Privacy on the Blockchain

The need for privacy

- Supply chain privacy:
    - A manufacturer does not want to reveal how much it pays its supplier for parts.
- Payment privacy:
    - A company that pays its employees in crypto wants to keep list of employees and salaries private;
    - End users need privacy for rent, donations, purchases;
- Business logic privacy:
    - Smart Contracts code

Blockchains cannot reach their full potential without some form of private transactions

Types of privacy

- Pseudonymity: (weak privacy)
- Full anonymity: User’s transactions are unlinkable

Types of privacy

No privacy:
Everyone can see all transactions

Privacy from the public:
Only a trusted operator can see transactions

VISA MasterCard

Semi-full privacy:
only “local” law enforcement can see transactions

full privacy:
no one can see transactions

tornado

Privacy in Ethereum / MultiversX

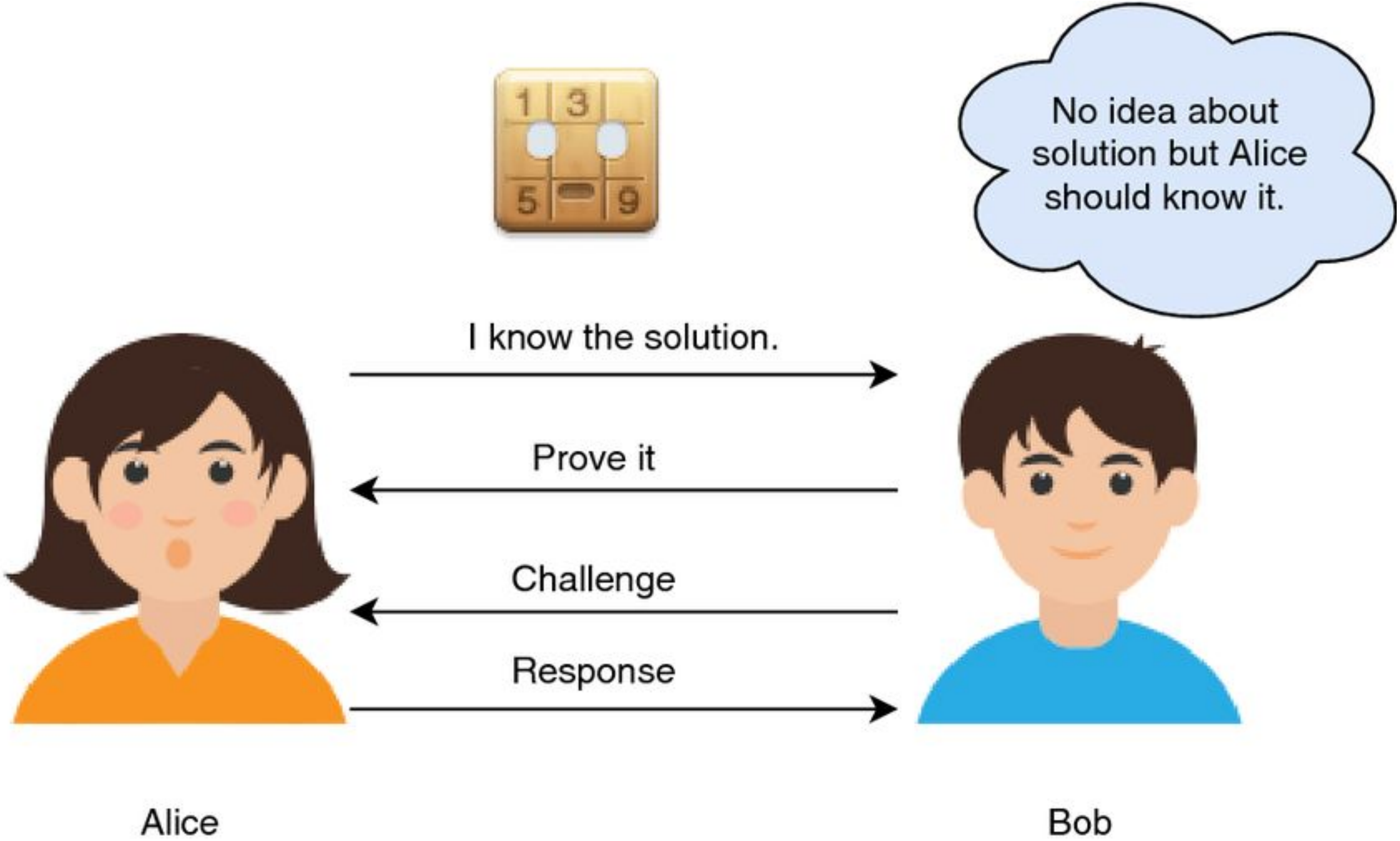
- Every account balance is public
- For Dapps: code and internal state are public
- All account transactions are linked to account
- In time: Linking an addresses to an identity

Negative aspects of complete privacy

- Criminal activity
- Challenge:
  - How to support positive applications of private payments, but prevent the negative ones?
  - Can we ensure legal compliance while preserving privacy?
  - **Zero knowledge proofs**
  - **Fully Homomorphic Encryption**

Zero Knowledge

- **Prover** can prove to **Verifier** that they possess certain information without revealing the information
- **Completeness**: If the statement is true, an honest prover can convince an honest verifier
- **Soundness**: If the statement is false, no dishonest prover can convince the verifier of its truth, except with negligible probability
- **Zero-Knowledge**: The verifier learns nothing about the information other than the fact that the statement is true



No idea about solution but Alice should know it.

I know the solution.

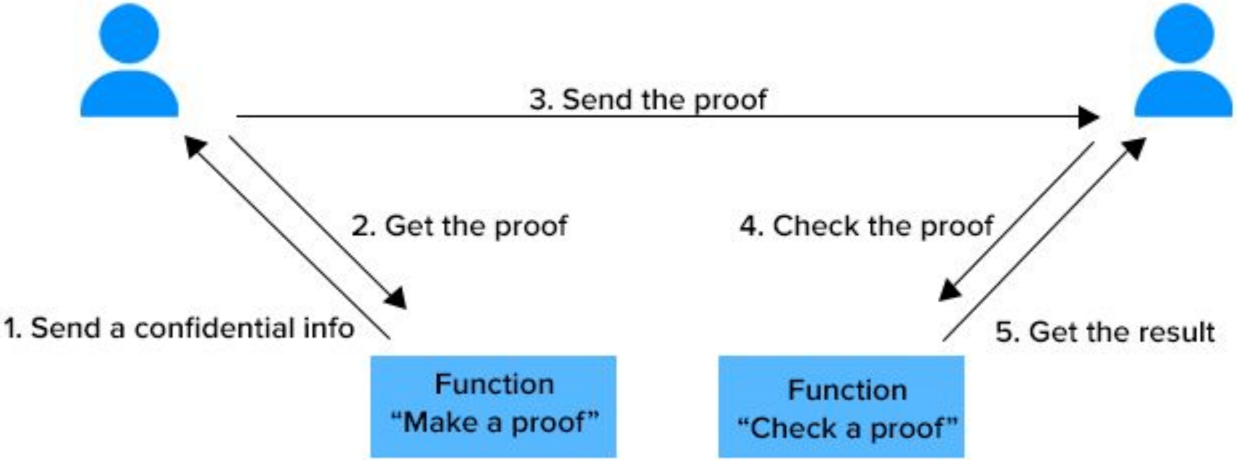
Prove it

Challenge

Response

Alice

Bob



3. Send the proof

2. Get the proof

4. Check the proof

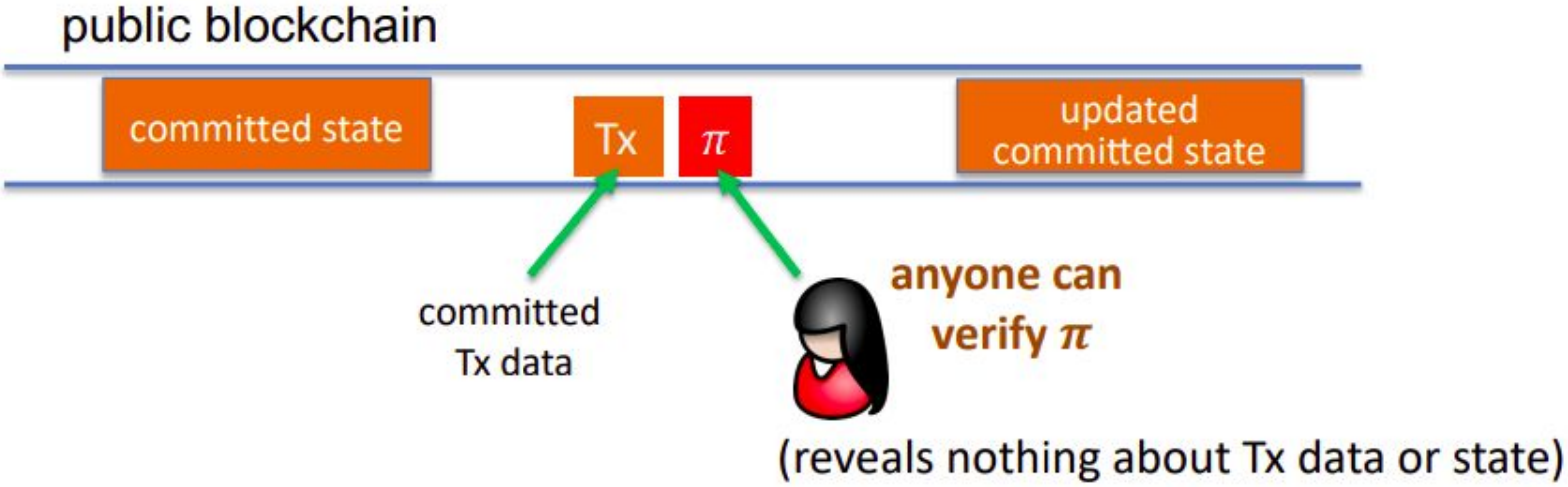
1. Send a confidential info

5. Get the result

Function
“Make a proof”

Function
“Check a proof”

Paradigm for Private Transaction



ZK Proofs - Applications

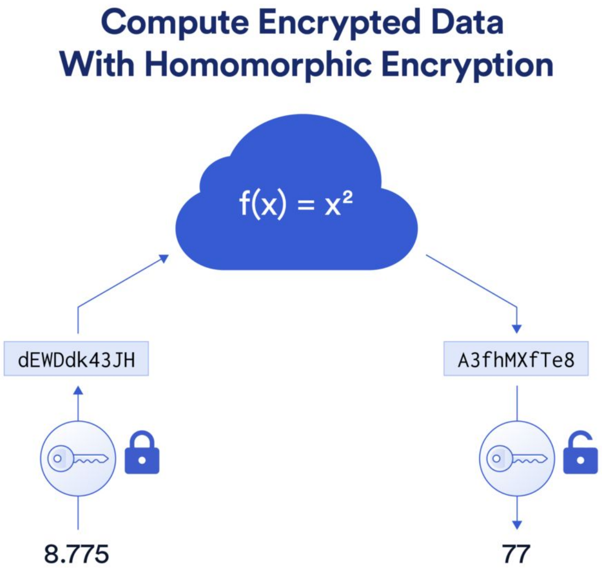
- Blockchain privacy
  - Zcash
  - Allows users to prove the validity of a transaction (e.g., ownership, amount) without revealing details
- Authentication
  - Passwordless authentication where users prove they know a password without revealing it
- Voting Systems
  - Ensures that votes are valid and counted without revealing who cast each vote
- Data Sharing
  - Prove compliance (e.g., age, income level) without disclosing sensitive details

ZK Proofs - Challenges

- High computational cost
  - Prover: 96 vCPUs and 384 GB RAM may be required for large-scale proofs.
  - Solver: Consumer-grade devices (laptops, smartphones) can easily verify proofs in milliseconds
- Implementation complexity
  - Developing and verifying ZKP systems can be intricate.
- Trust assumptions
  - Some systems (like ZK-SNARKs) require a "trusted setup" phase.

Fully Homomorphic Encryption (FHE)

- Allows computations to be performed on encrypted data without the need for prior decryption



Compute Encrypted Data
With Homomorphic Encryption

Resources

- Rekt
  - https://rekt.news/leaderboard/
- Bug Bounties
  - https://immunefi.com/bug-bounty/
- Healthcare + Blockchain + FHE
  - https://www.ledgerinsights.com/nebula-genomics-blockchain-data-privacy/