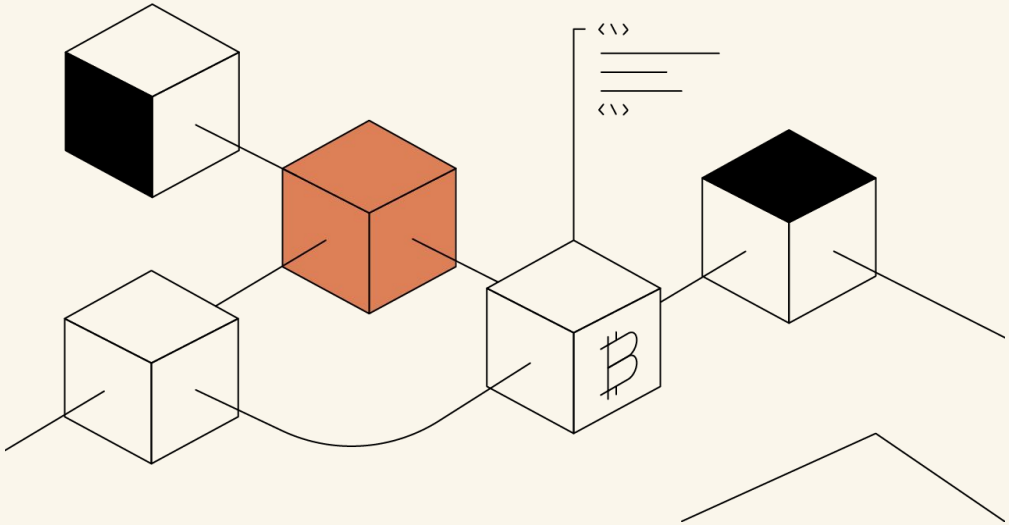
LEHIGH UNIVERSITY

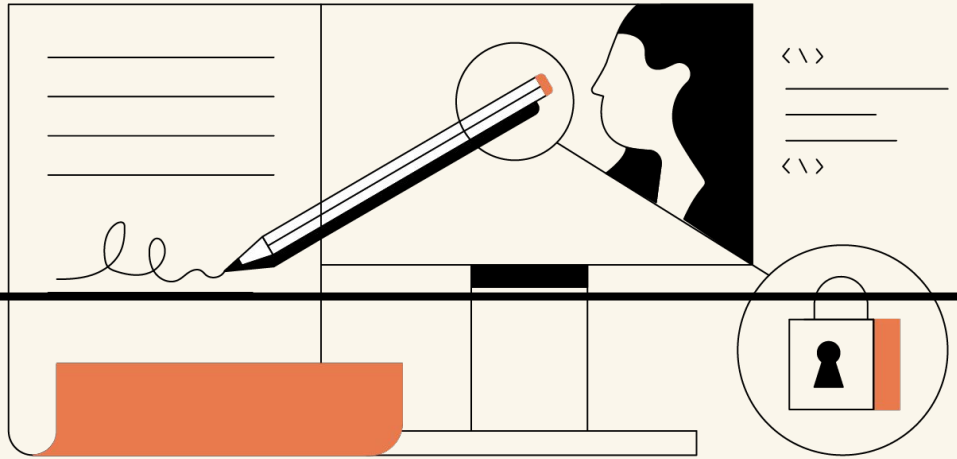
Zcash: Enabling Fully Confidential Transactions with Zero Knowledge

Tal Derei

Agenda



1. **Why Bitcoin isn't Private**
2. **Privacy on a Public Blockchain?**
3. **Zcash Protocol**



Why Bitcoin isn't Private

Bitcoin History

Bitcoin was introduced by **Satoshi Nakamoto** (we don't know who he/she/they are) in the famous 2009 whitepaper.

First time a digital asset (BTC) can be exchanged on a decentralized, peer-to-peer network without an intermediary like banks, governments, or other central authority.

Bitcoin History

But wait….

Isn’t the public image of cryptocurrency like Bitcoin an international criminal’s currency of choice – an **anonymous**, **untraceable** means of **laundering** money.

…but the opposite is true

Bitcoin History

Bitcoin is the most **transparent** payment method ever developed and has the potential to become a powerful tool in the fight against financial crime.

How?

Bitcoin blockchain itself is **public** (and so are all transactions)

Bitcoin is **pseudonymous** rather than **anonymous**

- Your pseudonym (bitcoin address) is recorded on-chain
- Your identity is not
- Identities may be linked to bitcoin addresses

Bitcoin History

If Amanda (Bitcoin address) transacts with Olivia (large centralized exchange or online retailer), Amanda is leaking her customer identity information.

.
.
.

Bitcoin therefore provides the **ultimate** paper trail for law enforcement agencies, tax authorities and compliance professionals.

- This traceability also makes bitcoin theft a far less attractive endeavour, as the funds are “tainted”.

Examples

[1] Canada's Trucker Protests – “Freedom Convoy”

- Justin trudeau invokes the **Emergencies Act** – “banks can immediately freeze or suspend bank accounts tied to the truckers without a court order and without fear of civil liability.”

- Ordered all regulated financial firms to cease facilitating any transactions from 34 crypto wallets tied to funding trucker-led protests in the country.

Examples

[2] Silk Road (illegal online marketplace)

Over $1 billion in transactions took place on Silk Road between February 2011 and July 2013, and almost $80m of commissions were earned by the operators – all in bitcoins

…

FBI Agent Yum searched the bitcoin blockchain for transactions involving the bitcoin addresses found in the Silk Road wallet and those on **Ross Ulbricht's (creator of Silk Road)** laptop – and he found transactions (700k bitcoins) going directly between them.

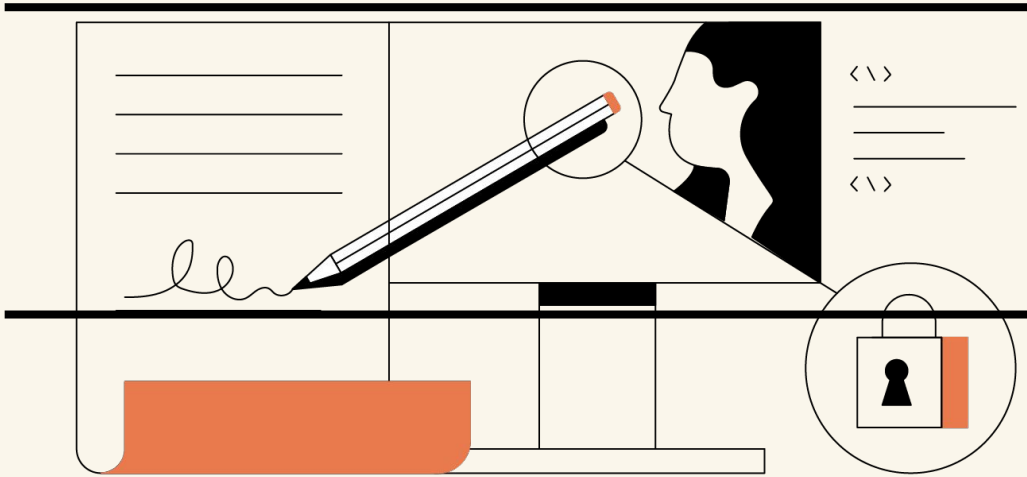
They seized the bitcoin and auctioned it on the open market.

Bitcoin Flawed

So what's the conclusion?

Bitcoin is architecturally incapable of satisfying it's original goal

... with serious implications for user privacy



Privacy On a Public Blockchain?

Public Transaction Model

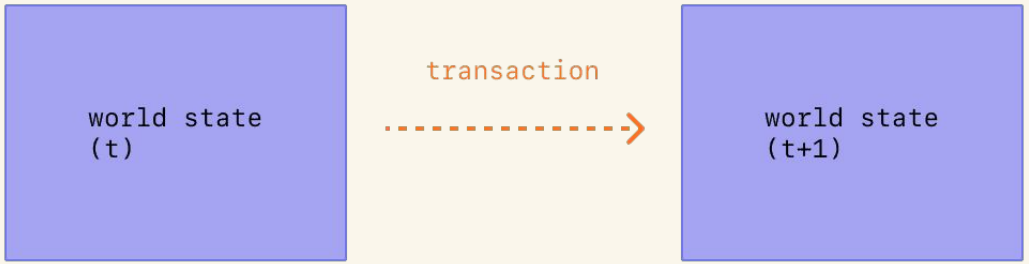
Q. Is Ethereum's public transaction model any different in terms of privacy?

Public Transaction Model

→ **1.** An account will initiate a transaction to update the state of the Ethereum network

→ **2.** Transactions are cryptographically signed instructions from **accounts** (i.e. accounts initiate transactions)

→ **3.** Miner executes the transaction, initiating a **state change** of the EVM that's broadcasted to the entire network (e.g. *Gossip Protocol + Consensus Mechanism)*



Executing Transactions

Transaction's Object Payload:

```
{
    from: "0xEA674fdDe714fd979de3EdF0F56AA9716B898ec8",
    to: "0xac03bb73b6a9e108530aff4df5077c2b3d481e5a",
    gasLimit: "21000",
    maxFeePerGas: "300",
    maxPriorityFeePerGas: "10",
    nonce: "0",
    value: "10000000000"
}
```

Signing Transactions

→ Transaction object needs to be signed using the sender's private key

JSON-RPC Call:

```
{
    "id": 2,
    "jsonrpc": "2.0",
    "method": "account_signTransaction",
    "params": [
        {
            "from": "0x1923f626bb8dc025849e00f99c25fe2b2f7fb0db",
            "gas": "0x55555",
            "maxFeePerGas": "0x1234",
            "maxPriorityFeePerGas": "0x1234",
            "input": "0xabcd",
            "nonce": "0x0",
            "to": "0x07a565b7ed7d7a678680a4c162885bedbb695fe0",
            "value": "0x1234"
        }
    ]
}
```

Transaction Response

JSON-RPC Response:


```
{
        "jsonrpc": "2.0",
        "id": 2,
        "result":
                {
                "raw":"0xf883800182033394 07a565b7ed7d7a678680a4c162885bedbb695fe080a44401a6e400000000",
                "tx": {
                        "nonce": "0x0",
                        "maxFeePerGas": "0x1234",
                        "maxPriorityFeePerGas": "0x1234",
                        "gas": "0x55555",
                        "to": "0x07a565b7ed7d7a678680a4c162885bedbb695fe0",
                        "value": "0x1234",
                        "input": "0xabcd",
                        "v": "0x26",
                        "r": "0x223a7c9bcf5531c99be5ea7082183816eb20cfe0bbc322e97cc5c7f71ab8b20e",
                        "s": "0x2aadee6b34b45bb15bc42d9c09de4a6754e7000908da72d48cc7704971491663",
                        "hash": "0xeba2df809e7a612a0a0d444ccfa5c839624bdc00dd29e3340d46df3870f8a30e"
                        }
                }
}
```

Potential Problem?

Sender and Recipient addresses are <u>PUBLIC</u>!

Pseudonymous, but still…

<u>Unanswered Questions:</u>

Q. Want to receive annual salary in cryptocurrency, but don't want to reveal how much you make?

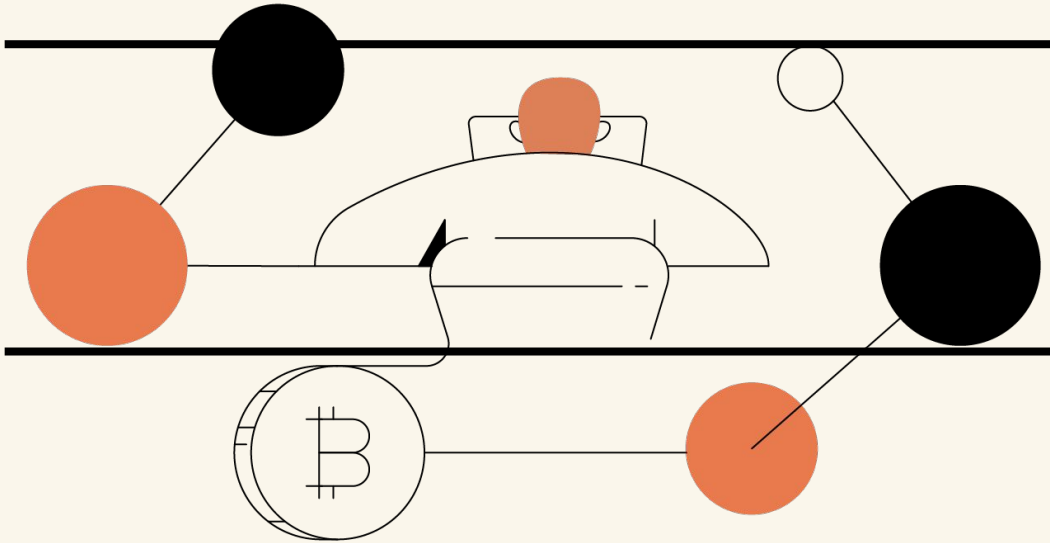
Q. Want to receive interest/dividend payments (i.e. more complex payment types)?

Q. Want to Interact with DeFi protocols privately (i.e. to take out a loan)?

New Solution

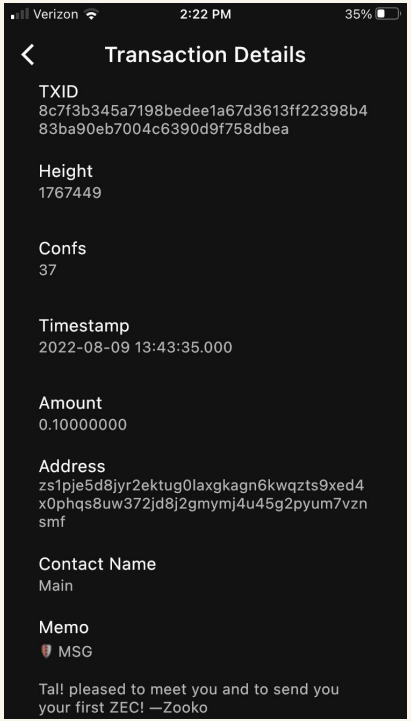
Enable users to confidentially transact on a blockchain protocol w/ "shielded transactions"

Zcash!

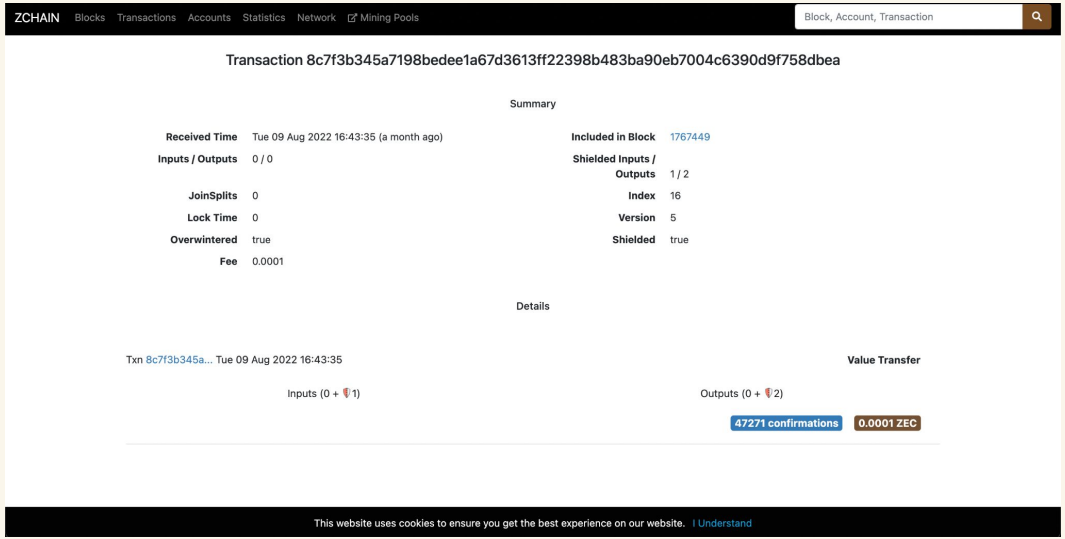


Zcash Protocol

Zcash Transaction



Transaction from **Zooko**, Zcash founder and CEO



Zcash Protocol

Zcash is a **privacy-preserving blockchain** (and **cryptocurrency**) for confidential transactions using…

.
.
.

Zero-Knowledge Proofs

So we can say more specifically, it's an open source **zero-knowledge protocol** for building privacy on blockchains

Zcash Protocol

Components:

→ **1.** UTXO Model

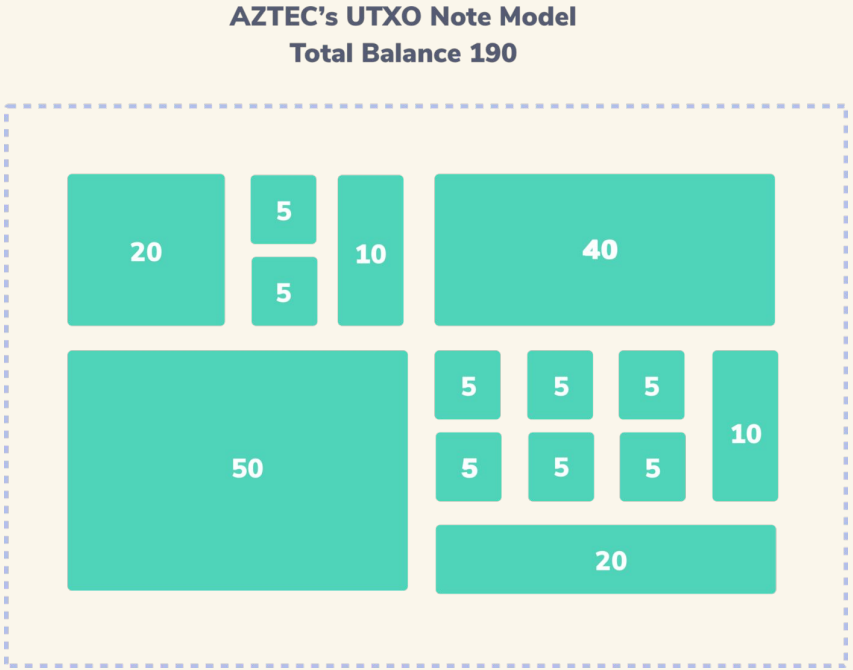
→ **2.** Transactions

→ **3.** Example

1. Unspent Transaction Output (UTXO) Model

Zcash Note

Zcash follows a **UTXO** model (different than Ethereum's account-based model) similar to that of **Bitcoin**. The core of any Zcash transaction is a **Note**



Zcash Note

In the same way an **account has a balance**

… a note has an owner

Zcash Notes

Notes (encrypted representation of value)

Makeup of a note:

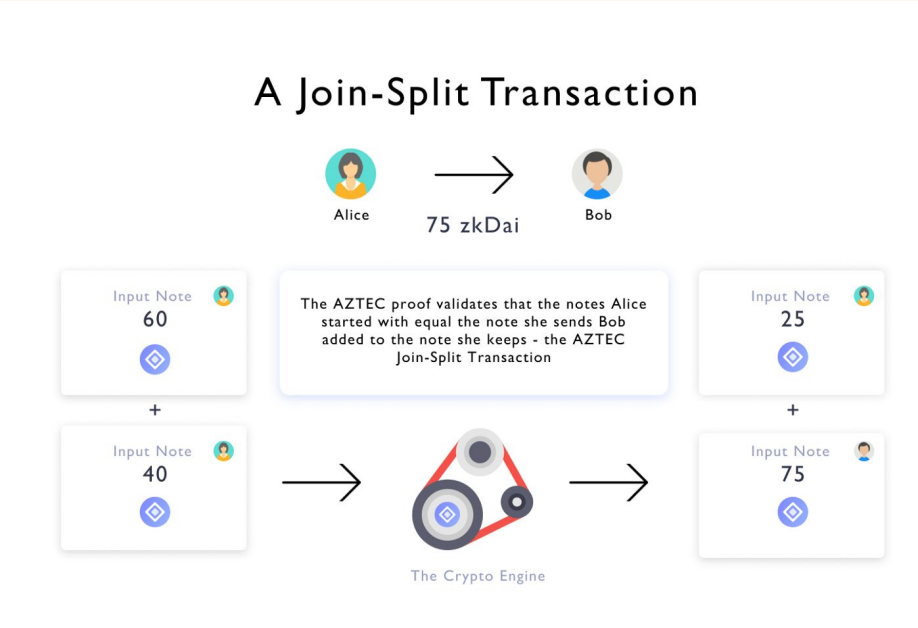
On-Chain Data

→ 1. Zcash address of **owner**
→ 2. Zcash note **public key**

Private Data

→ 1. Note **value**
→ 2. Note **viewing key**
→ 3. Note **spending key**

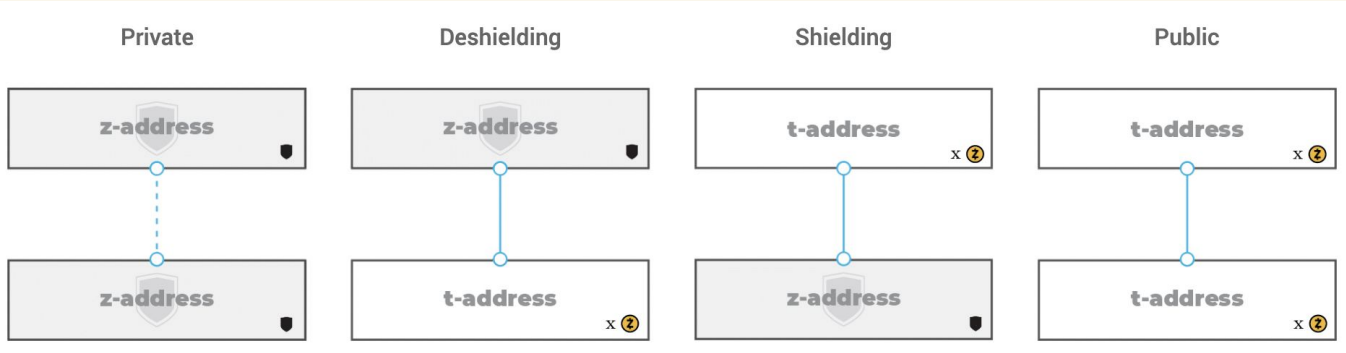
Confidential Transfers are *Balancing Relationships…*



"**Join Split** proof allows a set of input notes to be joined or split into a set of output notes."

2. Transactions

Transactions Types



- Zcash addresses are either private **(z-addresses)** or transparent **(t-addresses)**
  - **Z-to-Z:** transaction amount and the memo field are all encrypted and not publicly visible
  - **T-to-T:** sender, receiver and transaction value are publicly visible (like Bitcoin)
  - **T-to-Z or Z-to-T:** are possible, but privacy implications need to be considered

Keys

Spending Key = spend something

Viewing Key = view something

Proving Key = construct zk proof about something

Verification Key = verify zk proof about something

Transactions

Q. How are zk-SNARKs (i.e. zero knowledge proofs) applied to create a shielded transaction?

Transactions

Recall: **Bitcoin** transactions are validated by linking the sender address, receiver address, and input and output values on the public blockchain.

Bitcoin tracks unspent transaction outputs (**UTXOs**) to determine what transactions are spendable

.

.

.

In **Zcash,** we want to satisfy these conditions without leaking crucial information about the sender/receiver and amounts!

Transactions

How?

The sender of shielded transaction constructs a proof to show, with high probability:

- (1) Σ inputs values = Σ output values (joint-split proof)
- (2) Owns private spending key associated with input note
- (3) Private spending key of input notes cryptographically linked to a signature over whole transaction

Any other conditions shielded transactions need to satisfy?

Transactions

Zcash shielded equivalent of 'Bitcoin UTXO' is a **commitment**, and spending a commitment involves revealing a **nullifier**.

→ commitments + nullifiers are stored as *hashes* in a database

→ Zcash nodes keep lists of all the commitments that have been created, and all the nullifiers that have been revealed

Transactions

1. For each new **note** created by a shielded payment, a commitment is published:
   commitment = hash(recipient address, amount, rho, random nonce)

2. When shielded transaction is spent, sender uses their private sender key to:
   → publish a **nullifier** = hash (spending key, rho) to show commitment has not been spent already

   → provides a **zero-knowledge proof** that
   - for each input note commitment exists and
   - nullifiers/commitments are computed correctly and
   - infeasible for the nullifier of an output note to collide with the nullifier of any other note

Transactions

In addition to the **spending/viewing 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 (2016).

Transactions

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 privacy of Zcash's shielded transactions relies:

[1] cryptography (hash functions and stream ciphers)

[2] zk-SNARKs + system of commitments and nullifiers

...which ultimately allows senders and receivers of shielded transactions to prove that encrypted transactions are valid.

3. Example

Example

Each unspent transaction output (UTXO) describes an unspent note:

- **Address/public key of owner**
- **Balance**

Note 1 = (PK1), Note 2 = (PK2), etc…

Example

Now add a random ‘serial number’ which acts as a unique identifier:

Note 1 = (PK1, r1), Note 2 = (PK2, r2), etc…

To preserve privacy, nodes can only store hashes of the notes:

H1 = HASH(Note1), H2 = HASH(Note2), etc…

But, how can we distinguish between unspent notes? Nullifier Set!

Example

So let's say Alice wanted to send a note to Bob with PK4:

1. Randomly choose new serial number r4, and define new note Note4 = (PK4, r4).
2. Send Note4 to Bob privately.
3. Send nullifier of Note 1, nf1 = HASH(r1) to all nodes.
4. Send hash of new note H4 = HASH(Note 4) to all nodes.
5. Each nodes checks whether nullifier exists and adds hash of note and nullifier to the database

But, we didn't check whether note belongs to Alice or exists at all!

Example

So let's say Alice wanted to send a note to Bob with PK4:

1. Randomly choose new serial number r4, and define new note Note4 = (PK4, r4).
2. Send Note4 to Bob privately.
3. Send nullifier of Note 1, nf1 = HASH(r1) to all nodes.
4. Send hash of new note H4 = HASH(Note 4) to all nodes.
5. Each nodes checks whether nullifier exists and adds hash of note and nullifier to the database

But, we didn't check whether note belongs to Alice or exists at all!

Example

6. Alice publishes a proof convincing nodes that whoever published the transaction knows PK1, SK1, and r1 such that:

- HASH(Note 1) = (PK1, r1) exists in set of hashed notes.
- SK1 private key corresponds to PK1.
- HASH(r1) = nf1 nullifier, and nullifier is not currently in nullifier set.

Architecture (Complex)

Zcash: Halo 2 (Proving system) and Orchard (Circuit/Protocol)

Halo 2: Proving system based on PLONK-style circuit arithmetization. Eliminates the trusted setup by using another commitment scheme based on inner products (i.e. bulletproofs).

Orchard Pool: uses the Halo 2 Proving System with Plonkish arithemizations for zero knowledge proofs.

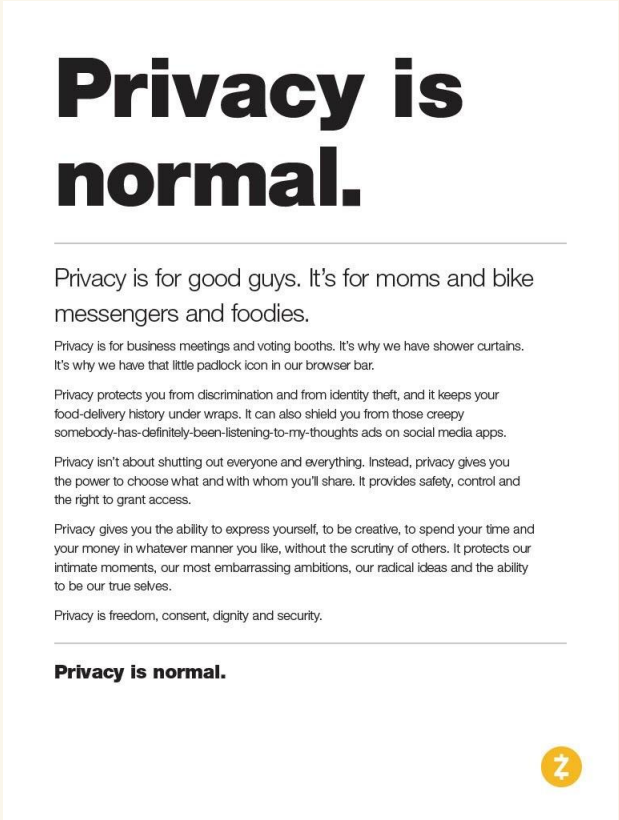
Orchard Shielded Pool
- Note commitments for hiding and binding ownership of ZEC
- Merkle tree used to privately reference the note commitments
- Note nullifers to prevent double spending

Sprout and Sapling shielded pools suffer from two primary issues:

[1] Lack of recursion, which requires cycles of elliptic curves

[2] Requires trusted setup with MPC

References



Privacy is normal.

Privacy is for good guys. It's for moms and bike messengers and foodies.

Privacy is for business meetings and voting booths. It's why we have shower curtains. It's why we have that little padlock icon in our browser bar.

Privacy protects you from discrimination and from identity theft, and it keeps your food-delivery history under wraps. It can also shield you from those creepy somebody-has-definitely-been-listening-to-my-thoughts ads on social media apps.

Privacy isn't about shutting out everyone and everything. Instead, privacy gives you the power to choose what and with whom you'll share. It provides safety, control and the right to grant access.

Privacy gives you the ability to express yourself, to be creative, to spend your time and your money in whatever manner you like, without the scrutiny of others. It protects our intimate moments, our most embarrassing ambitions, our radical ideas and the ability to be our true selves.

Privacy is freedom, consent, dignity and security.

Privacy is normal.

- **Zcash Motto**

References

https://z.cash/technology/zksnarks/

https://electriccoin.co/blog/zcash-private-transactions/

https://grayscale.com/wp-content/uploads/2021/10/grayscale-building-blocks-zcash-october-2021.pdf.pdf

https://www.coindesk.com/policy/2022/02/16/canada-sanctions-34-crypto-wallets-tied-to-trucker-freedom-convoy/

https://www.elliptic.co/blog/bitcoin-transactions-money-laundering

Thank you!

https://sss.cse.lehigh.edu/