qedit

ZKPROOF

Privacy and Compliance – Striking a Delicate Balance

November 2023

Pablo Kogan
Director of Engineering

Abstract – Privacy and Compliance – Striking a Delicate Balance

- In the realm of public blockchains and payment systems, the need for privacy and regulatory compliance often presents a challenging dilemma. This technical talk, aims to explore the possibilities of integrating privacy measures while ensuring compliance with relevant regulations. The session will provide insights into the tools and **cryptographic protocols** that can enable policy makers to strike a delicate balance between privacy and compliance.

- To address these challenges, the session will introduce various cryptographic protocols and technological building blocks specifically tailored for privacy in payment systems. Concepts such as zero-knowledge proofs, secure multiparty computation, and confidential transactions will be explored, showcasing their potential to preserve privacy while ensuring compliance with anti-money laundering (AML) and know your customer (KYC) regulations.

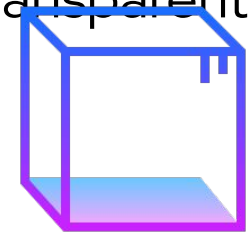
- The talk will also discuss the role of decentralized technologies, such as blockchain and distributed ledger systems, in enhancing privacy and compliance in payment systems. It will highlight the benefits and limitations of these technologies, along with practical considerations for their implementation.

- Attendees will gain a comprehensive understanding of the technical tools and strategies available to design privacy-preserving payment systems that comply with regulatory requirements. They will be equipped with the knowledge to navigate the complexities of privacy-enhancing technologies and make informed decisions when developing or evaluating payment systems.

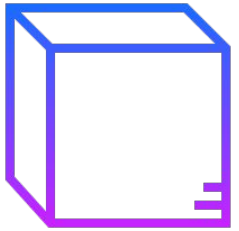
Setting the scene

- The axis of privacy

Fully Transparent



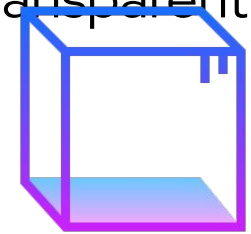
Fully Opaque



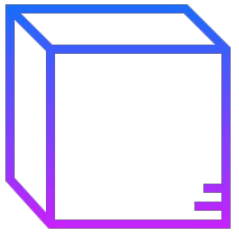
Setting the scene

- The axis of privacy

Fully Transparent



Fully Opaque

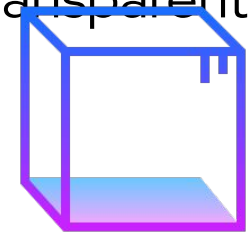


- A DB (Encrypted at rest)
- Unshielded public blockchain (Bitcoin, Eth..)

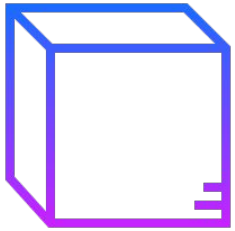
Setting the scene

- The axis of privacy

Fully Transparent



Fully Opaque



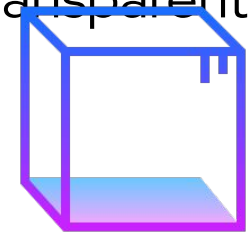
- A DB (Encrypted at rest)
- Unshielded public blockchain (Bitcoin, Eth..)

- Monero
- Zerocash/Zcash
- CT/MimbleWimble

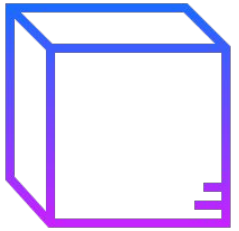
Setting the scene

- The axis of privacy
- Does not imply a distributed system

Fully Transparent



Fully Opaque



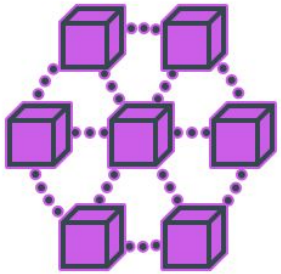
- A DB (Encrypted at rest)
- Unshielded public blockchain (Bitcoin, Eth..)

- Monero
- Zerocash/Zcash
- CT/MimbleWimble

Setting the scene

- Three types of systems

Public blockchain



Payment service

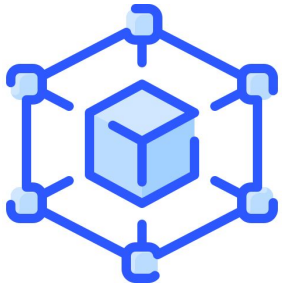
CBDC

Setting the scene

- Three types of systems

Public blockchain

Payment service



CBDC

Setting the scene

- Three types of systems

Public blockchain

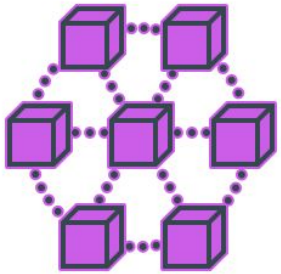
Payment service

CBDC

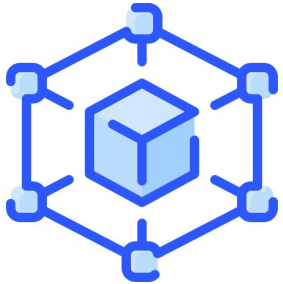
Setting the scene

- Three types of systems

Public blockchain



Payment service



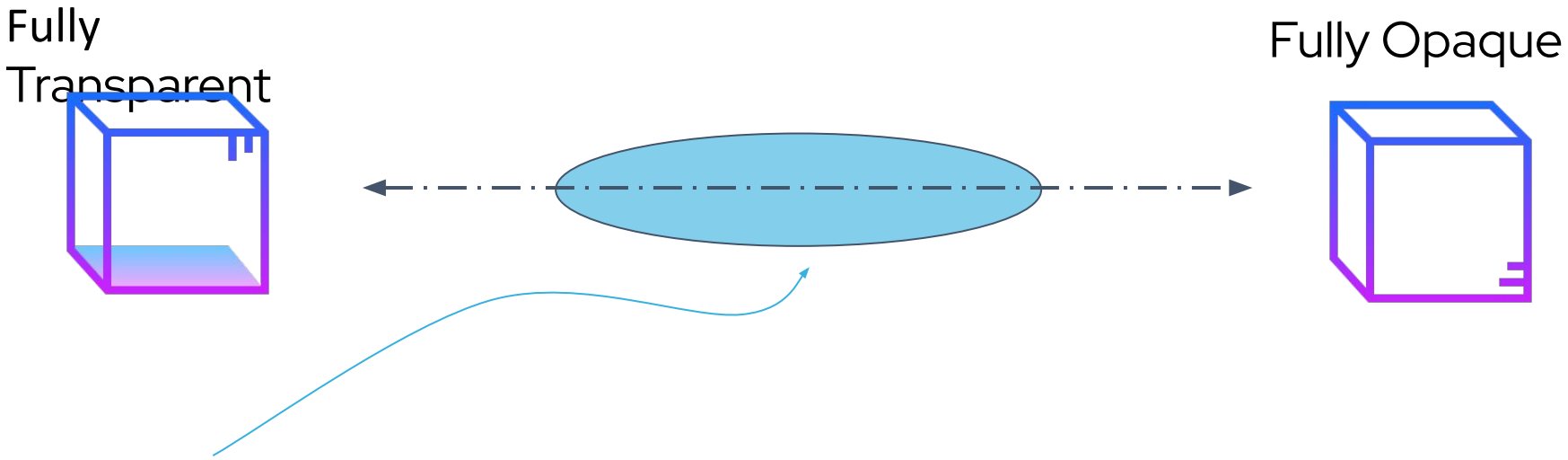
CBDC

Setting the scene

- Different requirements for different jurisdictions

Setting the scene

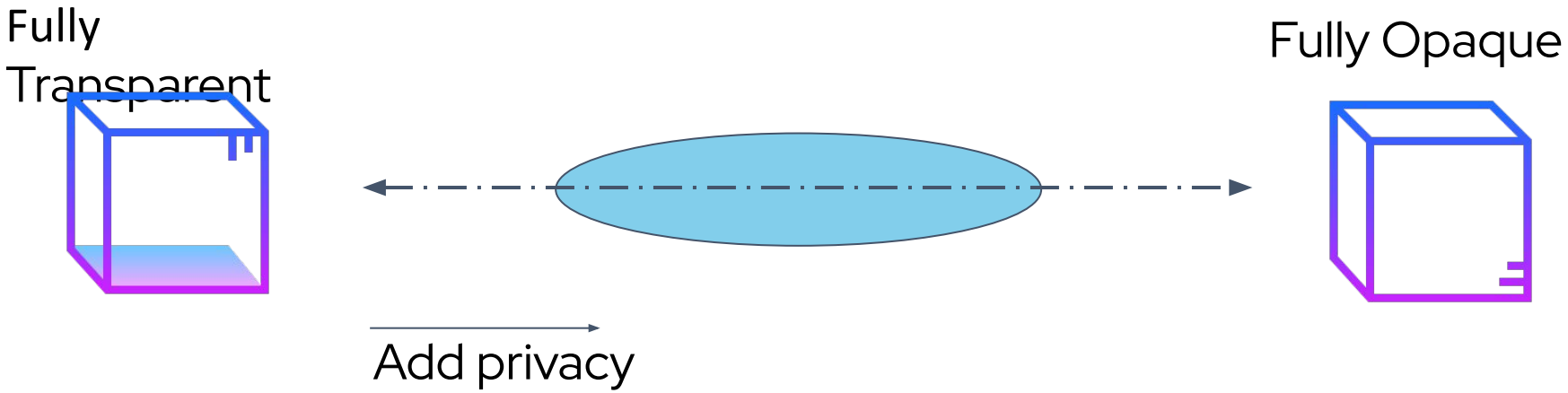
- The axis of privacy



Goal: discovering the “ideal” middle ground between compliance and privacy.

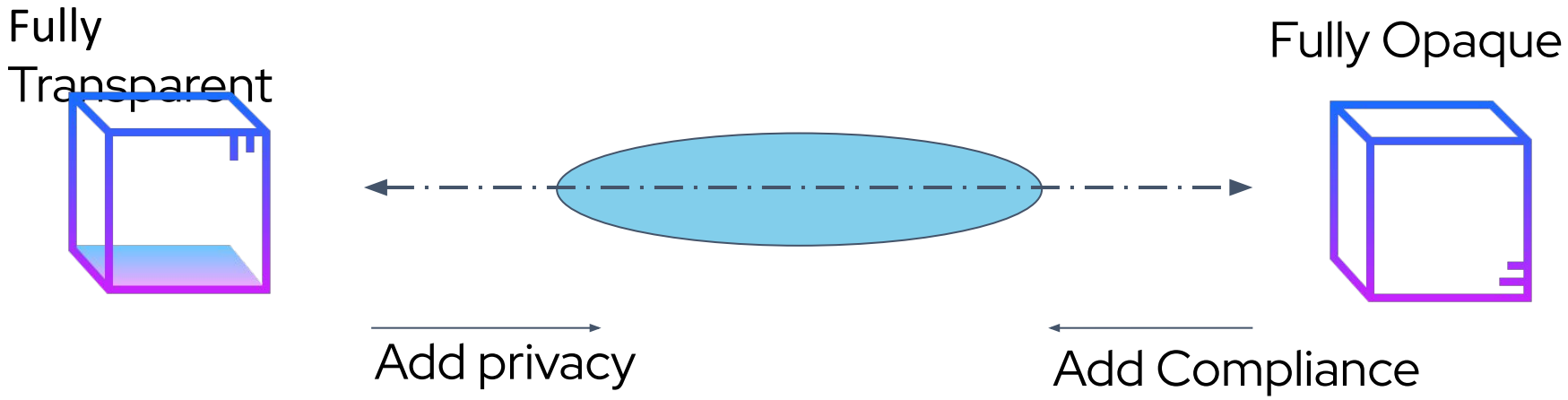
Setting the scene

- The axis of privacy



Setting the scene

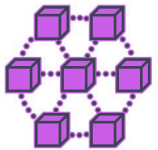
- The axis of privacy



A Fully Opaque system

Example: Zcash/Orchard – a fully opaque blockchain protocol

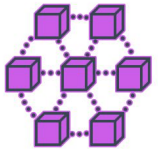
- Privacy achieved using homomorphic commitments, encryption, re-randomized signature scheme and a ZK proof system



A Fully Opaque system

Example: Zcash/Orchard – a fully opaque blockchain protocol

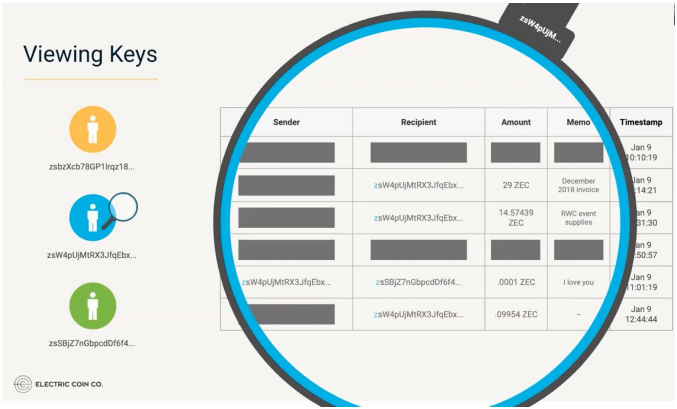
- Privacy achieved using homomorphic commitments, encryption, re-randomized signature scheme and a ZK proof system
- Combines both “shielded” and “unshielded” transactions



Viewing Keys

Example: Zcash/Orchard viewing keys

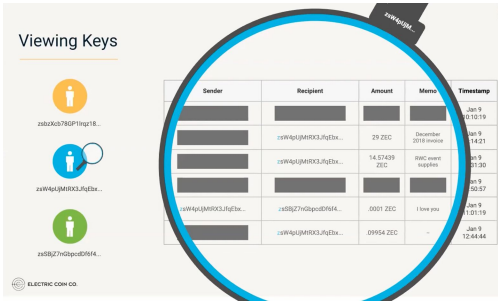
- Provide the ability to disclose transaction history
- Viewing key is separate from the spending key



Viewing Keys

Example: Zcash/Orchard viewing keys

- The viewing key will reveal the entire history

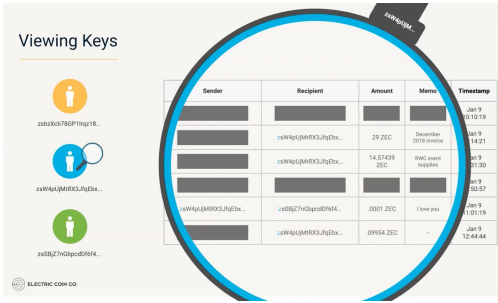


Viewing Keys

Example: Zcash/Orchard viewing keys

- The viewing key will reveal the entire history
- The revealed in/out addresses are

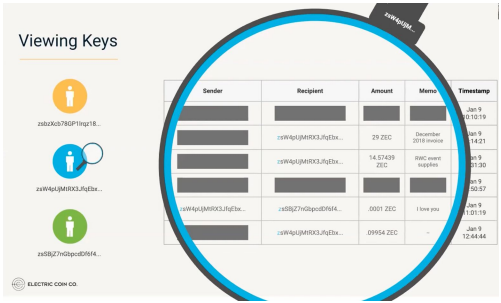
  re-randomized



Viewing Keys

Example: Zcash/Orchard viewing keys

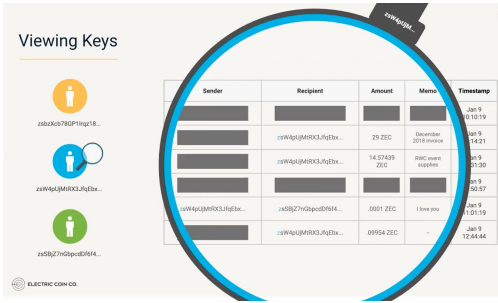
- The viewing key will reveal the entire history
- The revealed in/out addresses are re-randomized
- Correct decryption using the viewing keys is not guaranteed by the consensus.



Viewing Keys

Example: Zcash/Orchard viewing keys, **Potential improvements**

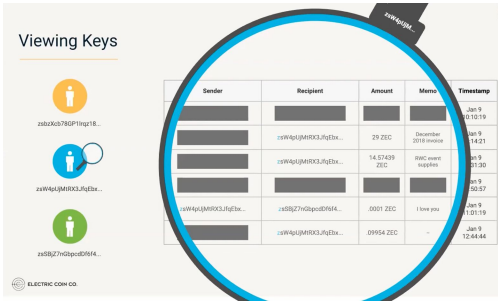
- Enforcement by the protocol



Viewing Keys

Example: Zcash/Orchard viewing keys, **Potential improvements**

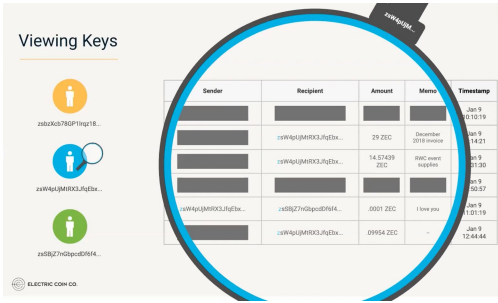
- Enforcement by the protocol
  - Need to ensure properties of the plaintext



Viewing Keys

Example: Zcash/Orchard viewing keys, **Potential improvements**

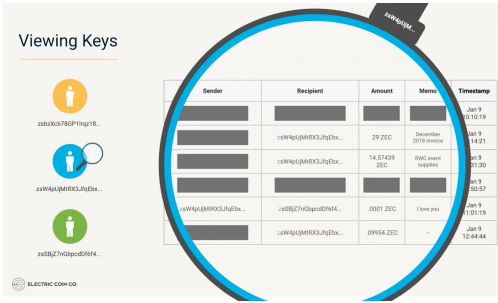
- Enforcement by the protocol
  - Need to ensure properties of the plaintext
  - ZK or Verifiable encryption



Viewing Keys

Example: Zcash/Orchard viewing keys, **Potential improvements**

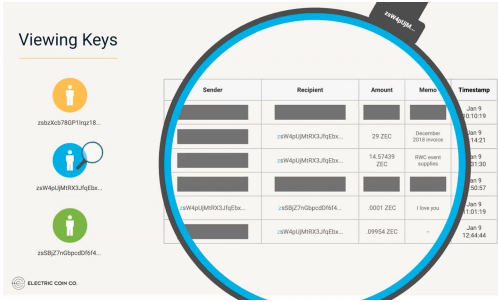
- (Provable) Granularity



Viewing Keys

Example: Zcash/Orchard viewing keys, **Potential improvements**

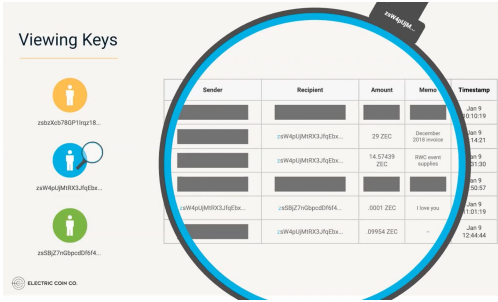
- Disable address re-randomization (On per-asset basis)



Viewing Keys

Example: Zcash/Orchard viewing keys, **Potential improvements**

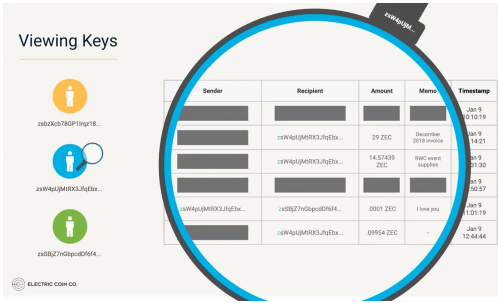
- Global viewing key (On per-asset basis)



Viewing Keys

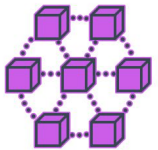
Example: Zcash/Orchard viewing keys, **Potential improvements**

- Global viewing key (On per-asset basis)
- Potentially split between multiple parties (SS)
    - I.E. The regulator and the issuer



Viewing Keys – Applicability

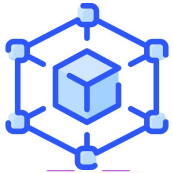
Direct applicability for a “shielded” blockchain



Viewing Keys – Applicability

For a payment provider

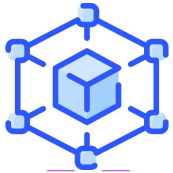
- Decouple operating the service from transaction inspection



Viewing Keys – Applicability

For a payment provider

- Decouple operating the service from transaction inspection
- The regulator will have the global viewing key



Viewing Keys – Applicability

For a CBDC

- Decouple operating the service from transaction inspection
- The operation can be delegated to a company

Blacklist / Whitelist

Compliance feature: Blacklist enforcement

Blacklist / Whitelist

Desired properties:

- Ability to freeze funds for a suspicious address
- Unfreeze if suspicion was removed

Blacklist / Whitelist

Potential approach assuming cryptographic accumulators

- An authority commits the **updated** accumulator digest to the chain

Blacklist / Whitelist

Potential approach assuming cryptographic accumulators

- An authority commits the **updated** accumulator digest to the chain
- Sender produces a non-membership proof for
  - The canonical address

    And / Or
  - The re-randomized address

Blacklist / Whitelist

Potential approach assuming cryptographic accumulators

- An authority commits the **updated** accumulator digest to the chain
- Sender produces a non-membership proof for
  - The canonical address

    And / Or
  - The re-randomized address

Guarantee: Sender is not blacklisted at the time of the transfer.

Blacklist / Whitelist

Advantages

- Privacy preserving (verifier learns nothing except non-inclusion)
- User friendly (if legit, can be removed from the blacklist)

Blacklist / Whitelist

Advantages

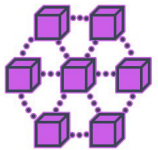
- Privacy preserving
- User friendly

Disadvantages

- Who controls the blacklist? (consider a committee or a per-asset authority)
- Blacklisting re-randomized addresses might not be effective and there is a difficulty deduce the canonical address (consider removing re-randomization)

Blacklist / Whitelist – Applicability

Direct applicability for a shielded Public blockchain



Compliance feature: Asset traceability

Asset traceability

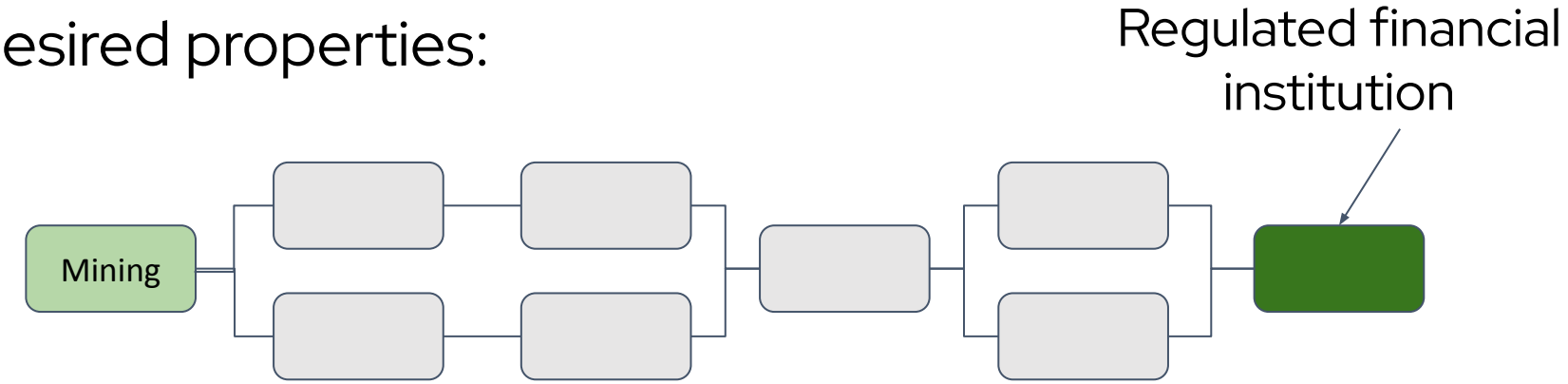
Compliance feature: Asset traceability

Desired properties:

- Ensure funds did not traversed via a malicious hop (post fact)

Asset traceability

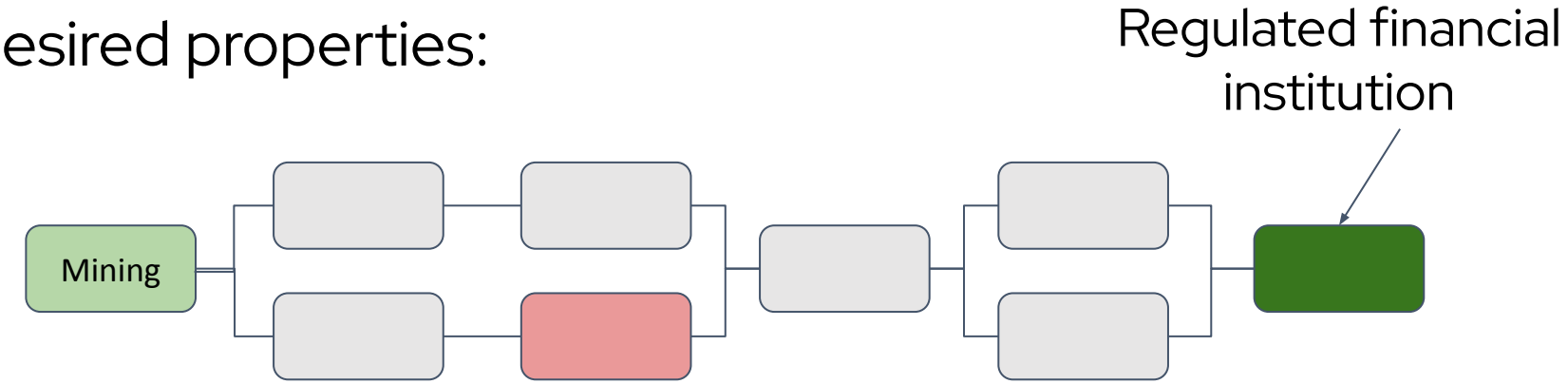
Desired properties:



Compliant

Asset traceability

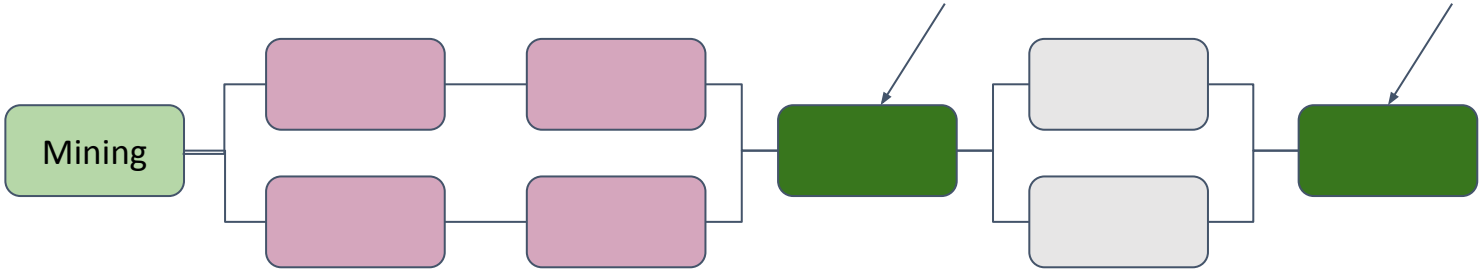
Desired properties:



Non compliant

Asset traceability

Desired properties:

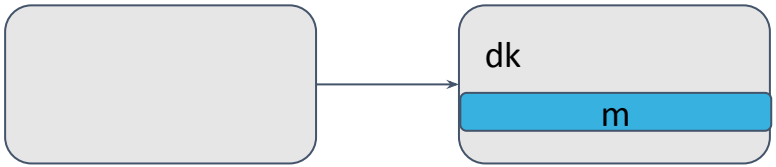


Compliant

Asset traceability

Potential approach assuming a “shielded” blockchain:

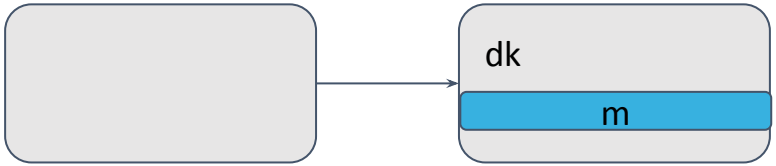
- Encrypt and attach the **canonical identity (ID)** as part of each transaction
- Prove correctness using Verifiable encryption / ZK
- $m=Enc_k(ID)$, $dk = f(ID)$
- k



Asset traceability

Who should have the key **k**?

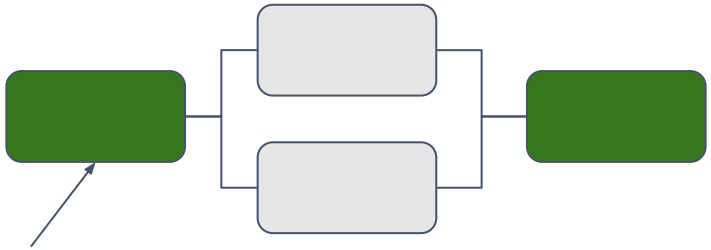
- Only the user – ***elective disclosure***



Asset traceability

Who should have the key **k**?

- Only the user – ***elective disclosure***
- Good enough to prove compliance for one hop

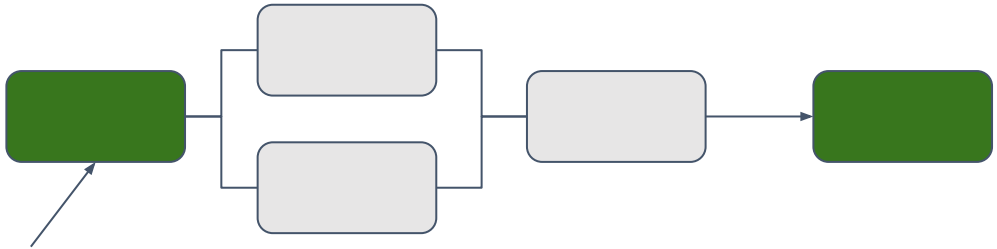


Mining or regulated

Asset traceability

Who should have the key **k**?

- The issuer

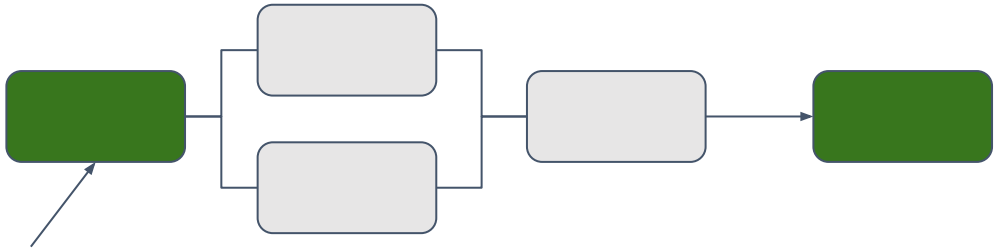


Mining or regulated

Asset traceability

Who should have the key **k**?

- The issuer
- The regulator

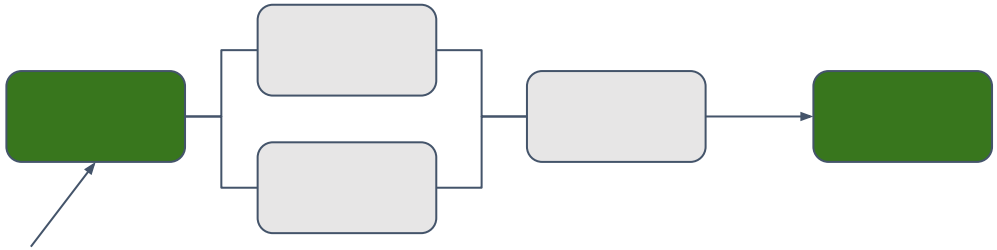


Mining or regulated

Asset traceability

Who should have the key **k**?

- The issuer
- The regulator
- Secret shared between both

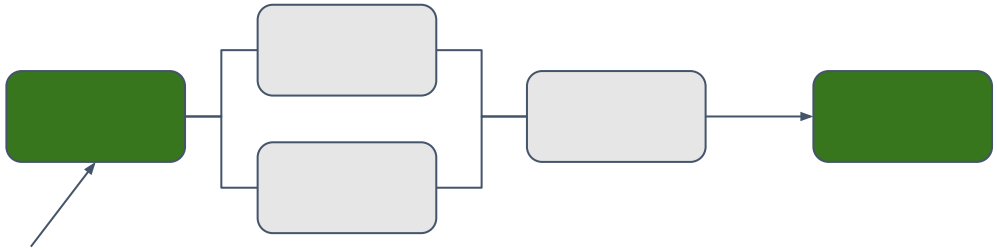


Mining or regulated

Asset traceability

Who should have the key **k**?

- The issuer
- The regulator
- Secret shared between both
- A committee



Mining or regulated

Privacy budget

Provide the ability to create auditable and un-auditable transactions

Privacy budget

Provide the ability to create auditable and un-auditable transactions

Desired properties:

- Large transactions can be audited (similar to bank transactions)
- Small transactions can not be audited (similar to cash)

Privacy budget

UTT: Decentralized Ecash with Accountable Privacy

Alin Tomescu* Adithya Bhat† Benny Applebaum‡ Ittai Abraham§
Guy Gueta§ Benny Pinkas¶ Avishay Yanai§

April 9, 2022

Privacy budget

UTT: Decentralized Ecash with Accountable Privacy

Alin Tomescu* Adithya Bhat† Benny Applebaum‡ Ittai Abraham§
Guy Gueta§ Benny Pinkas¶ Avishay Yanai§

April 9, 2022

Two events for a digital token:

- Mint event: a token is **minted** by being **signed** by the Central Bank
- Burn event: a token is **nullified** by being **added** to Central Bank **nullifier list**

Privacy budget

UTT: Decentralized Ecash with Accountable Privacy

Alin Tomescu* Adithya Bhat† Benny Applebaum‡ Ittai Abraham§
Guy Gueta§ Benny Pinkas¶ Avishay Yanai§

April 9, 2022

For privacy, UTT uses **re-randomizable signatures**:

- Users can re-randomize the original signatures they received and generate a new signature for the same token
- This new signature can be verified, but is un-linkable to the original signature

An example UTT transaction

Alice has two tokens of $50 and wants to pay Bob $70. Alice has a privacy budget of $500

An example UTT transaction

Alice has two tokens of $50 and wants to pay Bob $70. Alice has a privacy budget of $500

- Three incoming tokens:
  - T_1 – first token of Alice (with value of $50)
  - T_2 – second token of Alice (with value of $50)
  - BT – privacy budget token of Alice (with value of $500)
- Three outgoing tokens :
  - T_b – outgoing token for Bob (with value of $70)
  - T_a – outgoing token for Alice (with value of $30)
  - BT' - new privacy budget token of Alice (with value of $430)

An example UTT transaction

Alice has two tokens of $50 and wants to pay Bob $70. Alice has a privacy budget of $500

- Three incoming tokens:
  - T_1 – first token of Alice (with value of $50)
  - T_2 – second token of Alice (with value of $50)
  - BT – privacy budget token of Alice (with value of $500)
- Three outgoing tokens :
  - T_b – outgoing token for Bob (with value of $70)
  - T_a – outgoing token for Alice (with value of $30)
  - BT' - new privacy budget token of Alice (with value of $430)

The proof verifies that $\mathbf{T_1+T_2=T_a+T_{b'}}$ and that $\mathbf{BT'=BT-T_b}$

Privacy budget

An example UTT transaction

Alice has two tokens of $50 and wants to pay Bob $70. Alice has a privacy budget of $500

- Three incoming tokens:
  - T_1 – first token (with value of $50)
  - T_2 – second token (with value of $50)
  - BT – privacy budget token (with value of $500)
- Three outgoing tokens :
  - T_b – outgoing token for Bob (with value of $70)
  - T_a – outgoing token for Alice (with value of $430)
  - BT' - new privacy budget token of Alice (with value of $430)

The proof verifies that $\mathbf{T_1+T_2=T_a+T_{b'}}$ and that $\mathbf{BT'=BT-T_b}$

Pilot by VMware and BOI led by Ittai Abraham

Privacy budget – applicability

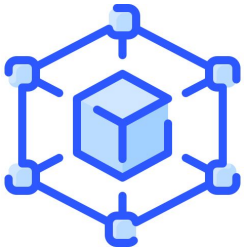
In the context of a **CBDC**

- A well-defined central authority to assign the privacy tokens
- A well-defined set of approved users on the receiving end

Privacy budget – applicability

In the context of a **payment system**

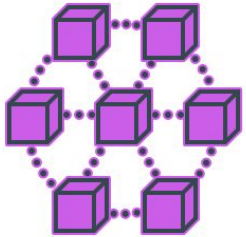
- A well-defined central authority to assign the privacy tokens
- A well-defined set of approved users on the receiving end



Privacy budget – applicability

In the context of **a public blockchain**

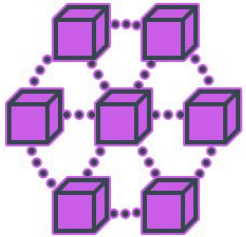
- A central authority is replaced by consensus to provably assign the privacy tokens



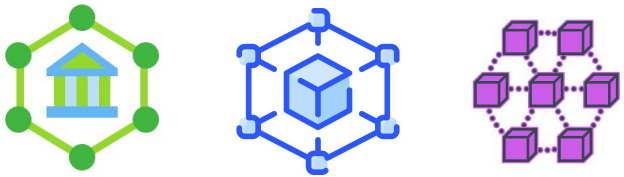
Privacy budget – applicability

In the context of **a public blockchain**

- A central authority is replaced by consensus to provably assign the privacy tokens
- Require some sort of **digital identity**

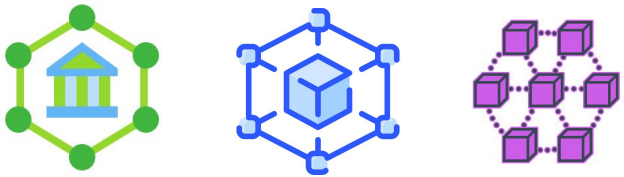


Methodology for real world testing



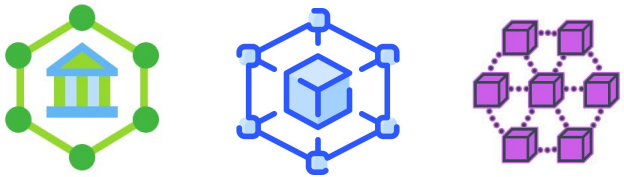
Methodology for real world testing

- No one-size fits-all solution



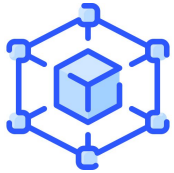
Methodology for real world testing

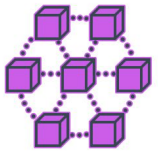
- Using an untested protocol for a CBDC deployment is undesirable.



Methodology for real world testing

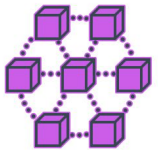
- Deploy a new public blockchain or payment system





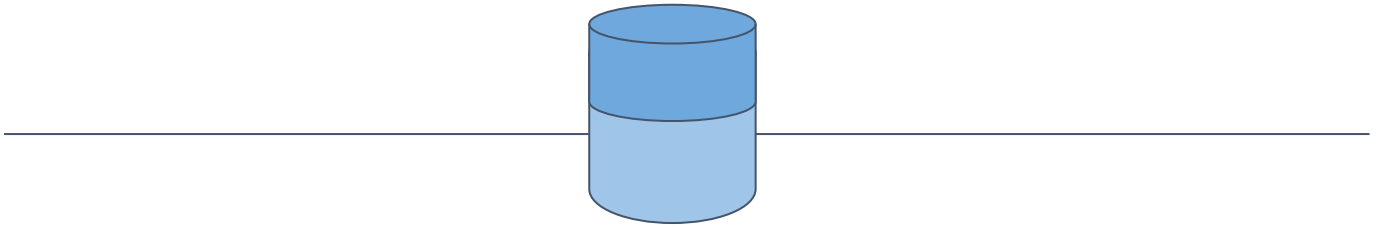
Methodology for real world testing

- Host multiple assets with different privacy/compliance guarantees on a single blockchain



Methodology for real world testing

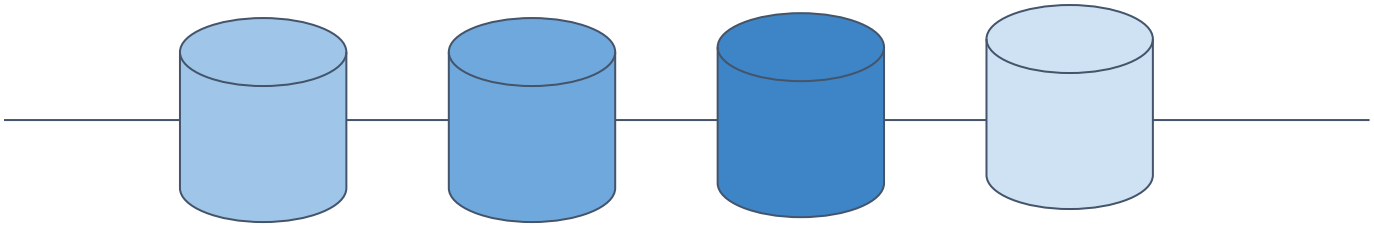
- Host multiple assets with different privacy/compliance guarantees on a single blockchain



Control compliance feature using issuance flags

Methodology for real world testing

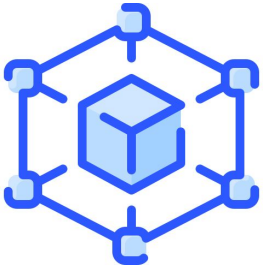
- Host multiple assets with different privacy/compliance guarantees on a single blockchain

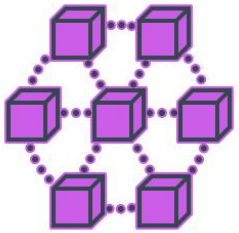


Same blockchain, multiple shielded pools

Conclusion

- We have the cryptographic building blocks to balance compliance and privacy.





Thank you!

