Towards a New Model to Secure IoT-based Smart Home Mobile Agents using Blockchain Technology

Badr Eddine Sabir
Laboratory of Watch for Emergent Technologies
FST, Hassan I University
Settat, Morocco
b.sabir@uhp.ac.ma

Mohamed Youssfi
Laboratory of SSDIA
ENSET, University of Hassan II Casablanca
Mohammedia, Morocco
med@youssfi.net

Omar Bouattane
Laboratory of SSDIA
ENSET, University of Hassan II Casablanca
Mohammedia, Morocco
o.boattane@gmail.com

Hakim Allali
Laboratory of Watch for Emergent Technologies
FST, Hassan I University
Settat, Morocco
hakim-allali@hotmail.fr

Abstract—**The Internet of Things (IoT) is becoming an indispensable part of the actual Internet and continues to extend deeper into the daily lives of people, offering distributed and critical services. Mobile agents are widely used in the context of IoT and due to the possibility of transmitting their execution status from one device to another in an IoT network, they offer many advantages such as reducing network load, encapsulating protocols, exceeding network latency, etc. Also, the Blockchain Technology is growing rapidly allowing for the addition of an approved security layer in many areas. Security issues related to mobile agent migration can be resolved with the use of Blockchain. This paper aims to demonstrate how Blockchain Technology can be used to secure mobile agents in the context of the IoT using Ethereum and a Smart Contract. The transactions within the Blockchain are used to detect the malevolent mobile agents that could infiltrate the IoT systems. The proposed model aims to provide a secure migration of mobile agents to ensure security and protect the IoT applications against malevolent agents. The case of a smart home with multiple applications is applied to verify the proposed solution. The model presented in this paper could be extended to a wider selection of IoT systems outside of the smart home.**

Keywords-internet of things; smart home; blockchain; ethereum; smart contract; solidity; multi-agent systems; mobile agents

I. INTRODUCTION

IoT is growing exponentially in the area of telecommunications and it will be an indispensable part of the future Internet [1-4]. It is referring to an approach where an extensive number of physical objects are interconnected and connected to the Internet [5]. It is a part of pervasive and ubiquitous computing networks offering distributed and transparent services [6]. IoT enables heterogeneous devices to interconnect to support various applications to serve users with different requirements [7-8] and is considered to be a good way to achieve Smart City [9-10]. Mobile agents are widely used in the context of the IoT and due to the possibility of transmitting their execution status in an IoT network, they offer many advantages such as reducing network load, encapsulating protocols and exceeding network latency. An agent-oriented infrastructure enables flexible coordination between IoT devices including robots, smartphones, and sensors. Agent-based systems enable cognitive management without constant human intervention [11-12]. Functionalities such as smartness, autonomy and dynamicity are required for IoT based infrastructure and can be offered by the presence of agents [11]. Besides using agents, processing can be performed closer to actual data sources to reduce the cost of processing [13]. The use of mobile agents in an IoT network is highly recommended due to their advantages [14].

Authors in [7] showed that mobile agents can be represented by mobile JavaScript code (AgentJS) that can be modified at run-time by agents that are processed by a modular and portable agent platform JAM in a protected sandbox environment encapsulating agent processes. The proposed approach enables agents to migrate between different host platforms including WEB browsers by migrating the program code of the agent, embedding the state and the data of an agent, too, in an extended JSON+ format. Authors in [4] presented mobile agents based on web technologies in the context of IoT. In the proposed approach, agents can move between different devices, and if necessary, it is also possible to clone agents to create numerous instances. This model enables the creation of increasingly complex configurations, where device and context-specific decisions can also be taken. This approach increases the flexibility of the system design and evolution of IoT since the new code can add new functionalities and adapt the device to new requirements. Moving code and especially agents can also be used to add autonomous intelligence to systems. Authors in [15] propose a distributed software-defined multi-agent architecture for unifying IoT applications. This

Corresponding author: Badr Eddine Sabir

architecture can tackle the main challenges that the IoT faces, including heterogeneity, interoperability, scalability, flexibility, and security of IoT applications. Authors in [16] present an architecture based on MAS, SOA, and Semantic Web technologies to automate the integration and management of devices in IoT environment. A prototype system was implemented and tested in a simulated environment of a manufacturing context. In this context, the system demonstrated the ability to adapt incorporating new features and flexibility. In [17], a framework for IoT was presented which uses mobile agents for information transfer. The proposed approach can update information like the availability and usability of services dynamically. It also has speech processing modules to provide solutions using voice-based commands and prompts.

Mobile agents are widely used in the field of IoT. Therefore, the questions of how to ensure security during the process of migration of the mobile agents and protect the IoT application against malevolent agents arise. Authors in [18] describe four kinds of threat scenarios faced while using mobile agents: Agent corrupts the Platform (AcP), Platform corrupts Agent (PcA), Agent corrupts Agent (AcA) other malicious entities (third-party programs) corrupt Agent (OcA). In this paper, we present an architecture model to secure mobile agents and protect them against different types of threats in the context of the IoT. The proposed model aims to provide a secure migration of mobile agents to ensure security and protect the IoT applications against malevolent agents using a Smart Contract [19]. The data in the Blockchain is unchangeable once published and can help neutralize any attempt to altering the agent code source. The Blockchain approach provides the logging of events in a tamper-proof manner which allows us to detect any mobile agent that has turned malicious. It can be used to provide high trust in secure transactions in a heterogeneous network [20]. Once such a mobile agent is detected, the security agent isolates and destroys it. The case of a smart home can be a very suitable application of IoT to verify the proposed solution, as it involves a variety of devices and parameters to be connected [21], however, the presented model could be extended to a wider selection of IoT systems.

II. RELATED WORK

In this section, the related work is described with regard to the security of mobile agents in a Multi-agent environment. Authors in [13] provide a methodology to improve BROSMAP and make it a lightweight protocol to fulfill the needs of Multi-agent based IoT systems in general. They offer a new ECC-BROSMAP which is equivalent in security with the RSA-BROSMAP and implement both RSA-BROSMAP and ECC-BROSMAP before presenting a comparative performance study and implementation results of ECC-BROSMAP against RSA-BROSMAP. Authors in [22] illustrate the use of mobile agent systems in distributed applications in the domain of Ambient Intelligence. They focus on the ability to improve privacy by hiding information using the agent architecture. The shown scenario clarifies the necessity to consider the particular security requirements of mobile agents. Authors in [23] introduce Agent Identity in distributing the Symmetric Key to the newly attached Agents of the Platform. During the registration of the Agent and its Services at the Platform, it must obtain the key from the Key Distribution Process. The authors proposed a novel idea in fixing the Identity of the Agent for getting its Shared Key from the Key Distribution process. Every Trusted Agent in a platform has a tiny hardware called USB Dongle, which is password protected. It is configured during the initial environment formation. Authors in [24] propose a security framework that can be effectively used to protect agents from attacks by malicious hosts. The framework is based on restricting the access level of the agent according to the trust level that is assigned to the current host. Certain methods can only be executed on certain hosts that are minimally trusted. Methods that cannot be executed on a host are kept encrypted. Data are also selectively accessible according to the trust placed on the host.

In traditional security methods, discovering the determined private key is enough for message decryption that can be done through malicious attacks to the network nodes or listening to communication links. Accessing the private key can be considered as the endpoint of a malicious process. Authors in [25] propose an approach to improve private key security using two strategies: encrypting the private key using an encryption algorithm (AES algorithm is used in this paper) and breaking the encrypted private key into different units. A secure authentication model based on IBC for the multi-agent of a single domain, giving a second authentication model based on IBC for multi-agent of multi-domain was proposed in [26]. Authors in [27] described the general security requirements for mobile agent systems and existing security measures. Especially, they pointed out some weaknesses in the field of protecting the carried data of mobile agents. To mitigate this issue, they implemented a trust and reputation management to provide a secure path for mobile agent data protection.

Our work aims to present a simple approach to guarantee the security of the migration of mobile agents in an IoT network by ensuring maximum flexibility and without degrading performance.

III. BACKGROUND

A. Blockchain Technology

In 2008 the Blockchain was first propagated through bitcoin [28] to assure all parties that the payer had the means to satisfy the debt before concluding any transaction [29]. Basically, bitcoin was created with Blockchain technology to transfer money, but now Blockchain is used in many other areas. A Blockchain [28] is a decentralized distributed database that maintains a continuously growing list of data in a public or a private peer-to-peer network. Duplicated to all the peer nodes of the network, Blockchain offers a secured system, between untrusted collaborators, which everyone on the network can check and interact with, but no one can control or alter. This allows the Blockchain to be a trustworthy source without the requirement of a third-party [30]. A Blockchain is a series of block*s*. **E**very block has itw own cryptographic hash code, previous block hash, and its data [31]. As shown in Figure 1, each block in the Blockchain is connected to the previous block, containing a hash of the previous block. As a result, the history of transactions on the Blockchain cannot be altered or deleted without completely changing the content of the Blockchain [32]. A Blockchain network is formed by one or more nodes. A node can be any electronic device (a computer, a telephone, etc.) if it is connected to the Internet and has an IP address, and each node has a complete and separate copy of the Blockchain. All these nodes connect to form a Blockchain network. A transaction is not sent to the network but rather to a network node that it communicates with the other network nodes.

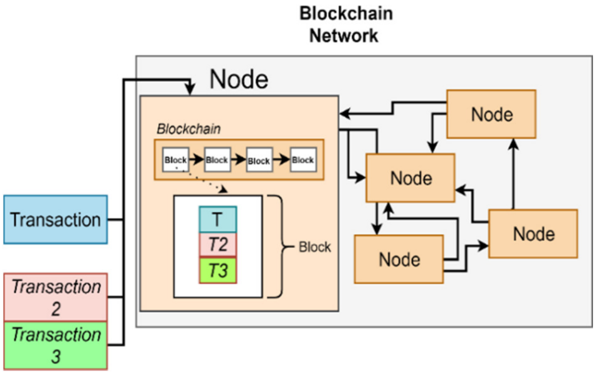

Fig. 1. Simplified diagram of a Blockchain network

The Blockchain has several characteristics, among which we can cite [33]:

- Decentralization: In the Blockchain, third parties are not required to verify transactions. Consensus algorithms and cryptographic mechanisms are used to maintain data consistency on Blockchain networks.
- Persistency: It is not possible to delete transactions that have already occurred.
- Auditability: Each transaction on the Blockchain refers to the previous transaction. This makes it easy to verify and track each transaction.

B. Ethereum

Ethereum is a Blockchain platform which surpasses some limitations of Bitcoin [19]. It allows users to run distributed applications in a decentralized manner. This means that applications running on Ethereum are available everywhere and every time [34-35]. Ethereum has several elements, the most important of them are [36]:

- Account: Every account on Ethereum has a 20-byte address and consists of four parts, namely nonce-counter, storage, ether balance, and contract code.
- Transaction: Transaction in Ethereum refers to a signed data package that stores messages.
- Technology used: Ethereum uses several technologies including web technology, client/node implementation, and data storage.
- Consensus algorithm: Ethereum has 3 types of consensus algorithms, namely Proof Stake (PoS), Proof of Authority (PoA), and Proof of Work (PoW). The most common one is PoW. The underlying principle in this consensus algorithm is the complicated mathematical puzzles which consume a certain amount of power to find the solution but whose verification is comparatively fast and easy. The process of finding a solution to the puzzle is known as Mining, while the nodes executing this process are known as Miners. If the Miner manages to find the solution (hash), the new block is formed, which is distributed on the network and if validated, the blocks get added, extending the chain. The protocols used for generating the hash for every block are cryptographic hash algorithms like SHA256 which compute the hash of the current block considering metadata like the hash of the previous block. This makes each hash unique and any attempt to change the content or metadata of a block will result in an entirely different hash generating a diversion in the chain [30].

C. Smart Contract

It was introduced in 1994 and defined a Smart Contract as a computerized transaction protocol that executes the terms of a contract [37]. Translating contractual clauses into code and embedding them into a property that can self-enforce them was suggested in [38]. Within the Blockchain context, Smart Contracts are scripts recorded on the Blockchain [39]. Since they reside on the chain, they have a unique address. We trigger a Smart Contract by addressing a transaction to it. It then executes independently and automatically in a prescribed manner on every node in the network, according to the data that were included in the triggering transaction [40]. The nodes in the network interact with the contract by requesting the functions of the contract code once it is deployed on the network. Smart Contracts are invulnerable and cannot be altered, even by the author, once deployed on the network. Smart Contracts on Ethereum are written in a high-level language and compiled via the Ethereum Virtual Machine. The most used programming language is Solidity [41] which we will use to write our own Smart Contracts.

IV. PROPOSED MODEL

In recent years, we have witnessed an impressive development of the IoT home devices. Home automation is a system controlled by a smart device. It can control home appliances such as lights, fans, air conditions, smart security locks, etc [42]. Many companies such as Google Home, Amazon Echo, and Samsung SmartThings have released innovative new products that allowed these devices to become widely available. While these devices have a variety of benefits, they also introduce a new target for potential security threats [43]. Device providers do not think about device security, supposing that devices in the home environment are trustworthy, and since the consumers do not have the resources to protect themselves from targeted security attacks on their home network, it becomes vulnerable to a variety of potential security threats. So, there is a real need for an intelligent and efficient home security model.

A. Components of the Proposed Model

The proposed model shown in Figure 2 aims to provide a migration of mobile agents while ensuring security and protection of the IoT application against malevolent agents by ensuring non-repudiation and their integrity using Blockchain technology. The main components of our model are:

- Agents: An agent can be attached to a source device to collect information or perform actions on the source device. An agent can migrate to another device keeping its state to perform operations on the destination device.
- Smart device: Extremely useful devices that are making our daily life easier [44] allowing users to configure, access and, control IoT devices through a user-friendly interface.

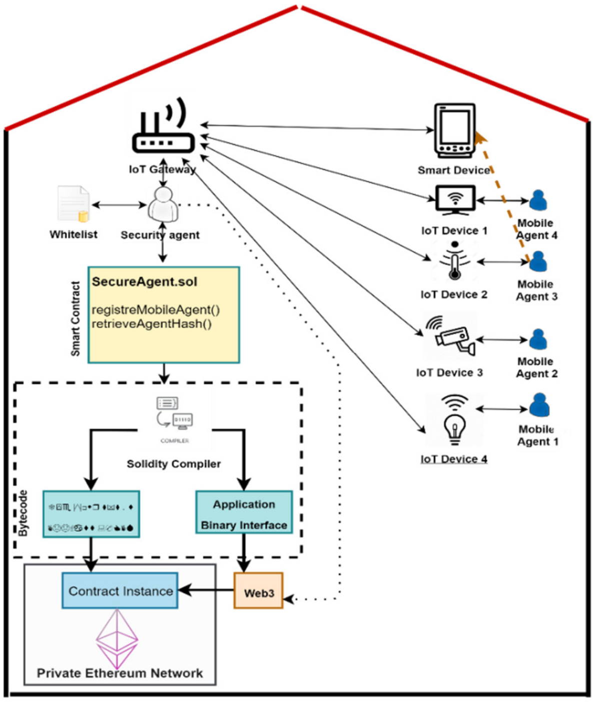

Fig. 2. Proposed model architecture

- Whitelist: It contains hashes of transactions returned by the Ethereum network after the registration of a new agent. An Agent Identifier (AID) is associated with a transaction hash.
- IoT Gateway: A user can access and control the IoT devices using a smart device by accessing the IoT gateway. The gateway is responsible for authenticating and monitoring communication between devices, requesting the Security Agent in the event of an agent migration.
- Security Agent: Mainly performs the operations of registering the source code hash of agents who wish to migrate to other devices in the Blockchain network and of the verification of the agents after their migration to guarantee integrity, authentication and non-repudiation agents.
- IoT device: In our architecture, the IoT device is a piece of hardware with a sensor.
- SecurityAgent.sol: It is a Smart Contract that provides two functions:
    - registerMobileAgent (AID, AgentHash): this function allows sending a transaction to the Blockchain network to register the source code of the agent. The function takes in two parameters, the hash of the source code of the agent (AgentHash) and the Agent Identifier (AID) and returns the identifier of the transaction recorded in the Blockchain (tranactionHash).
    - retrieveAgentHash (transactionHash): this function allows us to recover the hash of an agent's source code from the Blockchain network. The function takes the identifier of the transaction that allowed registering the hash of the source code of the agent and returns the hash of the source code of the agent.

This Smart Contract is developed in Solidity, which is a Contract Oriented Language, used for writing Smart Contracts that can be deployed on an Ethereum Virtual Machine. It follows an object-oriented approach and supports features like inheritance and complex data types.

- Solidity Compiler: SecurityAgent.sol is what we call "Contract Definition". This code is not executed on the Ethereum network, thus we need to compile our "Contract Definition" using a "Solidity Compiler" which will produce two separate files. A file which will contain "Byte Code" which will then be deployed on the Ethereum network in the form of a contract instance using in our case Truffle which is a development environment and a testing framework that helps the automatic compilation and deploying of contracts on Blockchain, and is also used to deploy contracts on a private Ethereum Blockchain. The compilation will produce an "Application Binary Interface (ABI)". This ABI will be used to call the functions exposed by the Smart Contract instance deployed in the Ethereum network using the Web3 library. Ethereum provides an interface with the Ethereum network for developers in the web3.js API. This allows applications to interpret events sent from the Ethereum network and to submit transactions to the network [45].

B. Secure Communication Between Agents

The diagram shown in Figure 3 illustrates the migration steps of an agent from a Temperature Sensor to a Smart Device in chronological order:

- The Mobile Agent measures and collects information from the Temperature Sensor.
- The Mobile Agent requests the IoT Gateway to migrate to the Smart Device.
- The Security Agent generates the hash of the source code of the Mobile Agent.
- The Security Agent invokes the "registerMobileAgent" function of the SecureAgent.sol Smart Contract, with passing in parameters the hash of the Mobile Agent source code and the AID of the agent.
- The hash of the agent source code is registered in Blockchain Ethereum Network.

- The result of this operation is an identifier of the transaction validated on the Blockchain in the form of a hash of the transaction.
- The Security Agent registers the hash of the transaction in the whitelist by associating it with the AID of the Mobile Agent for which the source code has been saved in the Blockchain.
- The IoT Gateway allows the Mobile Agent to migrate.
- After the migration and the generation of graphs in the Smart Device, the Mobile Agent requests to return to the Temperature Sensor.
- The IoT Gateway requests the Security Agent to verify the integrity and the authentication of the Mobile Agent after its migration.
- The Security Agent retrieves the transaction identifier from the whitelist passing in the AID of the Mobile Agent.

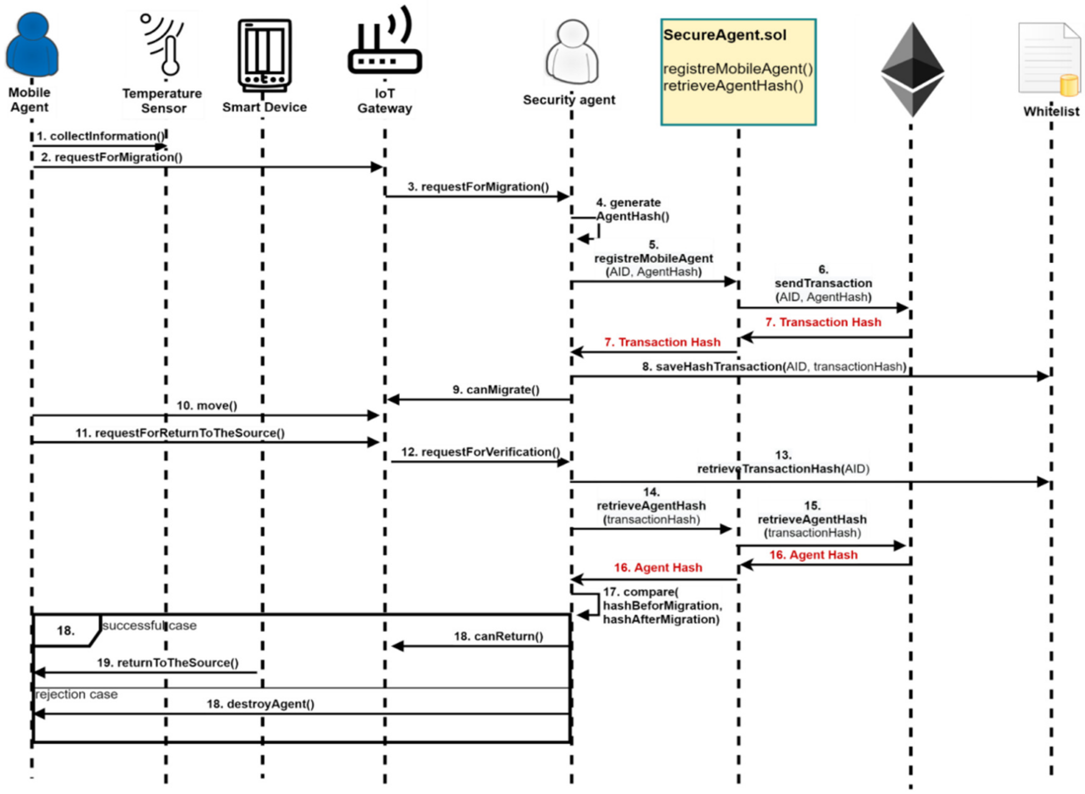

Fig. 3. Sequence diagram: Migration of an agent

- The Security Agent invokes the “retrieveAgentHash” function of the SecurAgent.sol Smart Contract passing the transaction identifier.
- The hash of the source code of the Mobile Agent stored in the Blockchain is returned to the Security Agent.
- The Security Agent compares the hash recovered from the Blockchain with the current Mobile Agent hash.
- If the hashes are same, then the Security Agent allows the Mobile Agent to return to the Sensor Temperature.
- Else the Security Agent destroys the Mobile Agent.

V. CONCLUSION

This document presents an overview of the current state of research in the field of mobile agent security in a multi-agent environment, and the utility of employing Mobile Agents in IoT systems such as reducing network load, protocol encapsulation and exceeding network latency. An architecture using the Blockchain technology has been presented in this document to secure mobile agents and protect them against different types of threats in the context of the IoT using a Smart Contract deployed in a private Ethereum network. The smart home use case with multiple IoT devices using mobile agents was applied to verify and explain the proposed solution. Though this document presents the smart home use case, there is a potential for this model to be extended to other types of IoT systems with some modifications. Our future work aims to set up a private Ethereum network using Go Ethereum before testing the implementation of the proposed model, and to this end, we are in the process of developing a smart home testbed environment based on web technologies.

REFERENCES

[1] T. Alam, M. Benaida, “CICS: Cloud–Internet Communication Security framework for the internet of smart devices”, International Journal of Interactive Mobile Technologies, Vol. 12, No. 6, pp. 74-84, 2018

[2] S. Li, L. D. Xu, S. Zhao, "The internet of things: A survey", Information Systems Frontiers, Vol. 17, No. 2, pp. 243-259, 2015

[3] S. K. Anithaa, S. Arunaa, M. Dheepthika, S. Kalaivani, M. Nagammai, M. Aasha, S. Sivakumari, "The internet of things: A survey", World Scientific News, Vol. 41, pp. 150-158, 2016

[4] M. Weyrich, C. Ebert, "Reference architectures for the Internet of Things", IEEE Software, Vol. 33, No. 1, pp. 112-116, 2016

[5] L. Jarvenpaa, M. Lintinen, A. L. Mattila, T. Mikkonen, K. Systa, J. P. Voutilainen, "Mobile agents for the Internet of Things", 17th International Conference on System Theory, Control and Computing, Sinaia, Romania, October 11-13, 2013

[6] S. Bosse, "Mobile multi-agent systems for the internet-of-things and clouds using the javascript agent machine platform and machine learning as a service", 4th International Conference on Future Internet of Things and Cloud, Vienna, Austria, August 22-24, 2016

[7] D. Lake, A. Rayes, M. Morrow, "The Internet of Things", The Internet Protocol Journal, Vol. 15, No. 3, pp. 10-19, 2012

[8] G. M. Lee, J. Y. Kim, "The Internet of Tthings: A problem statement", International Conference on Information and Communication Technology Convergence, Jeju, South Korea, November 17-19, 2010

[9] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, "Internet of Things for smart cities", IEEE Internet of Things Journal, Vol. 1, No. 1, pp. 22-32, 2014

[10] J. Jin, J. Gubbi, S. Marusic, M. Palaniswami, "An information framework for creating a smart city through Internet of Things", IEEE Internet of Things Journal, Vol. 1, No. 2, pp. 112-121, 2014

[11] G. Fortino, A. Guerrieri, W. Russo, C. Savaglio, "Middlewares for smart objects and smart environments: Overview and comparison", in: Internet of Things Based on Smart Objects, pp. 1-27, Springer, 2014

[12] F. Aiello, G. Fortino, A. Guerrieri, R. Gravina, Maps: A mobile agent platform for WSNS based on java sun spots, University of Calabria, 2009

[13] H. Hasan, T. Salah, D. Shehada, M. J. Zemerly, C. Y. Yeun, M. A. Qutayri, Y. A. Hammadi, "Secure lightweight ECC-based protocol for multi-agent IoT systems", 13th International Conference on Wireless and Mobile Computing, Networking and Communications, Rome, Italy, October 9-11, 2017

[14] H. Yu, Z. Shen, C. Leung, "From Internet of Things to Internet of Agents", International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China, August 20-23, 2013

[15] L. Jarvenpaa, M. Lintinen, A. L. Mattila, T. Mikkonen, K. Systa, J. Voutilainen, "Mobile agents for the Internet of Things", 17th International Conference on System Theory, Control and Computing, Sinaia, Romania, October 11-13, 2013

[16] R. L. Cagnin, I. R. Guilherme, J. Queiroz, B. Paulo, M. F. O. Neto, "A multi-agent system approach for management of industrial IoT devices in manufacturing processes", 16th International Conference on Industrial Informatics, Porto, Portugal, July 18-20, 2018

[17] P. Verma, M. Gupta, T. Bhattacharya, P. K. Das, "Improving services using mobile agents-based IoT in a smart city", International Conference on Contemporary Computing and Informatics, Mysore, India, November 27-29, 2014

[18] D. Calvaresi, A. Dubovitskaya, J. P. Calbimonte, K. Taveter, M. Schumacher, "Multi-agent systems and blockchain: Results from a systematic literature review", in: Lecture Notes in Computer Science, Vol 10978, pp. 110-126, Springer, 2018

[19] V. Buterin, Ethereum white paper. A next-generation smart contract and de-centralized application platform, 2014

[20] T. Alam, "IoT-Fog: A communication framework using blockchain in the internet of things", International Journal of Recent Technology and Engineering, Vol. 7, No. 6, pp. 1-5, 2019

[21] V. Tiwari, A. Keskar, N. C. Shivaprakash, "Design of an IoT enabled local network based home monitoring system with a priority scheme", Engineering, Technology & Applied Science Research, Vol. 7, No. 2, pp. 1464-1472, 2017

[22] F. Piette, C. Caval, A. E. F. Seghrouchni, P. Taillibert, C. Dinont, "A multi-agent system for resource privacy: Deployment of ambient applications in smart environments", International Conference on Autonomous Agents & Multiagent Systems, Malaysia, Singapore, May 9–13, 2016

[23] R. Kumaravelu, N. Kasthuri, "Distribution of shared key (secret key) using USB dongle based identity approach for authenticated access in mobile agent security", International Conference on Communication and Computational Intelligence, Erode, India, December 27-29, 2010

[24] P. J. Marques, L. M. Silva, J. G. Silva, "Establishing a secure open-environment for using mobile agents in electronic commerce", in: Proceedings. First and Third International Symposium on Agent Systems Applications, and Mobile Agents, IEEE, 1999

[25] A. Esfandi, A. M. Rahimabadi, "Mobile agent security in multi agent environments using a multi agent-multi key approach", 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, August 8-11, 2009

[26] Y. Yu, X. Zheng, M. Zhang, Q. Zhang, "An identity-based authentication model for mobile agent", Fifth International Conference on Information Assurance and Security, Xi'an, China, August 18-20, 2009

[27] G. Geetha, C. Jayakumar, "Implementation of trust and reputation management for free-roaming mobile agent security", IEEE Systems Journal, Vol. 9, No. 2, pp. 556–566, 2015

[28] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system", available at: https://bitcoin.org/bitcoin.pdf, 1997

[29] I. Purdon, E. Erturk, "Perspectives of blockchain technology, its relation to the cloud and its potential role in computer science education", Engineering, Technology & Applied Science Research, Vol. 7, No. 6, pp. 2340-2344, 2017

[30] I. Ishita, D. Kulkarni, T. Semwal, S. B. Nair, "On securing mobile agents using blockchain technology", Second International Conference on Advanced Computational and Communication Paradigms, Gangtok, India, February 25-28, 2019

[31] T. Alam, "Blockchain and its role in the internet of things (IoT)", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 5, No. 1, pp. 151-157, 2019

[32] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, P. Rimba, "A taxonomy of blockchain-based systems for architecture design", International Conference on Software Architecture, Gothenburg, Sweden, April 3-7, 2017

[33] Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends", International Congress on Big Data, Honolulu, USA, June 25-30, 2017

[34] C. Dannen, Introducing Ethereum and Solidity: Foundations of cryptocurrency and blockchain programming for beginner, Apress, 2017

[35] D. Patel, J. Bothra, V. Patel, "Blockchain exhumed", ISEA Asia Security and Privacy, Surat, India, January 29-February 1, 2017

[36] C. Saraf, S. Sabadra, "Blockchain platforms: A compendium", IEEE International Conference on Innovative Research and Development, Bangkok, Thailand, May 11-12, 2018

[37] D. Tapscott, A. Tapscott, Blockchain revolution: How the technology behind bitcoin and other cryptocurrencies is changing the world, Penguin, 2018

[38] N. Szabo, "The idea of smart contracts", available at: https://nakamotoinstitute.org/the-idea-of-smart-contracts, 1997

[39] "Using stored routines (procedures and functions)", in: MySQL reference manual, Oracle, 2016

[40] S. J. Pee, J. H. Nang, J. W. Jang, "A simple blockchain-based peer-to-peer water trading system leveraging smart contracts", International Conference on Internet Computing and Internet of Things, Las Vegas, USA, July 27-30, 2018

[41] M. Wohrer, U. Zdun, "Smart contracts: Security patterns in the Ethereum ecosystem and Solidity", International Workshop on Blockchain Oriented Software Engineering, Campobasso, Italy, March 20, 2018

[42] T. Alam, A. A. Salem, A. O. Alsharif, A. M. Alhejaili, “Smart home automation towards the development of smart cities”, APTIKOM Journal on Computer Science and Information Technologies, Vol. 3, No. 1, pp. 1-2, 2020

[43] L. Rafferty, F. Iqbal, S. Aleem, Z. Lu, S. C. Huang, P. C. K. Hung, “Intelligent multi-agent collaboration model for smart home IoT security”, IEEE International Congress on Internet of Things, San Francisco, USA, July 2-7, 2018

[44] T. Alam, “Middleware implementation in cloud-MANET mobility model for internet of smart devices”, International Journal of Computer Science and Network Security, Vol. 17, No. 5, pp. 86-94, 2017

[45] V. P. Ranganthan, R. Dantu, A. Paul, P. Mears, K. Morozov, “A decentralized marketplace application on the ethereum blockchain”, 4th International Conference on Collaboration and Internet Computing, Philadelphia, USA, October 18-20, 2018