Blockchains and Authorization in Constrained IoT Environments

Vasilios A. Siris
Mobile Multimedia Laboratory
Athens University of Economics and Business, Greece
vsiris@aueb.gr
panel “Blockchain for IoT”
IoT Week 2019, 20 June 2019, Aarhus, Denmark

EU H2020 SOFIE: Secure Open Federation for Internet Everywhere
Motivation and challenges

- Why constrained IoT environments?
- Why or why not blockchains? Which type of blockchain?
- Goal: identify and quantify tradeoffs in terms of transaction cost, transaction delay, trust, and privacy

Challenges

- Transaction cost and delay
- Fully decentralized solution
- Ensuring that IoT devices actually provide promised access
- Constrained client devices & constrained IoT resource devices

Single public ledger not enough

Blockchain interaction with real world is a challenge

vsiris@aueb.gr
Why constrained IoT environments?

- Because many IoT devices are constrained in terms of
  - processing and storage resources
  - network connectivity

Reducing usage also reduces power consumption & security threats

Scalability of IoT systems can be addressed by utilizing device-to-device communication

Device-to-device technologies exist and are becoming mature

New challenge: how to achieve trusted device-to-device communication

vsiris@aueb.gr
Why/which type of blockchains?

- **Decentralized trust**, i.e. no single trusted third party
  - Public ledgers: *wide-scale decentralized trust*
  - Permissioned ledgers: *degree of trust* determined by permissioned set

- **Immutability**
  - related to first point, majority of nodes need to agree to change state
  - depending on scenario, can be achieved by other means

- **Transparency**
  - not only a feature but a *requirement* for decentralized trust
  - tradeoff with *privacy*

- **Availability**, through *decentralized storage and execution*
  - can be achieved other ways

vsiris@aueb.gr
Constrained IoT resources

- IoT resource has limited processing, storage and only D2D connectivity
- Authorization Server (AS) handles requests on behalf of IoT resource
  - OAuth 2.0 authorization framework
  - Based on access tokens
- Client and AS always connected and can interact with blockchain

vsiris@aueb.gr
Two approaches with one blockchain

1. Record only hashes
2. Smart Contract
Single blockchain: execution cost

• Smart contract requires 2.5 times EVM gas compared to simply recording hashes
• Only write transactions cost gas
  • Reading data has zero cost
• Quantifies cost for higher functionality of smart contracts
  • Authorization policies & logic

vsiris@aueb.gr
Smart contracts and two blockchains

1. Record only hashes
2. Smart Contract
3. Smart Contract & two blockchains

vsiris@aueb.gr
Execution cost

- Two blockchains achieve lower cost compared to one
  - Only payment transaction on public ledger

- Tradeoffs
  - Two ledgers: trust, transparency, and privacy for authorization transactions determined by permissioned node set
  - Public ledger: wide-scale decentralized trust and transparency

vsiris@aueb.gr
• Up to now assumed that client
  • has continuous connectivity
  • Interacts directly with blockchain
• Client AS can, on behalf of client,
  • interact with ledger
  • Interact with IoT resource AS
• Client must obtain authorization information from client AS at some prior instance (asynchronously)
Connected resource & disconnected client

- Connected IoT resource acts as relay for disconnected client
Challenges

- High cost & delay incurred by blockchains
  - Due to public ledger
  - Combining public & private/permissioned ledgers can provide different tradeoffs of cost, trust, and privacy
  - Off-chain transactions: unidirectional payment channels sufficient for some IoT applications

- Single AS
  - Blockchain advantages are limited to assets & transactions residing in the blockchain
  - Once we traverse blockchain boundaries we loose these benefits
  - Solely adding multiple ASes not a solution because IoT resource not directly connected to blockchain
  - Need processing at client to reduce data & ensure trust with constrained IoT resource

Move smart contract to permissioned ledger and/or only record hashes on public ledger

Achieved by combining public with private/permissioned ledger

vsiris@aueb.gr
Challenges (cont)

• Trust that resource indeed provides access
  • Trusted Execution Environments (TEEs) such as ARM’s TrustZone, Intel’s SGX, Keystone (open source RISC V)

• Constrained clients
  • Need client proxy/agent (analogous to AS acting as proxy of IoT resource)

Further info: https://mm.aueb.gr/blockchains/