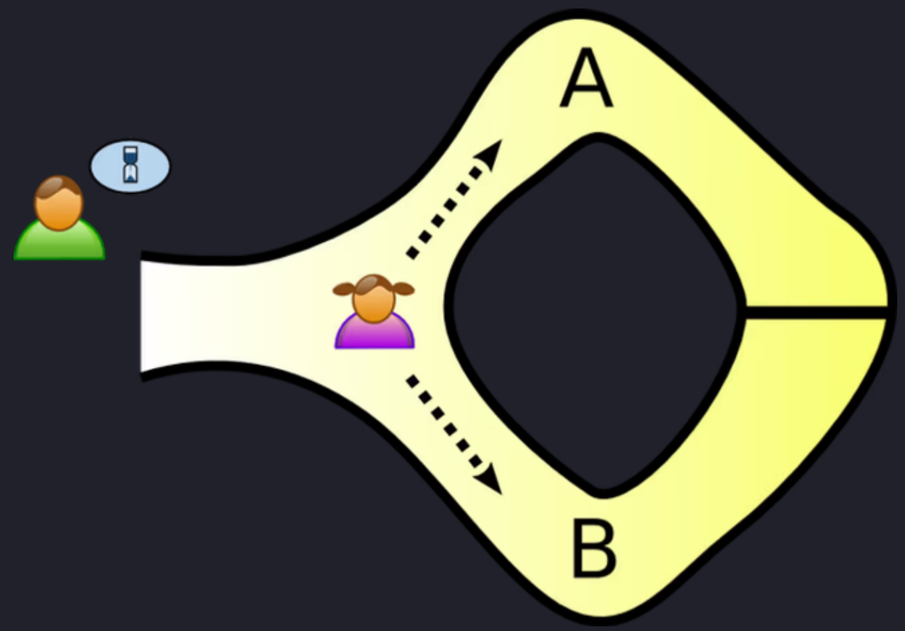
A cryptographic system should be designed to be secure, even if all its details, except for the key, are publicly known.

Auguste Kerckhoffs

ZKP

A cryptosystem

- Prover and Verifier want to prove $\exists w \mid Q(x,w)=0$
- Anyone has access to the context x
- Prover has a secret w
- Verifier wants to verify Q(x,w)=0
- Prover and verifier cannot share w
- Prover sends proof π which does not reveal w but verifies Q(x,w)=0



An exemple of zkp

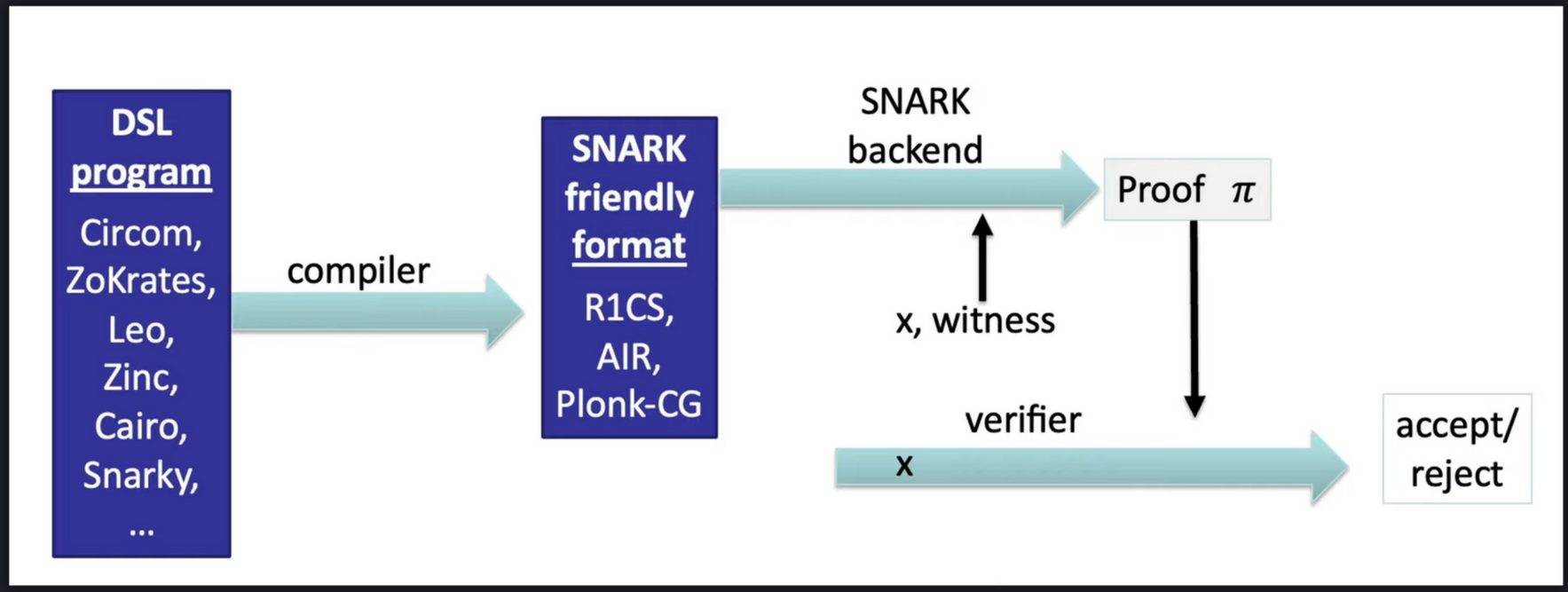
zk-SNARK

Succinct non-iterative argument of knowlege

- **Argument of knowledge:** V accepts π => P knows w | Q(x,w)=0;
- **Succint:** π is easy to compute and easy to verify (vital to minimize gas fees);
- **Non-iterative:** a single exchange between P and V is sufficient to transmit π.

zk-SNARK

An actual infrastructure



A crowd

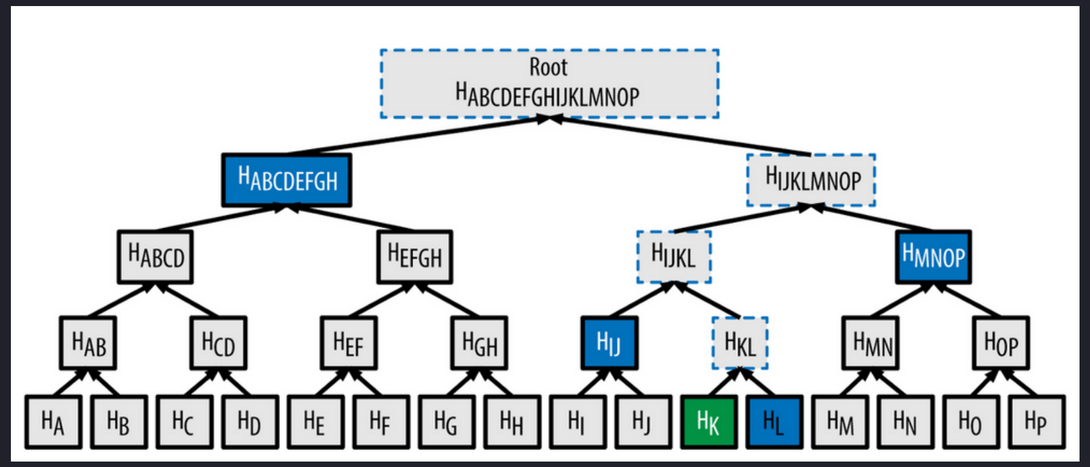
Tornado Cash

A zk-SNARK application

- Pool which aggregates the token;
- A pool for each amount (0.1 ETH, 1 ETH, 10 ETH)
- zk-SARK protocol to withdraw founds without revealing identity

Setup

A zk-SNARK application



Merkel tree

- Two hash functions H1 and H2;
- 20-level Merkel Tree T, all leaves are initialized with 0, each node is calculated with the H2 of each previous node;
- The leaves list is public;
- The root R of T is also public;
- Merkel Proof for the next leaf entrance (contains the hash of each node needed to compute the root from the next entrance);
- A list of the nullifier associated with previous withdrawal.

Depositing

A wants to deposit N Eth

- A generate k and r, two random number of 248bits
- A compute C=H2(k || r)
- A send to the smart contract, C and N Eth

- The contract received this transaction then :
- place C into the next leaf
- update the merkel tree and the merkel proof

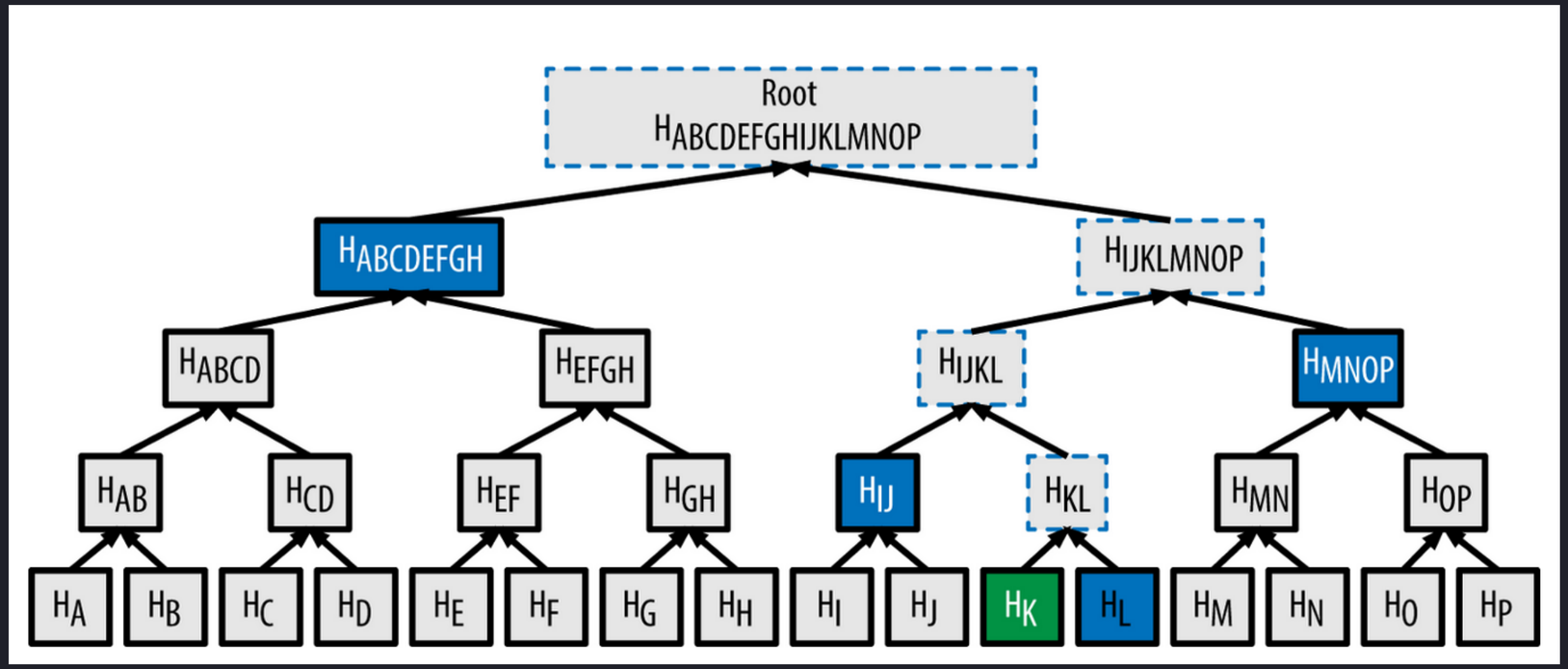
Withdrawing

A knows k and r, A wants to withdraw

- A doesn't want to reveal to which leaf she is linked, this is where zk-Snark is used
- A can prove that she has the (k,r) for a leaf without revealing which one and (k,r)
- w=(k,r,C,MerkelProof(C))
- x=(R,nf,WithdrawalAdress) where nf=H1(k)
- Build π
- Send (nf,π,WithdrawalAddress) to the contract
- Contract verify π, and verify nf isn't into the list of nullifier
- Contract send N Eth to WithdrawalAddress and add nf to the nullifier list

Merkle trees

A lightweight storage solution



Departure fees

An open problem

Tornado cash logo

- To interact with the smart contract, you need to pay gas fees, no matter how small the computation;
- Filling the payer's wallet with tokens may expose the payer's identity or link with the sender;
- Off chain solution, relayer network.