Enhancing Trust in IoT Transactions Using Blockchains and Smart Contracts

Christos Profentzas
Chalmers University of Technology, Sweden, chrpro@chalmers.se

***Abstract*—With the rise of the Internet of Things (IoT), billions of smart embedded devices will interact frequently. These interactions will produce billions of transactions. With IoT, users can utilize their phones, home appliances, wearables, or any other wireless embedded device to conduct transactions. For example, a smart car and a parking lot can utilize their sensors to negotiate the fees of a parking spot. The success of IoT applications highly depends on the ability of wireless embedded devices to cope with a large number of transactions. However, these devices face significant constraints in terms of memory, computation, and energy capacity.**

With our work, we target the challenges of accurately recording IoT transactions from resource-constrained devices. We identify two main problems: a) non-repudiation of IoT transactions, and b) inability of IoT transactions to include sensors readings and actuators. The motivation comes from the fact that we need to store transactions from IoT devices that are owned or operated by different stakeholders. We propose two architectures for the above problems. First, we propose IoTLogBlock, an architecture to record off-line transactions of IoT devices. Second, we propose TinyEVM, an architecture to execute off-chain smart contracts on IoT devices with an ability to include sensor readings and actuators as part of IoT transactions.

Keywords: Internet of Things, Distributed Systems, Blockchain, Smart Contracts, Embedded Systems

I. IoTLogBlock

For any distributed system, and especially for the Internet of Things, recording interactions between devices is essential. At first glance, blockchain seems to be suitable for storing these interactions, as they allow multiple parties to share a distributed ledger. However, at a closer look, blockchain requires heavy computations, large memory capacity, and always-on communication to the cloud; these are three properties that are challenging for IoT devices with limited resources.

In the first part of our work, we present **IoTLogBlock** to address these challenges. **IoTLogBlock** connects resource-constrained IoT devices to the blockchain, and it consists of three building blocks jointly enabling recording transactions: a lightweight contract signing protocol, a blockchain network, and a smart contract. The contract signing protocol allows devices to interact locally to perform transactions, even if no communication to the cloud and the blockchain exists at that moment. At a later time, devices forward the stored transactions to the blockchain, where a smart contract ultimately verifies the transactions.

IoTLogBlock focuses on resource-constrained devices like TI CC2538, We evaluate our design on low-power devices and quantify the performance in terms of memory, computation, and energy consumption. Our results show that a constrained device can create and sign a transaction on average of three second. Finally, we expose the devices in different network scenarios with an edge connection ranging from fifteen minutes up to two hours.

II. TinyEVM

In the second part, we introduce **TinyEVM**, a novel system to generate and execute off-chain smart contracts. With **TinyEVM**, smart contracts have access to sensor readings and actuators of IoT devices. **TinyEVM** allows IoT devices to create and perform off-chain payment channels considering the resource-constraints of IoT devices. Through this work, we show the trade-offs of executing off-chain payment channels on resource-constrained devices. We are considering a broad spectrum of applications where the myriads of IoT devices are frequently exchanging payments in two steps. First, they use their sensor data and actuators upon their environment to agree on payment conditions. Second, they enforce these payments by publishing a final state.

We motivate this work using a parking service scenario. Through this scenario, we show the challenges of negotiating a parking place between two parties using a smart contract. In order to make this scenario feasible, we need to ensure two properties. First, we need the utilization of the sensor data to determine the value of the parking place. Second, we need to enable the parties to finish their interactions and complete a transaction in a matter of seconds.

TinyEVM focuses on resource-constrained devices like TI CC2538, and it consists of three components: a) publishing the on-chain smart contract, b) creating off-chain channels, and c) committing on the on-chain. We achieve these steps by separating the logic of on- and off-chain transactions by designing two smart contracts. Moreover, we design, implement, and extend with IoT opcodes an Ethereum Virtual Machine to execute smart contracts on resource-constrained devices.

We investigate the trade-offs to execute smart contracts on recourse-constrained IoT devices. We test our system with 7,000 publicly verified smart contracts, where **TinyEVM** manages to deploy 93% of them. Finally, we evaluate the execution of off-chain smart contracts in terms of run-time performance, energy, and memory requirements. Notably, we find that recourse-constrained devices can deploy a smart

contract in 215 ms on average, and they can complete an off-chain payment in 584 ms on average.