Blockchain-based Ownership Management for Medical IoT (MIoT) Devices

M. Alblooshi, K. Salah, Y. Alhammadi
Department of Electrical and Computer Engineering
Khalifa University, UAE
{mansoor.albloushi, khaled.salah, yousof.alhammadi}@ku.ac.ae

***Abstract*—Recently, blockchain has been foreseen by industry and research community as a transformational and disruptive technology that is poised to play a major role in transforming the way we transact, trade, bank, track and register assets, and do business. Blockchain can potentially be used to provide trusted authenticity, origin, and ownership for IoT devices, in a way that is secure and decentralized without the involvement of a Trusted Third Parties (TTP) or centralized authorities and services. A TTP can be a subject to a single point of failure, and it can be disrupted, compromised or hacked. A TTP can potentially misbehave and become corrupt in the future, even if it is trustworthy now. This paper shows how Ethereum blockchain (with the powerful feature of programmability through smart contracts) can provide a trusted ownership management for medical IoT (MIoT) devices. Tracking the true ownership of MIoT devices is of a paramount importance as using counterfeited MIoT devices can be life-threatening to patients. The paper presents a general framework and solution to manage and trace back the true origin of ownership for an MIoT; whereby an MIoT device can be owned by multiple parties during its life cycle. The paper presents a detailed overview of our proposed system architecture, design, entity relationship, and interactions among all involved parties. The source code of the Ethereum smart contracts is made publicly available. Key aspects related to the algorithms, interactions, implementation and testing are discussed in the paper. Furthermore, we provide security analysis of our proposed solution in terms of satisfying the general security goals and also resiliency against popular attacks as those of DDoS, eavesdropping and replay attacks.**

***Keywords*—IoT, IoT Security, Blockchain, Authentication, Trust, IoT Device Management**

I. INTRODUCTION

During the past few years, Internet of Things (IoT) had experienced wide-spread development and had gained fast-growing popularity worldwide [1]. Nowadays, IoT is widely used in smart cars, smart homes, wearables, smart cities and healthcare. The market of IoT devices is growing quickly, covering a wide range of areas, such as traffic surveillance, wildlife monitoring, and gas leakage detection.

Today's massive adoption and deployment of IoT in many fields and industries introduce many key challenges which include security and privacy issues and management of IoT devices, as well as the data being generated from these IoT devices [2]. Unauthorized surveillance and uncontrolled data use are some of the key privacy challenges facing IoT devices. Management of IoT devices and their ownership, in a way that is authentic and trusted, is a key challenge these days, especially if the IoT device is sold and re-sold by multiple parties. There is a need to know and trace back the origin of the IoT device, and determine if it is original or counterfeited, specifically in healthcare industry. This is complicated by having multiple stakeholders, individuals, organizations, and third parties, being involved in IoT device ownership and management.

In the area of healthcare, authenticity and originality of medical IoT (MIoT) devices being used on patients are of paramount importance. MIoT devices can come today in many variety and forms of wearables, implantables, or injectables. A counterfeited MIoT device can be life threatening to patients. Therefore, it is important to have the ability to trace back the history of MIoT device in a credible and trusted manner with no centralized management or the use of a Trusted Third Party (TTP). Third parties and organizations take role in providing enhanced services for the individual and for the society at large [3]. Multiple solutions have been proposed in the literature to overcome these IoT challenges including privacy and ownership. Most of these solutions propose involvement of trusted third party acting as a centralized management. TTP can be a single point of failure, hacking, compromise, and potential of corruption. Therefore, a decentralized ownership management of MIoT devices becomes key.

In this paper, we show how the newly emerging technology of blockchain can play a major role in providing a decentralized trust and management of MIoT devices. The main contributions of this paper can be summarized as follows:

- We propose blockchain-approach using Ethereum smart contracts to manage the ownership of IoT devices in a decentralize manner with no trusted third party.
- We present a detailed overview of our proposed system architecture, design, entity relationship, and interactions between all parties interacting with the smart contracts which are uploaded on the Ethereum blockchain platform.
- We provide the full source code of the Ethereum smart contracts and discuss key aspects related to its algorithms, interactions, and logical flow. The full code is made avalable at Github[1]. In addition, we show how we implemented and tested the smart contract, and ascertain the correctness of the overall system functionality and behavior.

[1]https://github.com/MansoorUAE/IoTOwnership/blob/master/Manufacturer.sol

978-1-5386-6673-9/18/$31.00 ©2018 IEEE

- We analyze and discuss the security of our proposed solution in terms of satisfying the general security goals and also resiliency against popular attacks as those of DDoS, eavesdropping and replay attacks.

The rest of the paper is organized as follows. Section II gives a brief overview and background on blockchain, and Ethereum Smart Contracts. Section III highlights related work. Section IV details out the proposed solution and its implementation. Section V presents security analysis of our blockchain-based framework and solution. Finally, Section VI presents our conclusions and future work.

II. BACKGROUND

Our proposed solution is based on using blockchain and Ethereum smart contracts. This section provides a brief introduction on blockchain and Ethereum smart contracts.

A. Blockchain

Blockchain is a distributed database or ledger that can store records and data which can be globally accessed and shared among a network of independent parties [4] in a decentralized and trusted manner. Creation of immutable records is a key feature of blockchain. It gives the ability for any participant to manage the ledger securely without the need for a trusted intermediary or centralized trusted third party [4]. Removal of centralized authority while maintaining data integrity was the main motivation of blockchain [4]. Blockchain consists of a number of chained blocks, where each block contains a list of transactions recorded into a ledger, and they are chained by a hash, linking the new block to the previous one [4–6]. Any recorded data on blockchain is tamper resistant. Any new preformed record (block) that need to be added to an existing chain must be through a consensus mechanism, where special nodes (called miners) have to validate such transactions [4, 7]. The miners on blockchain network maintain their own replicated, shared, and synchronized database digital information [7]. All transactions on blockchain network are available to all of its users. It provides the ability of automating transactions, real time settlement with respect of providing protection against fraud [7]. Each user on the blockchain has the ability to read and write from the database. Each stored information and state of the database are validated, and then confirmed though cryptographic consensus mechanism. The added transaction is tamper resistant and it cannot be updated or deleted [7].

B. Ethereum Smart Contracts

The first type of blockchain network was Bitcoin. The second type is called Ethereum blockchain which has the same features and traits of the Bitcoin blocchain, but can be programmable using the notion of smart contracts. Smart contracts concept is the key feature of Ethereum blockchain. It provides users ability to design their own business code and programs to be executed by all miners. The execution outcome is validated by all mining nodes. This concept has been developed by Vitalik Buterin the founder of Ethereum in 2014 [8].

A smart contract is basically a program that validates a transaction along with immediate effect of the contract, without involvement of trusted third party [8]. Any deployed smart contract on a blockchain has a unique Ethereum public address. Users on the network access the smart contract using this address. In a smart contract, there are two types of functions: getters and setters. A getter function is used for retrieving information and state variables; whereas, a setter function is used for setting values to variables. Using qualifier called ”only” can restrict access to these functions. Smart contract provides user an ability of logging any changes on any concerned value by using events. Whenever the user executes a function, he/she would have to pay an amount of gas. The amount of gas depends on computational steps. The concept of smart contracts can be used in managing ownership of the IoT device among two parties as a decentralized manner.

III. RELATED WORK

This section briefly surveys and summarizes existing solutions in the literature addressing the IoT ownership problem. The solutions can be categorized as trusted third party ownership management, item level access control framework, and authentication and access control for the entire IoT device life cycle. First, the proposed ownership management system by [9] is based on a trusted third party (TTP) model. It covers both the ownership establishment and transfer modes as shown in Figures 1 and 2, as presented in [9]. In the figures, K denotes a secrete key and ID(p), ID(o), and ID(n) denote the identity of the IoT device, current owner, and new owner, respectively.

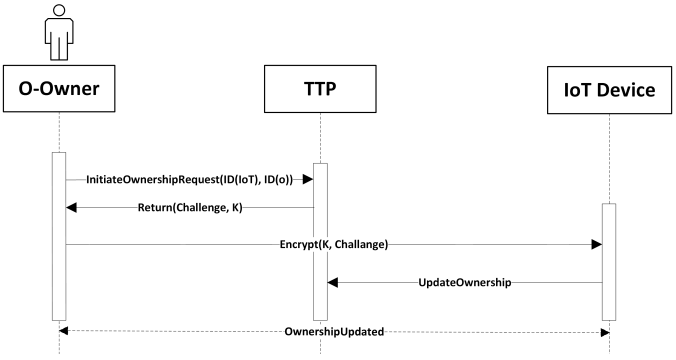

Fig. 1: Ownership creation model as proposed in [9]

According to [9], owner registration and the establishment of the security credentials with the particular IoT device is required. Moreover, TTP is responsible for generating ownership challenges as shown in Figure 1 and 2. Furthermore, the IoT device has a pre-installed secret shared with TTP and has a one-way function h() to generate the secret key for ownership. An ownership creation model is used when the IoT device ownership first is registered. The transfer procedure requires both the current owner and the new owner to be registered with the TTP as shown in Figures 1 and 2.

The authors in [10] argued that, in case of ownership

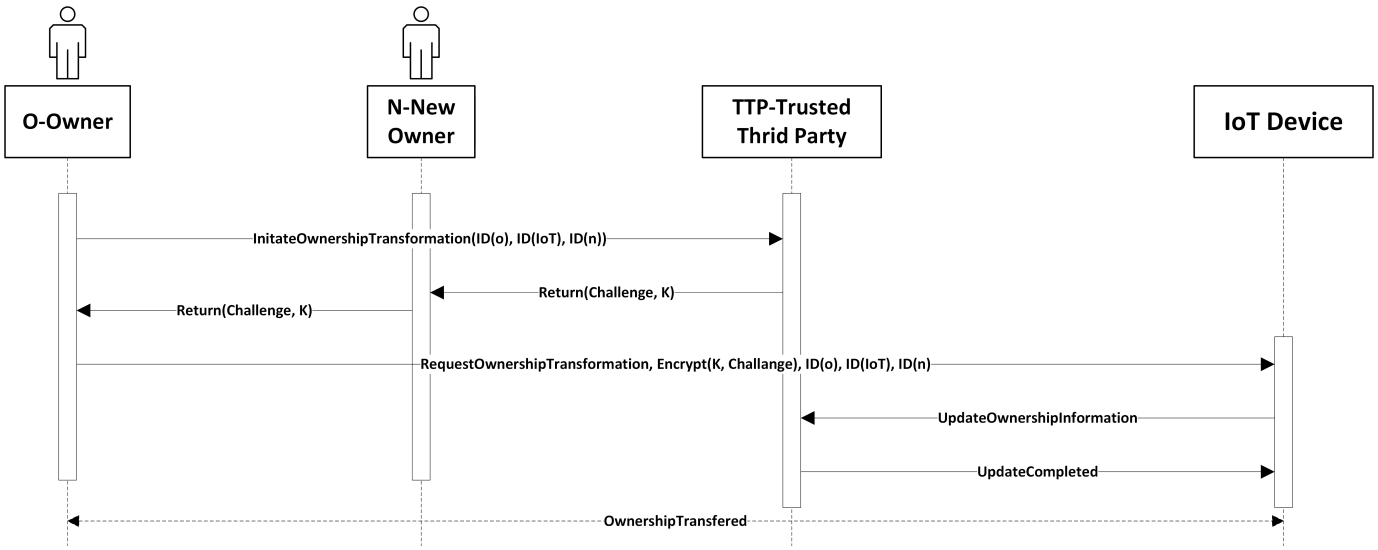

Fig. 2: Ownership transfer model as proposed in [9]

transfer for an IoT device, a trust should be established between both owners: current and new owner. This includes permissions and access control. A framework based on item level access control through mutual trust has been presented in [10] for IoT devices. In short, the newly manufactured device is given a key by a trusted third party. This key will be used to provide the current owner device permission control. For transfer of ownership, a token will be created by current owner of IoT device. This token is combined with device RFID identity to grant the new owner the permission control along with IoT device ownership.

The authors in [11] presented an IoT device life cycle, which is essentially divided into five stages, pre-deployment, ordering, deployment, functioning, and retirement. The cryptographic keys of the server are loaded into IoT device via the manufacturer at the pre-deployment stage. In the ordering stage, the new owner of the IoT device receives an IoT access PIN. The development stage establishes a trust relationship between the device and a TTP which handles the key management and the access control.

All of the previous solutions have clear limitations stemming from the use of centralized management entity or TTP for the ownership management and the key management. The solution has clear limitations. The solution depends on TTP (centralized management) which is subject to corruption, single point of failure, and being comprised or hacked. In addition, according to [9], the solution has performance issues. Also, the solution does not keep ownership history that is open and trusted.

IV. BLOCKCHAIN-BASED SYSTEM FOR DEVICE OWNERSHIP MANAGEMENT

This section gives a system overview and architectural design for solving the ownership problem for medical IoT devices (MIoTs) in a decentralized, trusted manner by using Ethereum smart contracts. Figure 3 illustrates the main participants of the system which include the medical IoT device, the potential owners of the device (patient), manufacturer, physicians, clinic and hospital. The system consists

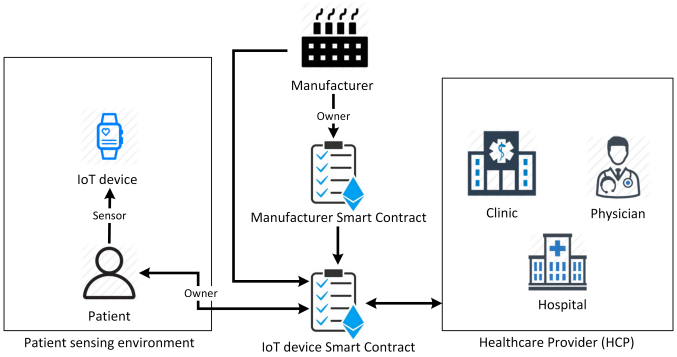

Fig. 3: System overview

of two smart contracts code, manufacturer and IoT device smart contract, which get deployed on Ethereum blockchain network. Manufacturer smart contract is used by manufacturer in order to deploy IoT device smart contract whenever IoT device is manufactured.The other smart contract gives the IoT device owner the ability to set rules and conditions for access and modifying the records pertaining to the MIoT device ownership records stored on the Ethereum blockchain network. The smart contract provides the owner with the mechanism for transferring the MIoT ownership without involving a TTP or centralized entity. In a way, any participant can interact with the contract to get, set, or modify its variables based on the rules that been set by the owner without involving a central authority, as the execution of the smart contract is carried out and the execution outcome is validated and agreed on by the thousands of miner nodes.

The following subsections discuss in more detail design aspects, smart contract logic and code, implementation and

testing of the overall system functionality.

A. Design Aspects

The proposed system consists primarily of five key components which all have Ethereum Addresses (EA): Patient, manufacturer, IoT device smart contract which gets created by the manufacturer, and the healthcare provider.

Patient. The owner of the IoT Device who can set the rules and conditions on the MIoT device smart contract for device management.

Manufacturer. The MIoT device manufacturer creates the original smart contract once the device is manufactured.

Smart Contract (SC). The smart contract holds all the code and logic for the rules to manage device ownership. The smart contract actually contains the getter, setter, and modifier functions to be accessed by participants. Moreover, it has the ability to log events. In our proposed solution, we use two smart contracts. One SC is used by the manufacturer for every MIoT device, and one SC is used by the owner.

Healthcare provider (HCP). HCP may include users who are interested in knowing, owning, or transferring the ownership of the MIoT device.

Figure 4 illustrates the entity relationship of the proposed system among the different participants in the context of Ethereum environment. As shown, each entity can have multiple relationship with the smart contract, and every IoT device is associated with a single owner and the owner can have ownership of multiple IoT Devices. Moreover, in order to deploy a successful system, each entity should be registered at Ethereum network with an Ethereum public address which is derived from the key pairs.

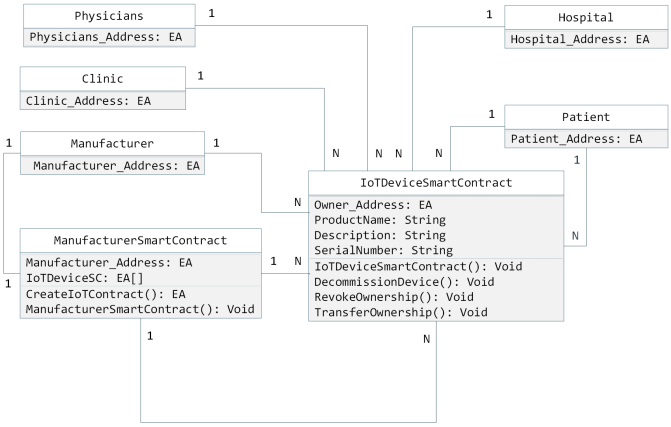

Fig. 4: Entity relationship of the proposed system

B. Logic Flow and Smart Contracts

This subsection illustrate briefly the logical flow of the developed smart contract. The manufacturer of the IoT device creates a contract through the constructor of the IoTCreation and specifies the owners public address along with the default IoT device details. Then the contract logs an event. Figure 5 shows the code for the IoTCreation constructor.

```
function IotCreation(address _owner, string _productName,
string _description, string _specification,
string _serialNumber){
// store IoT device owner record
owner = _owner;
productName = _productName;
description = _description;
specification.push(_specification);
serialNumber = _serialNumber;
// keep record of all IoT device owners
addNewOwner ('Device has been created', owner);
}
```

Fig. 5: IoT Smart Contract constructor

Figure 8 shows the sequence diagram of ownership transfer for the MIoT device, in which the new owner first needs to send his/her public address to the manufacturer. Once received, the manufacturer initiate the ownership transformation by calling TransferOwnership function, which records the new owner EA address and gives the new owner the full control of IoT device smart contract. Then, the new owner(Patient) can have control for modifying the records of the SC. The Solidity code for the ownership transfer is shown in Figure 6.

```
//function that allows owner to transfer device ownership
function ownershipTransfer(address newOwner) ifOwner {
owner = newOwner;
}
```

Fig. 6: Ownership transfer function code

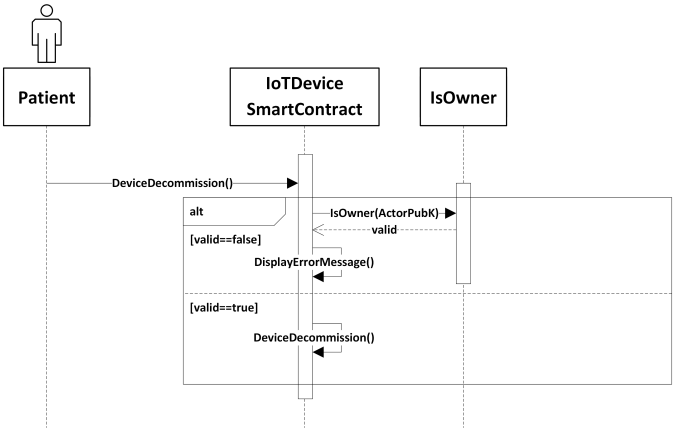

Fig. 7: Sequence diagram for decommissioning an MIoT device

If the Patient (or any owner) wishes to decommission the MIoT device, the Patient calls deviceDecommissioning function of the smart contract. The contract will verify the true ownership before proceeding with decommissioning using modifier only restriction as shown in Figure 7. The code for the deviceDecommissioning function is shown in Figure 9.

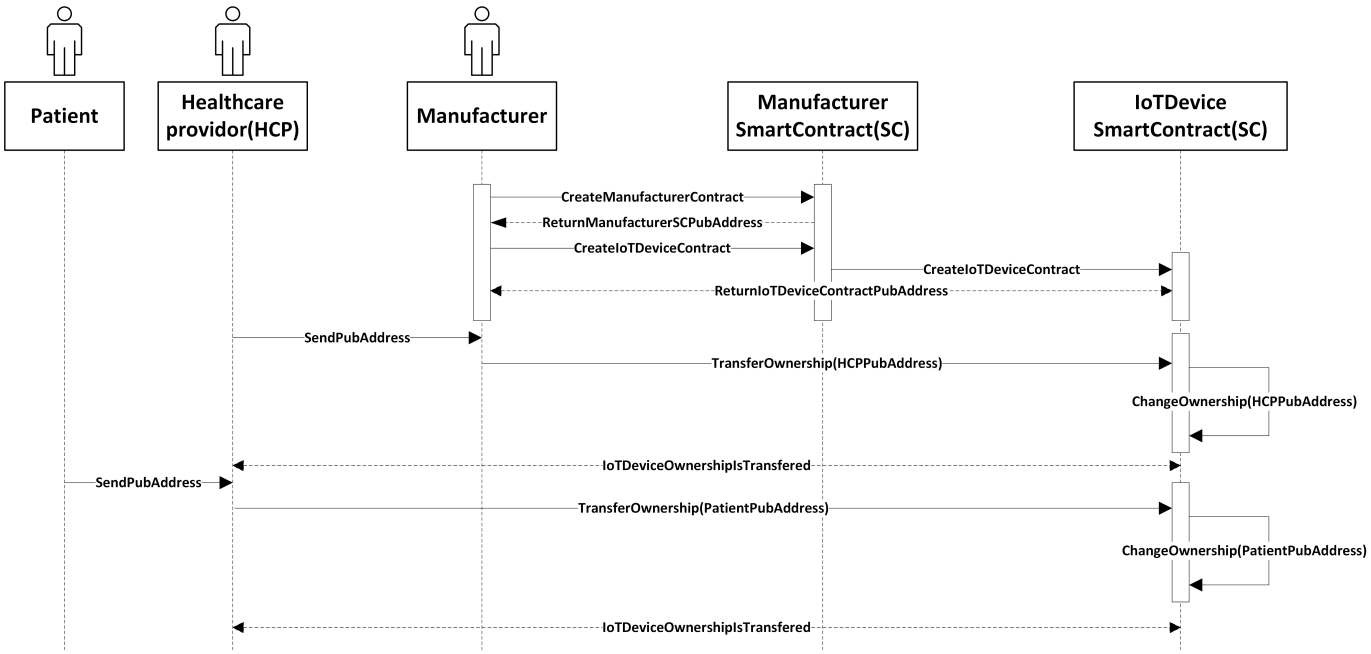

Fig. 8: Ownership management sequence diagram

```
//function controlled by the owner to decommission IoT device
function deviceDecommissioning() ifOwner{ //no revert back
    owner = 0;
    // keep record of IoT device decommissioning action
    addNewOwner ('Device has been decommissioned', owner);
}
```

Fig. 9: Function code for decommissioning MIoT device

status	0x1 Transaction mined and execution succeed
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to	IotCreation.ownershipTransfer(address) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
gas	3000000 gas
transaction cost	28464 gas
execution cost	5784 gas
input	0xd3a417a2000000000000000000000000014723a09acff6d2a60dcdf7aa4aff308fddc160c
decoded input	{ "address newOwner": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C" }

Fig. 10: Remix logs for successful ownership transfer

status	0x0 Transaction mined but execution failed
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to	IotCreation.ownershipTransfer(address) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
gas	3000000 gas
transaction cost	23191 gas
execution cost	511 gas
input	0xd3a417a2000000000000000000000000014723a09acff6d2a60dcdf7aa4aff308fddc160c
decoded input	{ "address newOwner": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C" }

Fig. 11: Remix logs for failed ownership transformation

C. Implementation and Testing

The smart contract has been developed, tested and implemented using the popular Ethereum Remix IDE[2]. This section covers key aspects related to testing smart contract functionality among different system participants. The testing includes all the aspects of the system such as ownership management and IoT device decommissioning. Ethereum Remix is a powerful IDE and tool that provides the ability of testing the smart contract on test network, prior to deploying it to the real network. The tool offers multiple Ethereum wallets, which allows for testing different scenarios with multiple parties. In order to verify and ensure the expected functionality and execution from all the functions (setters, getters, and restrict modifiers), various scenarios with multiple parties have been tested. In order to test the first scenario, an ownership transfer using Transfer ownership function has been initiated as shown in Figure 10. The ownership information of MIoT device has been successfully changed as shown in the logs. If the previous owner tries to initiate ownership transfer for the same device it will fail since he/she is no longer the owner of the device as as shown in Figure 11.

[2]https://remix.ethereum.org

V. SECURITY ANALYSIS

This section discusses and analyzes briefly the overall system security of our proposed solution. We address the fundamental objectives of security, and how our proposed system is resilient against known attacks and security vulnerabilities. Except for **confidentiality**, almost all security objectives of integrity, availability, and accountability are satisfied. Our solution uses Ethereum which is public or permissionless blockchain network in which all transactions are sent in the

clear so that the public minder nodes can verify and validate their content. Nevertheless, confidentiality can be achieved using private or permissioned blockchain Ethereum network, if confidentiality of records and transactions are required.

Non-repudiation is one of main security requirement for IoT device ownership management. This will verify the true message origination of participants on the blockchain network. This is achieved by providing any participant on blockchain network a 20-byte Ethereum Addresses (EA) along with asymmetric key pairs which will be used for signing messages and events. By design and as a feature of blockcahin transactions, every transaction or event issued to or from the smart contract among participants is cryptographically signed and verified, and therefore guarding against any attempt for **eavesdropping or Man-In-The-Middle attacks**. Moreover, every single transaction is embedded with nonce value and timestamps. This will provide the system with a protection mechanism against **replay attacks**.

Ownership information as well as interactions and execution outcomes are all stored, verified and validated by the tens of thousands of the public Ethereum miner nodes in a distributed and decentralized fashion. The outcome of the execution of the smart contract logic is agreed-on by consensus, and thus providing tamper-proof data storage with **high integrity, trust, resiliency, and availability** of records and data across the miner nodes. This is a powerful feature of blockchain platforms which is the resiliency against **Denial-of-Service(DoS)** attacks, as it would be practically impossible to DDoS attack all the globally distributed miner nodes, or alter their contents knowing such contents are hashed, and this contents are all duplicated in all miner nodes. Moreover, any attempt to tamper with or invalidate one block of data will invalidate all the blocks in the chain. By design, the records of IoT devices ownership and details of the IoT devices are all stored on Ethereum ledger in a decentralized manner. Such design is not subject to single point of failure, hacking, or compromise.

Lastly, we have performed **Smart contract vulnerability assessment** using Oyente tool[3] to ensure that our smart contract code presented in this paper is free of bugs and software vulnerabilities which can be exploited by attackers. For this, we used the popular Oyente which is an open source tool developed for scanning smart contracts. Oyente has the ability to detect and report if a smart contract code has known security vulnerabilities. As shown in Figure 12, no vulnerabilities have been detected in our smart contract.

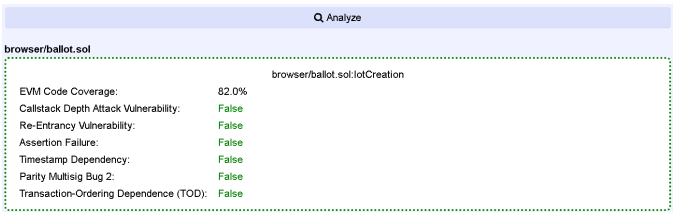

Fig. 12: Vulnerability assessment result

[3]https://oyente.melon.fund

VI. CONCLUSION

In this paper, we have presented a general framework and solution for ownership management and of MIoT devices to solve the counterfeiting problem. Our solution is based on using Ethereum blockchain smart contracts which offer a tamper-proof, trusted, secure, and credible tractability and tracking of origin history and true ownership in a way that is decentralized and resilient to common cybersecurity attacks. The paper provided security analysis of our approach, and discussed that except for the security goal of confidentiality, all security goals are satisfied. We also discussed and demonstrated that the smart contract code is free of bugs and security vulnerabilities. As a future work, we are in the process of implementing our solution on the real Ethereum network, and also developing individualized DApps (Distributed Applications) that will be utilized by the different system actors involved in managing the MIoT devices. Also we are investigating how blockchain can solve issues related to user authentication and access control for MIoT devices, without the involvement of trusted intermediaries or centralized entities.

REFERENCES

[1] Y. Zhang and J. Wen, "The IoT Electric Business Model: Using Blockchain Technology for the Internet of Things," *Peer-to-Peer Networking and Applications*, vol. 10, no. 4, pp. 983–994, 2017.

[2] X. Carron, R. Bosua, S. Maynard, and A. Ahmad, "The Internet of Things and Its Impact on Individual Privacy: An Australian Privacy Principle Perspective," *Computer Law & Security Review*, vol. 21, no. 1, pp. 4–15, 2016.

[3] D. McDermid, *Ethics in ICT: an Australian Perspective*. Pearson Higher Education AU, 2015.

[4] T. Laurence, *Blockchain for Dummies*. John Wiley & Sons, 2017.

[5] A. Bogner, M. Chanson, and A. Meeuw, "A Decentralised Sharing App Running a Smart Contract on the Ethereum Blockchain," in *Proceedings of the 6th International Conference on the Internet of Things*. ACM, 2016, pp. 177–178.

[6] R. Beck, J. S. Czepluch, N. Lollike, and S. Malone, "Blockchain-the Gateway to Trust-Free Cryptographic Transactions." in *ECIS*, 2016, p. ResearchPaper153.

[7] R. Lewis, J. McPartland, R. Ranjan *et al.*, "Blockchain and Financial Market Innovation," *Economic Perspectives*, no. 7, pp. 2–12, 2017.

[8] M. A. Khan and K. Salah, "IoT Security: Review, Blockchain Solutions, and Open Challenges," *Future Generation Computer Systems*, 2017.

[9] X. Leng, K. Mayes, and Y. Lien, "Ownership Management in the Context of the Internet of Things," in *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2014 International Conference on*. IEEE, 2014, pp. 150–153.

[10] Y. Xie and D. Wang, "An Item-Level Access Control Framework for Inter-System Security in the Internet of Things," in *Applied Mechanics and Materials*, vol. 548. Trans Tech Publ, 2014, pp. 1430–1432.

[11] A. L. M. Neto, A. L. Souza, I. Cunha, M. Nogueira, I. O. Nunes, L. Cotta, N. Gentille, A. A. Loureiro, D. F. Aranha, H. K. Patil *et al.*, "AoT: Authentication and Access Control for the Entire IoT Device Life-Cycle," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 2016, pp. 1–15.