BLOCKCHAINS

ARCHITECTURE, DESIGN AND USE CASES

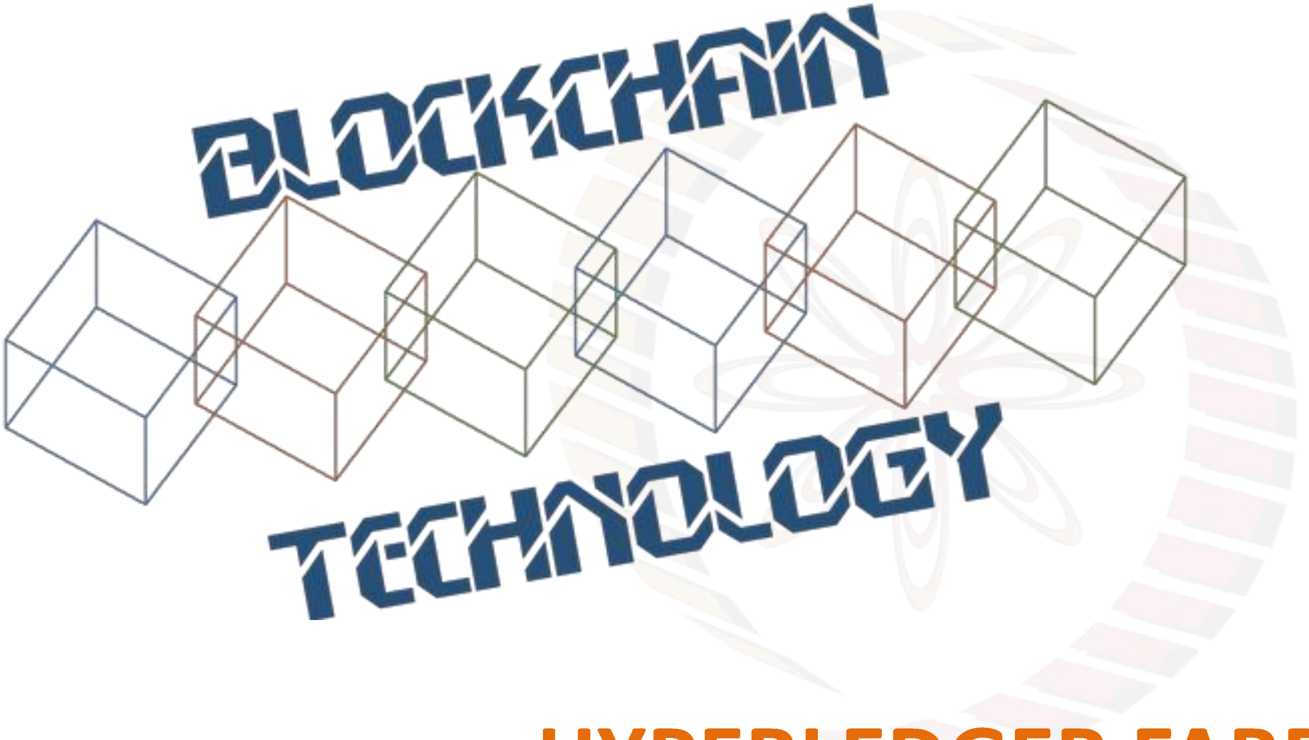
SANDIP CHAKRABORTY

COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

PRAVEEN JAYACHANDRAN

IBM RESEARCH,
INDIA

Image courtesy: http://beetfusion.com/



HYPERLEDGER FABRIC - PRIVACY

Privacy in a Blockchain System

- Transaction Data Privacy: Transactional activity of an entity (profiling of the transactors)
- State Data Privacy: Chaincode / smart contract data (data that the smart contract alters)
- Smart Contract Privacy: Logic of the chaincode / smart contract (e.g., business logic)
- User Privacy: Anonymity and Unlinkability

- None of these aspects of privacy are supported by permissionless blockchain platforms including Bitcoin and Ethereum, except pseudo-anonymity; Applications have to explicitly handle them
- Important aspect of permissioned blockchain platforms targeting enterprise applications

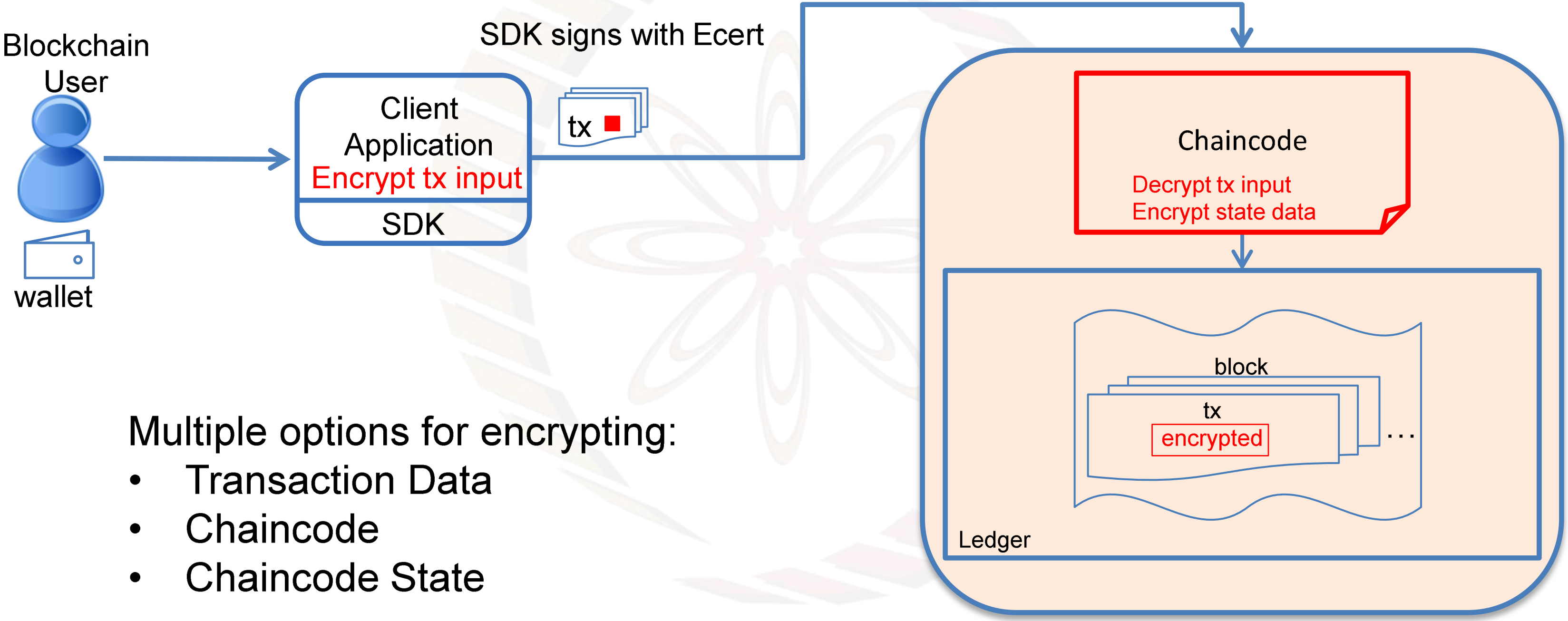
Privacy Using Channels in Hyperledger Fabric

- Channel:
  - Partitioning mechanism implementing its own total broadcast mechanism
  - Channel transaction ordering takes place independently of other channels and by the members of the channel
- Channel creation upon properly authenticated & authorized request
- Channel creation *request submitted to & evaluated by the ordering service*
  - Participation is restricted to a subset of organizations/participants
  - Participation is defined by means of [Reader policies, Writer policies, Admin policies]
- Provides transaction and data privacy to members of the channel; All members of a channel see all transaction and data within it
- **Note**: Data privacy is **not offered** w.r.t. ordering service with the use of channels

Data Privacy Using Encryption

- Can be **supported at the application layer** by:
  1. Application sends all data encrypted to chaincode and is stored as-is; keys managed by application
     - Even peer does not see unencrypted data
     - Disadvantage: Chaincode cannot manipulate/validate data easily
     - Alternative: Store data in an external data store and just have hash and metadata information on blockchain for immutability
  2. Encryption key sent via the *transient data field* inside the chaincode proposals
     - Chaincode data stored encrypted on peer, but chaincode can decrypt data
     - Fabric provides an encryption library, but onus on application and chaincode to encrypt/decrypt data
     - Combined with a trusted execution platform such as Intel SGX or IBM Z Secure Containers, its possible to lock down the chaincode and ensure that unencrypted data is not available outside the chaincode to even the peer

Data Privacy Using Encryption within Chaincode



Multiple options for encrypting:

- Transaction Data
- Chaincode
- Chaincode State

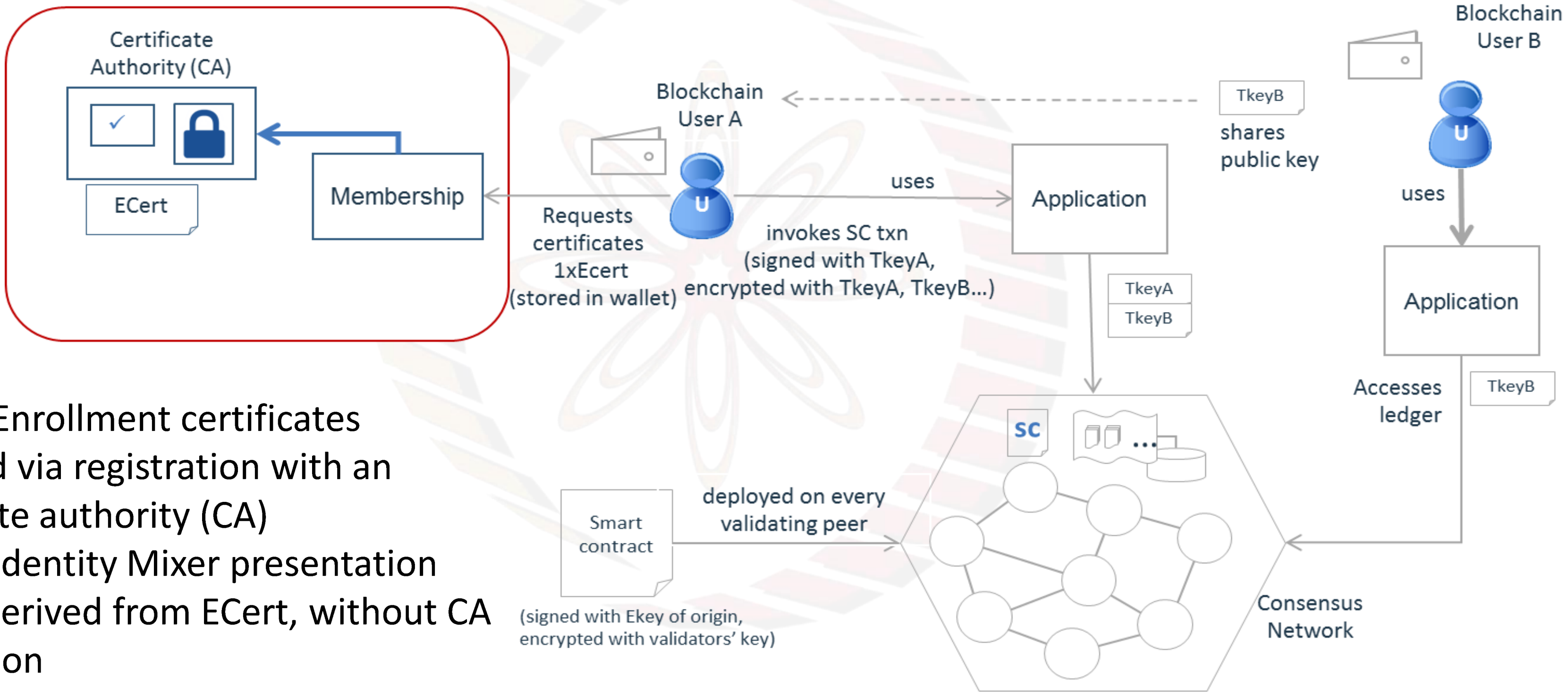
Chaincode Transient Data

- **Transient data:**
  - Field of proposals that is not included in the endorsement message/resulting transaction
  - Provided to the chaincode upon chaincode request
- **How to leverage it?**
  - Transient data can be leveraged to transfer to the chaincode confidential data, e.g., key-material, source of randomness

Smart Contract Confidentiality

- Two step chaincode deployment process: installation & instantiation
- Channel-free installation on a selected set of nodes
- Channel-specific instantiation of a chaincode solely by properly authorised identities & to peers with comply with the logic's trust model
  - Chaincodes can also be hardcoded with keys that they could use to generate per-chain keys to use for state encryption
- An instantiated chaincode's execution is restricted to a set of nodes, whose endorsement is required for the execution results to be committed; these nodes should comply with the executed logic's trust model
- Remember: Not all peers in a channel need to be endorsers for a chaincode, and set of endorsers for different chaincodes can be different
- Endorsement model provides resiliency to non-deterministic chaincode, a possible source of DoS attacks

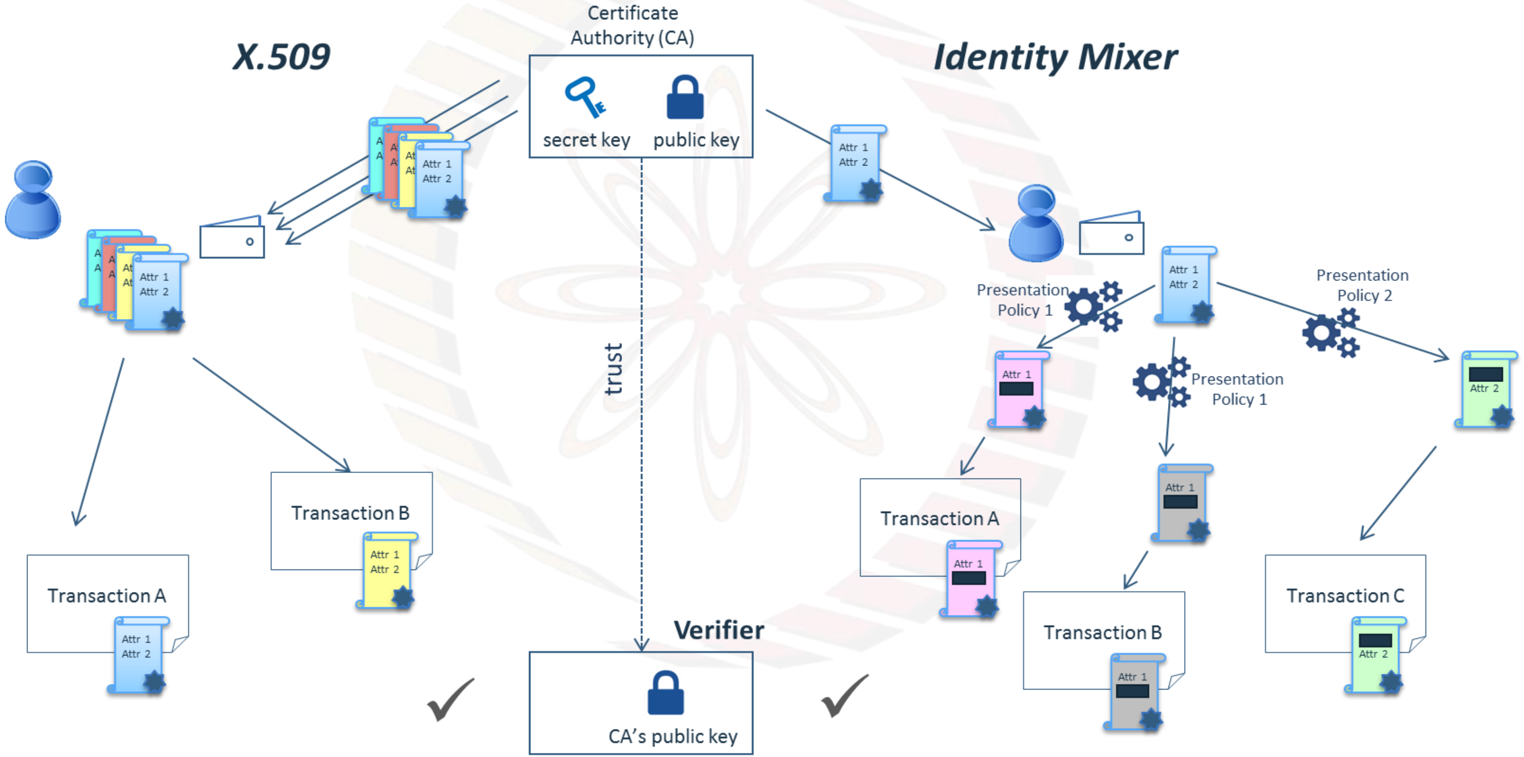
Anonymous and Unlinkable Transactions (Identity Mixer)


Certificate Authority (CA)
ECert
Membership
Requests certificates 1xEcert (stored in wallet)
Blockchain User A
U
uses
Application
invokes SC txn (signed with TkeyA, encrypted with TkeyA, TkeyB…)
TkeyA
TkeyB
TkeyB
shares public key
Blockchain User B
U
uses
Application
Accesses ledger
TkeyB
SC
…
Smart contract
deployed on every validating peer
(signed with Ekey of origin, encrypted with validators' key)
Consensus Network

ECerts: Enrollment certificates acquired via registration with an certificate authority (CA)

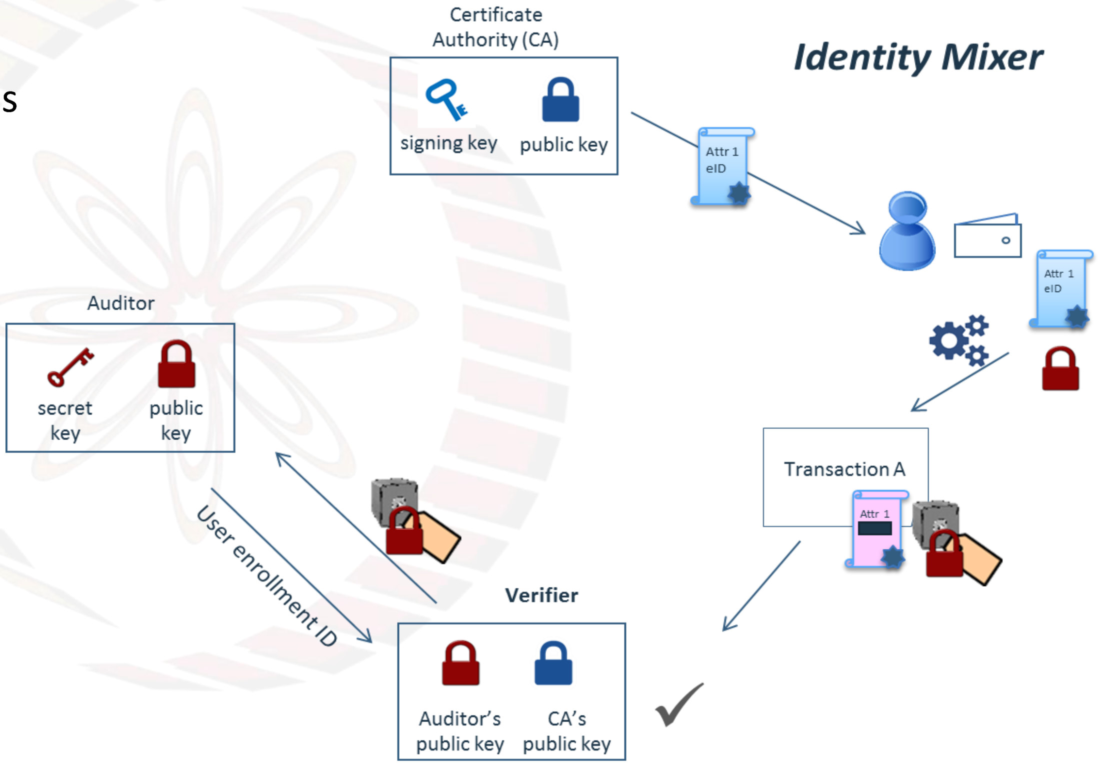
TCerts: Identity Mixer presentation proofs derived from ECert, without CA interaction

X.509 vs Identity Mixer



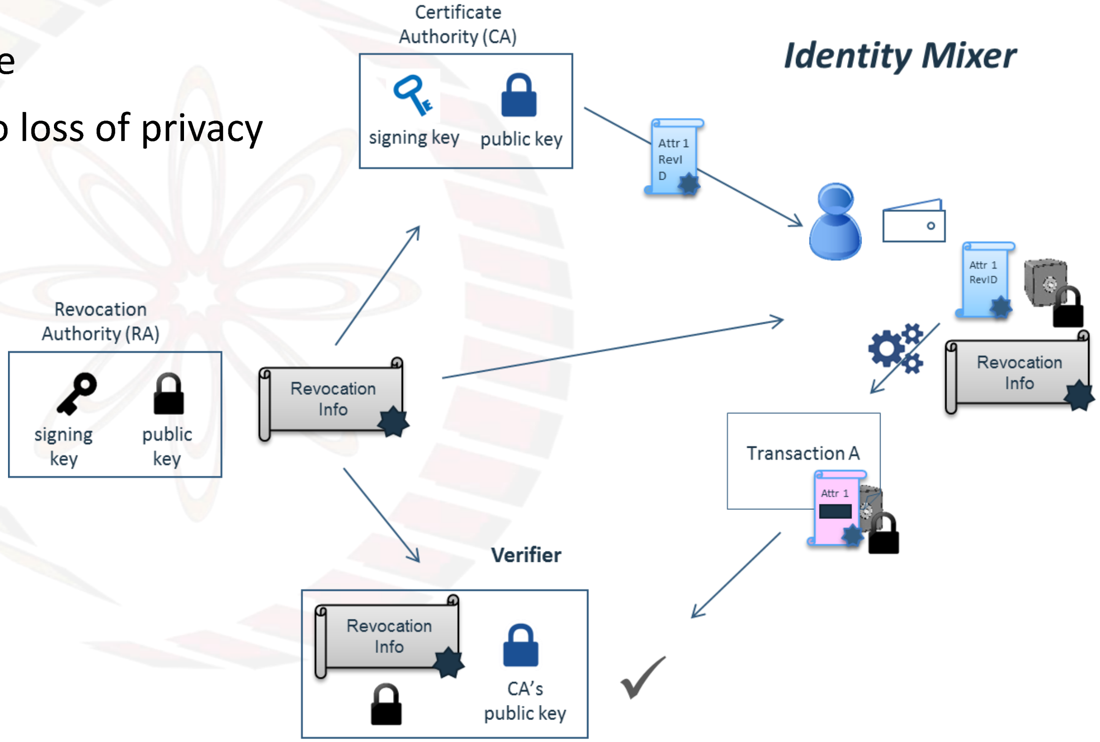
Anonymous and Unlinkable Transactions: Auditability

- Only Auditor can track the transactions
- Auditor’s secret key can be shared between multiple parties to distribute the trust

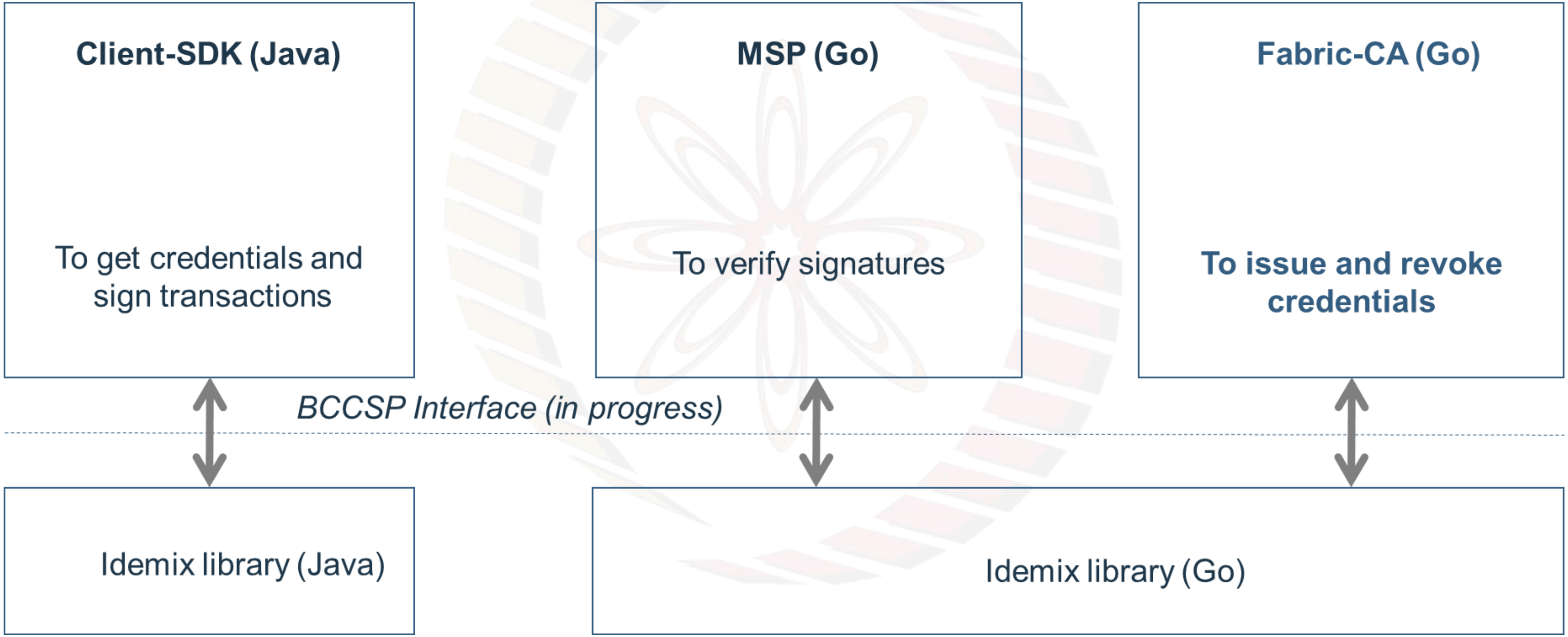


Anonymous and Unlinkable Transactions: Revocation

- Certificates can be revoked at any time
- Non-revocation proof is unlinkable: no loss of privacy for non-revoked users



Identity Mixer Integration



Client-SDK (Java)

To get credentials and sign transactions

MSP (Go)

To verify signatures

Fabric-CA (Go)

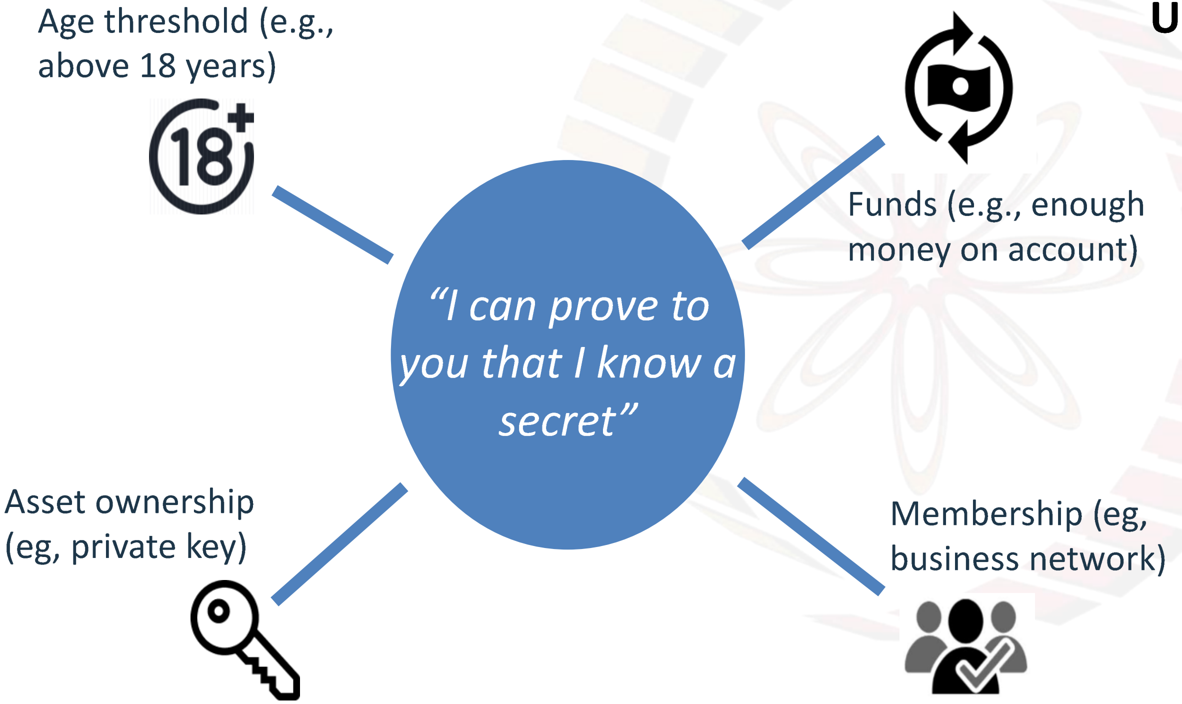
To issue and revoke credentials

BCCSP Interface (in progress)

Idemix library (Java)

Idemix library (Go)

Privacy with Zero-Knowledge Proof Cryptography



"I can prove to you that I know a secret"

Age threshold (e.g., above 18 years)

Funds (e.g., enough money on account)

Asset ownership (eg, private key)

Membership (eg, business network)

Used in/with Fabric

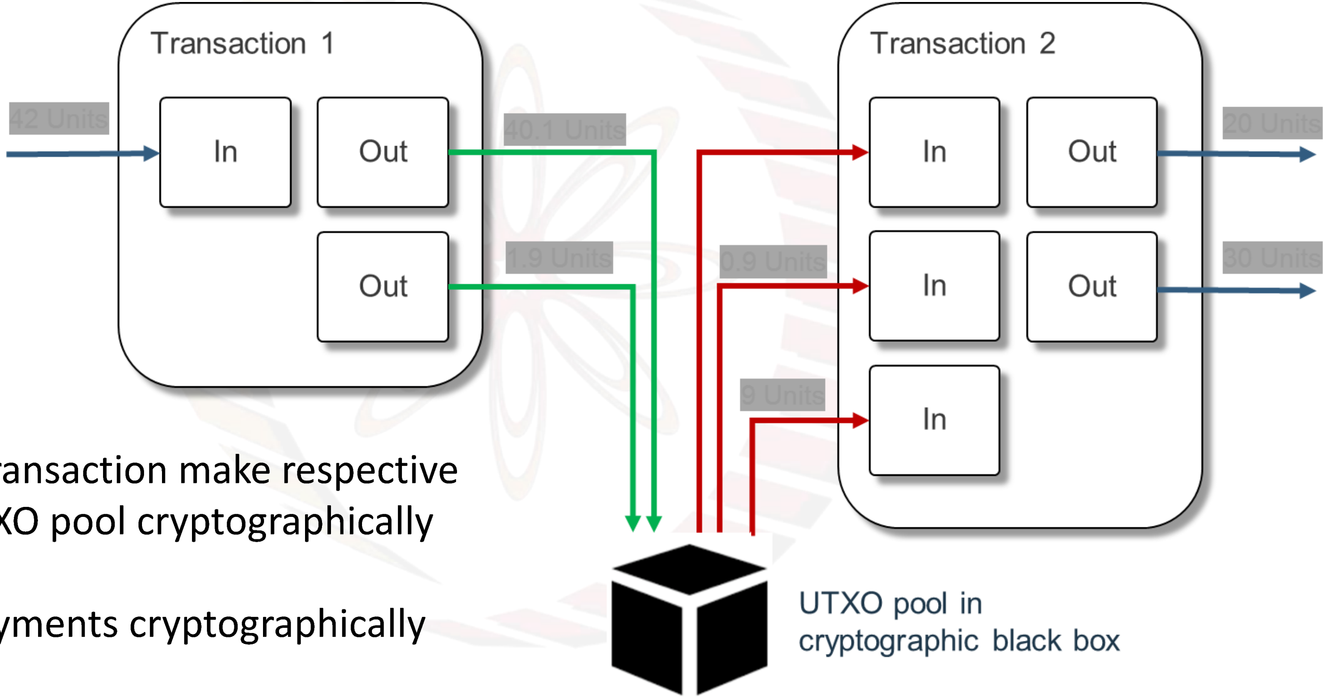
1. Identity Mixer TCerts (Secret: user certified identity & its ownership info)
2. Hyperledger Indy (Secret: identity properties)
3. Possible to integrate with transaction validation

Constructing proof is time consuming, verifying is simple

ZeroCash: UTXO Ownership Model with Privacy

- Zero Knowledge Proof for:
  - Full anonymity based on ZK-SNARKS (not just pseudoanonymity); prove you have private key
  - Concealing asset value transferred (inputs and outputs values in transaction); prove transaction is not double spending or generating coins
  - Conceal UTXO graph
- ZeroCoin cryptocurrency as extension to Bitcoin
- Improved in 2013 as ZeroCash, with 98% smaller proof sizes
- Time taken and size of proofs considered a concern
- Integrated with Ethereum, Quorum (variant of Ethereum for permissioned networks); ability to verify ZK-SNARKS on-chain

UTXO Model with Privacy



- Inputs of a valid transaction make respective outputs in the UTXO pool cryptographically unspendable
- Correctness of payments cryptographically enforced

Fun Reading

- TEDx talk, Using bitcoin blockchain to detect fraud: https://www.youtube.com/watch?v=507wn9VcSAE (linking transactions on bitcoin blockchain)
- Identity Mixer, website: https://www.zurich.ibm.com/identity_mixer/
  - Overview: https://www.zurich.ibm.com/pdf/csc/Identity_Mixer_Nov_2015.pdf
  - Github: https://github.com/IBM-Cloud/idemix-issuer-verifier
- Zero knowledge proof, Wikipedia: https://en.wikipedia.org/wiki/Zero-knowledge_proof
- Zerocash project: http://zerocash-project.org/
  - Research paper in IEEE Symposium on Security and Privacy, 2014: http://zerocash-project.org/media/pdf/zerocash-oakland2014.pdf

thank you!