CHALMERS
UNIVERSITY OF TECHNOLOGY

IoTLogBlock: Recording Off-line Transactions of Low-Power IoT Devices Using a Blockchain

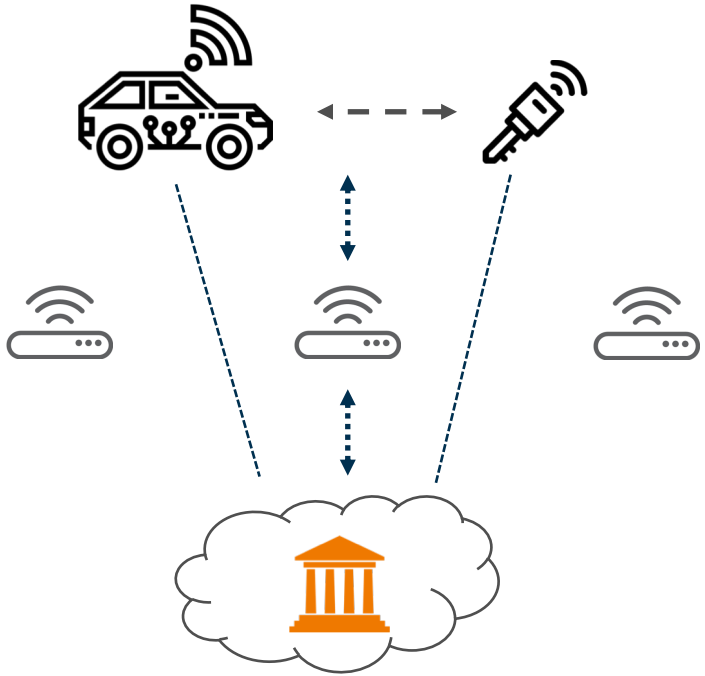
Christos Profentzas Chalmers University

Olaf Landsiedel Kiel University

Magnus Almgren Chalmers University

The Application Scenario

- Rent a smart car with a smart key
- Device trust?
- Centralized practical?
- IoT devices constraints:
  - Intermittent connectivity
  - Energy consumption
  - Computational capabilities
  - Limited memory



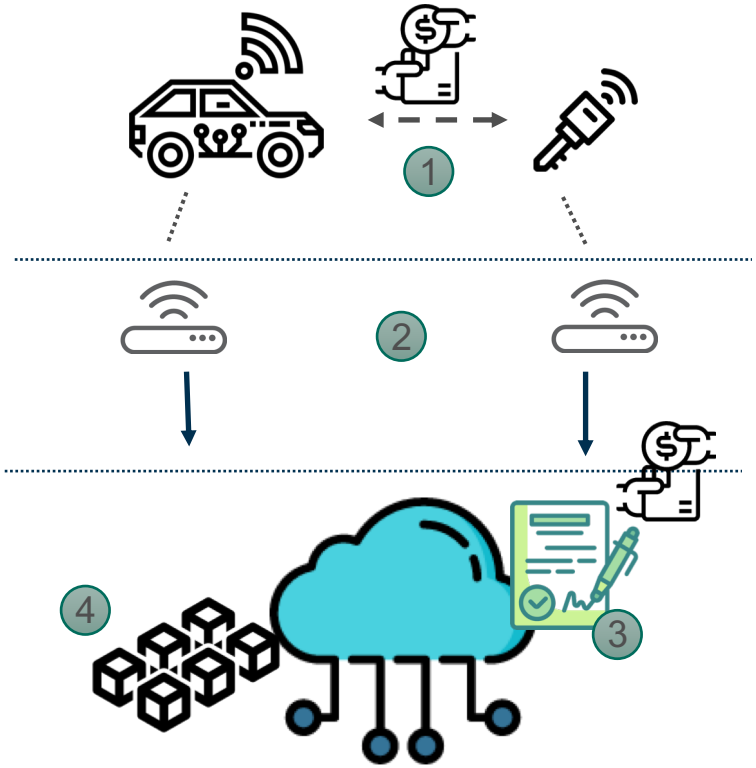
Outline

- IoT & Blockchain
- System Design
- Experiment Evaluation & Results

IoT & Blockchain - IoTLogBlock

- Devices need to work off-line

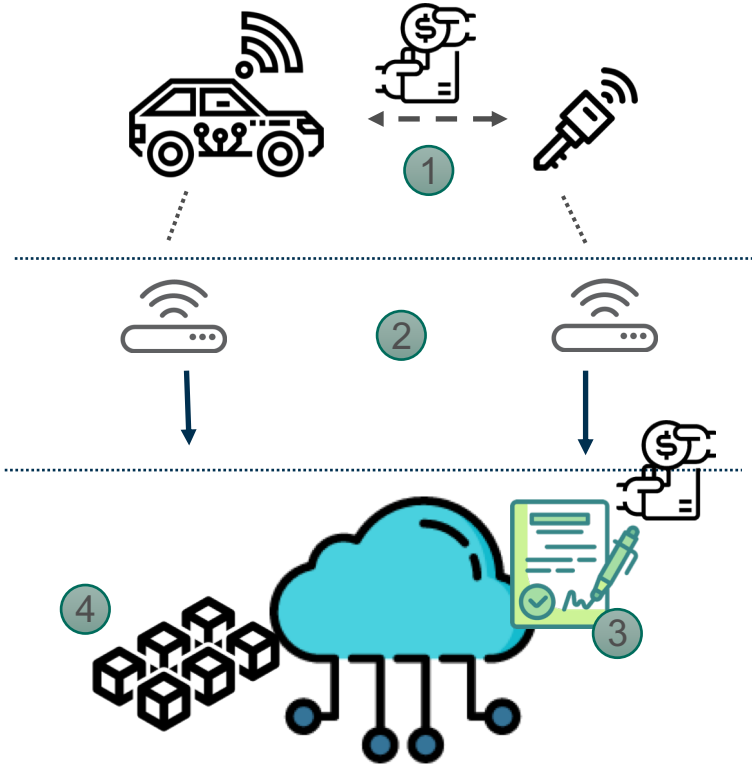
1. Lightweight protocol to sign transactions
2. Edge communication
3. Verify with a smart contract
4. Blockchain storage



IoTLogBlock - Contributions

- Devices need to work off-line

1. **Lightweight protocol to sign transactions**
2. Edge communication
3. **Verify with a smart contract**
4. **Blockchain storage**

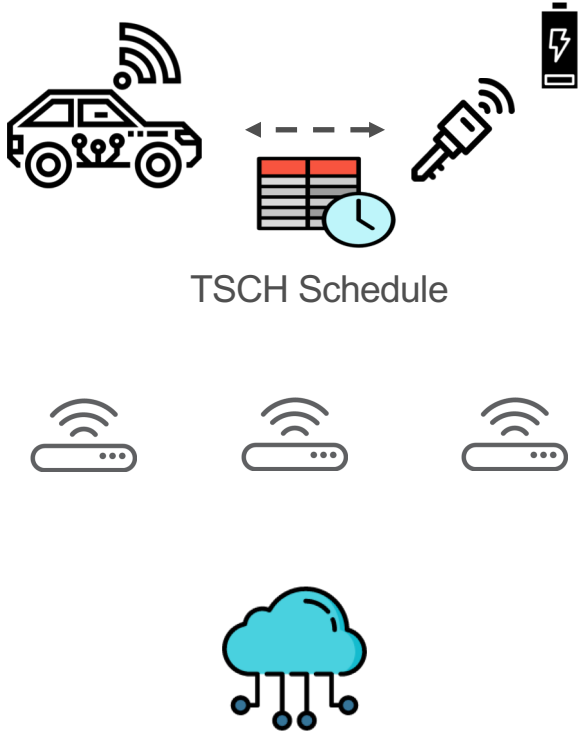


Outline

- IoT & Blockchain
- System Design
- Experiment Evaluation & Results

Design Details: Node Communication

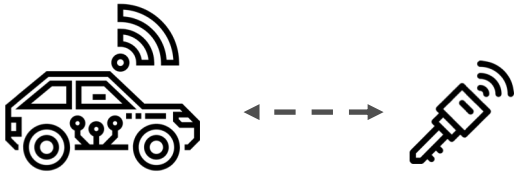
- Device discovery
- Low power protocols
- Sporadic edge connection
- Edge devices just a relay



Design Details: Create & Sign Transactions

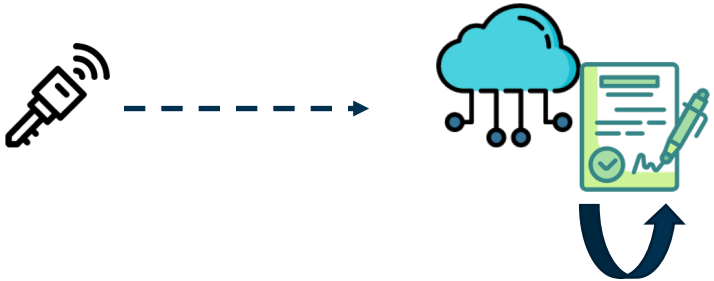
- Keep monotonic counters
- Uniquely identify transactions
- Transactions form:

  <Node IDs , Sequence Number, Signatures , Nonce>



Design Details: Deploy new Devices

- Public/private key-pair
- Invoke smart contract
- Built-in management



Design Details: Smart Contract

- The smart contract has three sub-protocols:
  1. Abort sub-protocol
  2. Resolve sub-protocol
  3. Validation sub-protocol

Outline

- IoT & Blockchain
- System Design
- Experiment Evaluation & Results

Evaluation Goals

- Memory Footprint
- Cryptographic Performance
- Energy Consumption
- Overall System Performance

Experiment Setup

Hardware

- OpenMotes (TI-CC2538)
  - Cortex M - 32 MHz
  - Cryptographic accelerator
  - Low-power radio

- Desktop server
  - Intel core i5

Software

- Contiki-NG
- ECC
- Hyperledger-Fabric
- Linux

Memory Footprint for IoT Device

- Contiki OS consumes 37% of RAM
- Cryptographic library 4%
- Storing off-line transactions on available RAM

Component	RAM Bytes	RAM Percent	ROM Bytes	ROM Percent
Contiki-NG OS	11,394	37%	40,527	10%
Cryptographic library	1,520	4%	10,775	1%
Application	4,201	13%	7,993	1%
Total footprint	17,115	54%	59,295	12%
Available memory	14,885	46%	452,705	88%

Performance of Cryptographic Operations

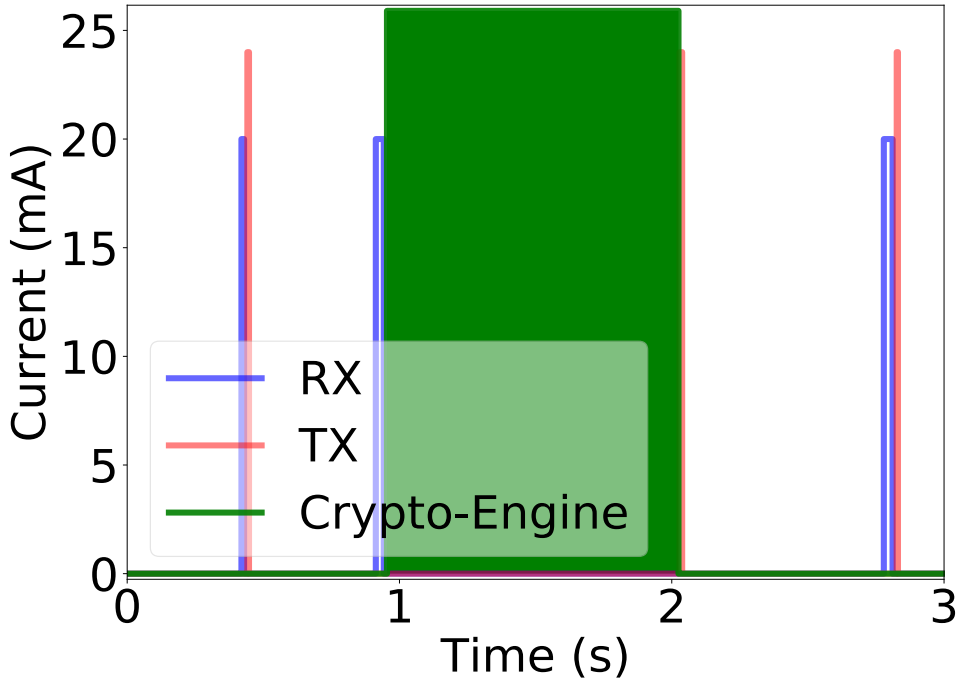
- Sign & verification takes ~1 s
- Hash function negligible
- Complete round less than 3 sec

Function type	Time
ECDSA-Sign on originator	350 ms
ECDSA-Verify on responder	715 ms
ECDSA-Sign on responder	350 ms
ECDSA-Verify on originator	715 ms
SHA256-Hash function	1 ms
Total time	2131 ms

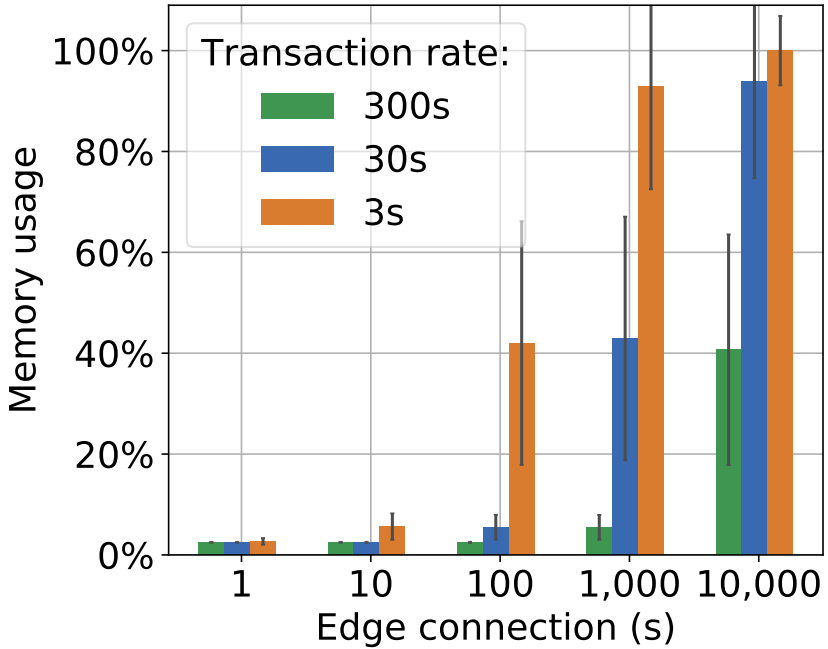
Energy Consumption – IoT Device

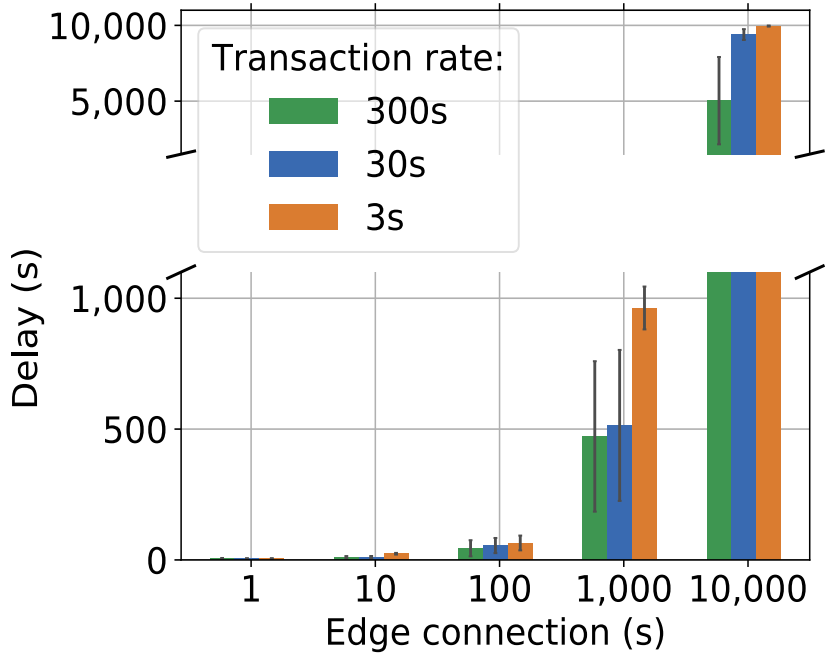
- A round of the protocol takes ~ 3 s
- Two major consumers:
  - Cryptographic operations - 57.9 mJ
  - The radio listening - 35.1 mJ

Energy Consumption – IoT Device



System Performance





Conclusion

- IoTLogBlock – open source architecture
- Connect low-power devices with Blockchain
- Create off-line transactions in ~3 s
- Available on Github

https://github.com/iot-chalmers/IoTLogBlock

CHALMERS

UNIVERSITY OF TECHNOLOGY