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

Preserving IoT Privacy in Sharing Economy via Smart Contract

Md Nazmul Islam, Sandip Kundu
University of Massachusetts Amherst, MA, USA
{mislam, kundu}@ecs.umass.edu

***Abstract*—The phenomenal growth of Internet-services has created a vibrant new domain for sharing economy. Millions of users around the world share personal services and possessions with others – often complete strangers. Such sharing schemes also increase the risk of violation of one's informational and physical privacy. Strangers often have to trust each other with their privacy (e.g. a surveillance camera in an Airbnb room). However, very little research has been devoted to investigate privacy in sharing economy. In this paper, we explore the privacy concerns associated with contractual renting or leasing of IoT devices-enabled home. We propose a methodology to eliminate privacy threats from IoT-enabled telematics devices in a smart home via blockchain-based smart contract. For the purpose of illustration, we focus on how we can circumvent the privacy threat from indoor surveillance IP cameras in a smart home-sharing economy.**

***Index Terms*—Blockchain, Smart Contract, Sharing economy, Privacy, Internet-of-Things.**

I. Introduction

'Sharing economy' platforms such as Airbnb have recently flourished in the tourism industry. However, relying on a centralized third party sharing platform inevitably leads to single point of weakness, higher fees, lack of trust and governance issues for both users and service providers [1]. Moreover, sharing any IoT-devices enabled smart house poses a serious threat to user's privacy. Airbnb hosts prefer to know what's going on in their rentals. Because of this, hosts may opt to have surveillance cameras in key places. This allows Airbnb hosts to spy on guests which is a serious infringement upon the guests' expectation of privacy. Similarly, by accessing the smart door lock, an intrusive homeowner can compromise the security system. By accessing the stored credentials on connected devices, hosts can take control of the IoT devices' sensors and can even disable an apartment's control of HVAC systems [2].

In this paper we propose blockchain technology-based smart contract to eliminate *(i)* distrust by decentralizing home-sharing economy and *(ii)* privacy threats from IoT-enabled telematics devices in a sharing house.

II. Motivation

Blockchain is a decentralized, immutable ledger that keeps records of digital transactions [4]. This immutable blockchain record can establish trust and eliminate the need for middle-men selling users' personal details. Everything from finding and booking a property to payments, reviews & post departure sequences can be handled through smart contracts on blockchain. Smart contact translates the existing contractual clauses into embedded hardware and software in such a way that it can self-verify that conditions have been met to execute the contract [5]. Smart contracts contain code functions and can interact with other contracts, make decisions, store data, and send tokens/money to others.

Smart contracts can also facilitate efficient IoT devices by automating their operations and decision making. This can be achieved by allowing IoT devices to interact with smart contracts and make decisions defined by the fixed contract logic. For example, in order to safeguard surveillance data, modern IP cameras are equipped with an on-board security chip, Trusted Platform Module (TPM) [3]. Using a symmetric cryptographic key, the TPM encrypts the data stream (Fig. 1). In this paper, we leverage the TPM to change its encryption key whenever there is a tenancy change recorded in smart contract. This key can only be computed using the device and the tenant's private key so that no one else can access the surveillance data other than the current tenant.

III. Proposed Methodology

Fig. 2 presents our proposed protocol to decentralize home sharing economy and preserve users' privacy. The protocol consists of the following steps:

A. Implementing a Smart Contract

In the first step of our proposed protocol, the manufacturer of an IoT device (an IP camera, for example, in our case) creates a smart contract (*possessionContract* in Fig. 2). This contract offers functions for managing the possession transfer and polling the possession of the device. The manufacturer then deploys the contract in blockchain and embeds the

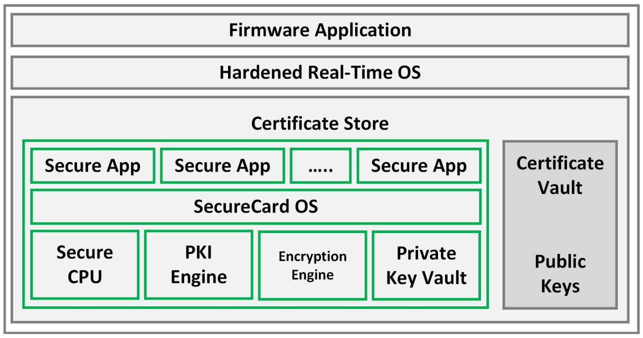

Fig. 1. Block diagram of a Trusted Platform Module (green block) embedded into the camera's software architecture [3].

0-7695-6387-2/18/$31.00 ©2018 IEEE
DOI 10.1109/IoTDI.2018.00047

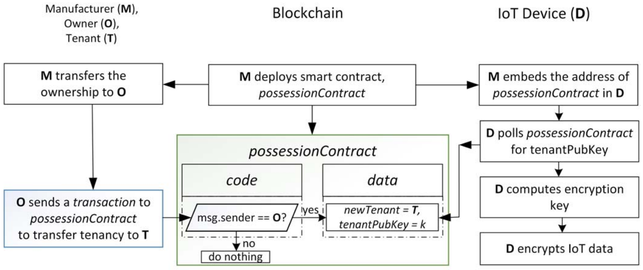
Manufacturer (M),
Owner (O),
Tenant (T)
Blockchain
IoT Device (D)
M transfers the ownership to O
M deploys smart contract, possessionContract
M embeds the address of possessionContract in D
D polls possessionContract for tenantPubKey
possessionContract
code
data
O sends a transaction to possessionContract to transfer tenancy to T
msg.sender == O?
yes
newTenant = T, tenantPubKey = k
no
do nothing
D computes encryption key
D encrypts IoT data

Fig. 2. Detailed diagram of showing the role of all entities in the proposed privacy protection protocol.

address of the contract in the device. The blockchain can be a public blockchain (e.g Ethereum), maintained by its community, including its developers, users, service providers (exchanges), miners and others. Finally, the manufacturer transfers the ownership to the first owner of the device.

The contract consists of *code* (its functions) and *data* (its state). The functions are *transferTenancy*, *pollTenancy* and the state variables are owner, tenant, tenant public key, device public key etc. Once the smart contract is appended to the blockchain, an IoT device executes whatever the contract makes it to execute. As the contract is immutable once it's uploaded onto the blockchain, no one can tamper with the code.

B. *Transferring Tenancy to a Tenant*

To transfer the tenancy to a tenant, the owner sends a transaction to *possessionContract*. This transaction defines all the necessary information related to tenancy transfer, such as, new tenant information, tenancy period, cost etc. The transaction includes tenant's public key which will be used by the IoT device for computing the data encryption key. If the sender of the transaction is the current owner of the device, the smart contract updates its new tenant and the new tenant public key.

C. *Establishing a Shared Encryption Key*

The IoT device polls the address of the embedded smart contract intermittently (e.g. once in everyday). For video stream encryption purpose, a symmetric key is established using the Diffie-Hellman protocol. On one hand, the device calculates the symmetric key using its private key stored secretly inside the device and the tenant public key from the smart contract. On the other hand, the tenant calculates the symmetric key using his own private key and the device public key. The computation happens based on some pre-established large prime number, p and generator, g which is a primitive root modulo p.

D. *Encrypting IoT Data with the Shared Encryption Key*

The encryption engine of the TPM changes the encryption key to newly computed symmetric key. Then it encrypts all the video data stream or other payload with encryption key (Fig. 3). The tenant can also decrypt the video data with the shared key. On the other hand, the owner can no longer decrypt the surveillance data as the key has been changed.

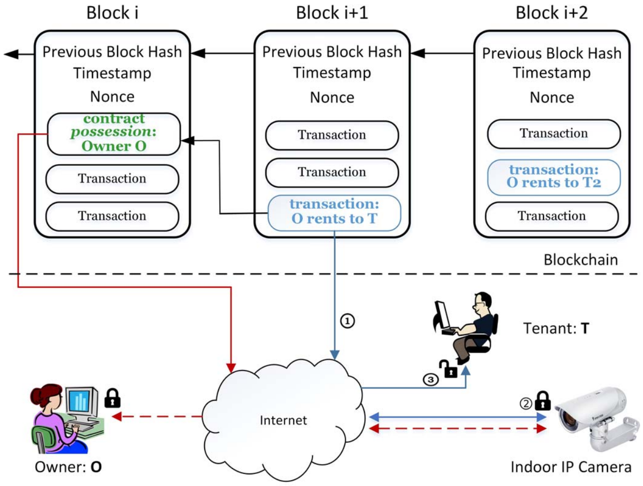

Fig. 3. Detailed diagram of the privacy protection protocol: (1) the smart contract notifies the IP camera about the tenancy change, (2) IP camera computes the symmetric key from tenant's public key and encrypts video data, (3) tenant calculates the symmetric key and can decrypt the video data. The dotted lines indicate inaccessible data.

E. *Change of Encryption Key after Tenancy Period*

The *transferTenancy* function in *possessionContract* defines that after the tenancy period, the tenant will be the original owner. So, the encryption key will be changed automatically according to original owner's public key.

IV. Hardware Collateral for Smart Contract

For implementing the proposed smart contract, the IP camera's system-on-chip (SoC) needs to be equipped with an encryption engine, such as AES, DES, 3DES etc. Most of the modern IP cameras are equipped with such security engines for safeguarding the data [3]. Similar privacy protection methodology can be applied to any other IoT devices present in a home. Alternatively, a Smart Home Hub can perform the privacy protection for all the IoT devices connected to it.

V. Conclusion

In our proposed protocol, the smart contract enables the decentralization of home-sharing economy. At the same time, by facilitating the change of encryption key via smart contract, it can preserve the IoT privacy. Our future plan includes deploying the proposed smart contract in Ethereum blockchain and building a Distributed Application (DApp) for home-sharing economy.

References

[1] Abraham. Decentralizing airbnb. [Online]. Available: https://www.smarthosts.org/posts/Zr4w8SEK5BH42CPZF/airbnb-blockchain-loyalty-travel

[2] S. Meza, "Airbnb hosts are recording their guests with hidden cameras," *URL: http://www.newsweek.com/airbnb-hidden-cameras-recording-guests-739709*, December, 2017.

[3] BOSCH, "Data security - how bosch secures the camera," *URL:http://resource.boschsecurity.com/documents/WP_TPM_WhitePaper_enUS_9007223261094667.pdf*.

[4] R. Wattenhofer, *The science of the blockchain*. CreateSpace Independent Publishing Platform, 2016.

[5] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.