2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation

Poster Abstract: Privacy in Blockchain-Enabled IoT Devices

Arman Pouraghily, Md Nazmul Islam, Sandip Kundu, Tilman Wolf
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA, 01003, USA
Email: {apouraghily, mislam, kundu, wolf}@umass.edu

***Abstract*—Over past two decades, the idea of Internet of Things has been adopted widely as a solution to many societal problems in different areas. These areas include but are not limited to healthcare, transportation, environment, etc. Low cost overhead of Internet connectivity feature has been the main contributing factor in the widespread use of such devices in building different IoT solutions. The original stovepipe architecture of IoT systems limits the possibility of sharing the hardware infrastructure of IoT solutions and therefore is the main barrier against novel solutions. In recent year, however, there have been efforts to come up with solutions for sharing the hardware infrastructure and therefore pave the way for innovative solutions by amortizing the capital cost of setting up the hardware. In this work, we propose an architectural guideline for blockchain enabled IoT devices which facilitates sharing them between multiple blockchain ecosystems and at the same time, ensures the exclusive access to them seamlessly through blockchain smart contracts.**

I. Privacy Challenges

Privacy is a concern whenever common resources are shared. In [1] a new IoT architecture has been proposed which facilitates the horizontal integration of different IoT ecosystems. The assumption in this work is that the resource being shared is controlled by the owner at all times and only supervised access to that resource can be granted to the requester by the owner. The main drawback of such solution is that there is no provisioning in it by which the requester can ensure the privacy of its access to that resource. Although this privacy concern is not a major problem in a variety of IoT applications, such as sharing the data collected from temperature sensors deployed in a farm, it could be a deal breaker when it comes to other types of applications. For example, the data collected by the motion sensors or CCTV cameras in a rental property cannot be shared with the owner of the property while it is occupied by a guest.

One way of addressing this challenge is to introduce a trusted third party [2] who receives the camera feed and shares it only with the designated party which is the renter while the property is rented out and the owner the rest of the time. Although finding such a trusted entity might be tricky in practice, it is not completely infeasible and still can be considered as a viable solution. On the other hand, having this extra entity imposes an extra cost on the sharing agreement.

When it comes to financial mechanisms and trusted third parties, blockchain technology is always a potential solution worth considering. The main feature of blockchain technology is democratizing the process of transaction validation and commitment. By doing so, blockchain eliminates the need for a trusted centralized ledger. Another powerful feature of blockchain is the concept of smart contracts which is supported by most of the blockchain based networks. Smart contracts not only allow the parties to issue transactions transferring assets between them but also enables them to regulate the financial transactions between them according to a non-repudiable agreement between them without the need for a trusted intermediary entity.

In this work, we propose an architectural guideline for blockchain enable IoT devices. This architectural guideline enables IoT devices to directly interact with the blockchain smart contracts through which they can change the entity they are serving and modify their security parameters. In our work, we use Ethereum as it offers a powerful protocol for building applications in a decentralized manner. Those applications include, but are not limited to, financial applications, which are described in more detail in [3]. Ethereum's cryptocurrency, Ether, is also one of the most popular cryptocurrencies in the market, which means it is more likely for our system to be adopted as a solution in practice.

II. Design Example

As an example let us consider the CCTV camera scenario inside a rental property. The camera is controlled by a smart contract on the blockchain. The contract is created by the manufacturer of the camera once it is produced. Assuming asymmetric encryption mechanism used by the camera for securely transferring the feed, the public key of the camera is hard-coded in the smart contract while the private key is securely stored on the camera's hardware. When a buyer purchases the camera, he/she provides the manufacturer with their blockchain account information as well as the Internet address to which they want the camera feed to be sent. Receiving this information, before publishing the contract on the blockchain, the manufacturer hard-codes this information in the smart contract as *owner information*. In addition to the public key of the camera and the owner information, the smart contract also has a set of fields dedicated to the potential renter's information. Once renting the device, a renter can fill those fields by sending an *initiate* message to the contract.

The renter information fields include the public key of the renter to be used for encrypting the camera feed and the Internet address to which the renter wants the feed to be sent.

0-7695-6387-2/18/$31.00 ©2018 IEEE
DOI 10.1109/IoTDI.2018.00045

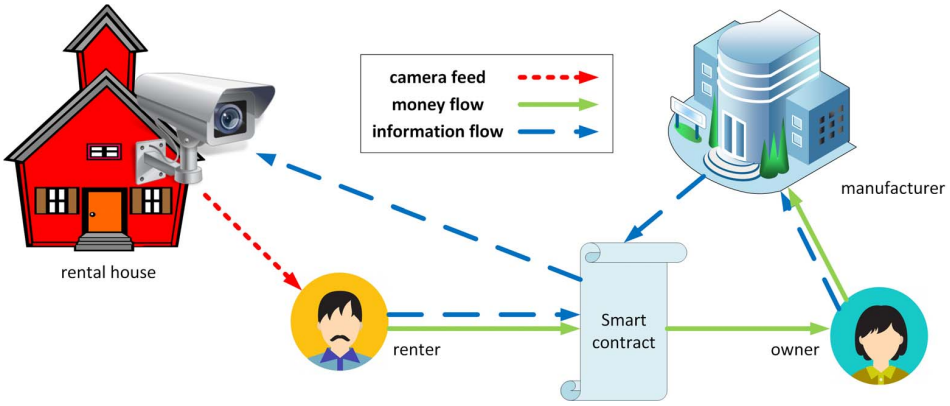

Fig. 1. Money and information flow in the rental scenario.

owner's IP	owner's public key
camera's IP	camera's public key
renter's IP	renter's public key
initiation date	contract term
balance	rental status (rented/unrented)

Fig. 2. Data structure of the smart contract.

In order to prevent the contract from being abused or hijacked by adversary parties, the *initiate* message needs to be signed by the owner as well. One last thing that needs to be specified in the contract, is the duration for which the feed is being redirected to the guest. This duration is also set by the *initiate* message. Fig. 1 depicts the money flow and information flow between the involved entities.

Having the contract duration set in the contract, the renter can make sure that the owner will not be able to terminate the contract and redirect the data stream back to himself/herself while the property is still legally occupied by the renter. On the other hand, the owner should also be able to ensure that he/she will be compensated for renting out the camera feed to the renter. In order to ensure the compensation, the contract also carries a *balance* which will be transferred to the owner's account upon termination of the contract. This balance is the value associated with the *initiate* message sent by the renter and will be withdrawn from renter's account received by the contract. It will be locked up in the contract until the end of the contract term. At the end of the contract term, the balance will be transferred to the owner's account which is hard-coded in the contract. The *balance* can be zero in case the camera feed is being transferred to the renter as a part of a broader contract (e.g. the whole house being rented out). The data structure used in the smart contract is shown in Fig. 2.

Finally, as the smart contracts are reactive entities, in order to terminate them, one should trigger them. This is done by the owner once the rental period is passed. The *trigger* message is only accepted and processed by the contract if it is coming from the owner and if the contract period is passed. Upon receiving the *trigger* message, if the rental period is over, the contract reverts its state to *unrented* which means that the camera is no longer going to use the renter's information to send its feed and it will redirect the stream to the owner's address and will use the owner's public key for encrypting its data.

III. Summary and Future Plans

Sharing IoT devices introduces new opportunities for innovation but at the same time, it makes it challenging to ensure private access to the resources being shared. In order to avoid the cost overhead of a trusted third party supervising the private access to the resource, we proposed the alternative solution of using blockchain technology. In our proposed method which is based on Ethereum blockchain network, the shared resources are directly connected to the blockchain and are controlled by a smart contract through which they receive and update their security parameters as well as the serving user's information. In this work, we presented the design guidelines for such IoT devices and the smart contracts controlling them. In the next step, we are planning to implement the proposed architecture on an embedded system such as Raspberry Pi and the smart contract on Ethereum testnet to explore the processing overhead of our proposed solution on a typical low cost IoT device.

References

[1] T. Wolf, M. Zink, and A. Nagurney, "The cyber-physical marketplace: A framework for large-scale horizontal integration in distributed cyber-physical systems," in *Proc. of the Third International Workshop on Cyber-Physical Networking Systems (CPNS) held in conjunction with the IEEE 33rd International Conference on Distributed Computing Systems (ICDCS)*, Philadelphia, PA, Jul. 2013, pp. 296–302.

[2] N. Jefferies, C. Mitchell, and M. Walker, "A proposed architecture for trusted third party services," in *Cryptography: Policy and Algorithms*. Springer, 1996, pp. 98–104.

[3] V. Buterin, "A next-generation smart contract and decentralized application platform," Dec. 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper