Implementing a Forms of Consent Smart Contract on an IoT-based Blockchain to promote user trust

Charalampos S. Kouzinopoulos[1], Konstantinos M. Giannoutakis[1], Konstantinos Votis[1],
Dimitrios Tzovaras[1], Anastasija Collen[2], Niels A. Nijdam[2], Dimitri Konstantas[2],
Georgios Spathoulas[3]*, Pankaj Pandey[3], Sokratis Katsikas[3]
[1]*Information Technologies Institute, Centre for Research and Technology Hellas*, Thessaloniki, Greece
[2]*Centre Universitaire d'Informatique, University of Geneva*, Geneva, Switzerland
[3]*Center for Cyber and Information Security, Norwegian University of Science and Technology*, Gjøvik, Norway
[1]{kouzinopoulos,kgiannou,kvotis,dimitrios.tzovaras}@iti.gr,
[2]{anastasija.collen,niels.nijdam,dimitri.konstantas}@unige.ch, *gspathoulas@dib.uth.gr,
[3]{pankaj.pandey,sokratis.katsikas}@ntnu.no

***Abstract*—The H2020 European research project Safe-Guarding Home IoT Environments with Personalised Real-time Risk Control (GHOST) aims to develop a cyber-security layer on IoT smart home installations. The proposed system analyses packet-level data flows for building patterns of communications between IoT devices and external entities. To ensure non-repudiation, integrity and authentication of the data captured, they are stored in a Blockchain, a distributed ledger network, as digitally-signed transactions. Since the data can potentially include sensitive user information, it is imperative to promote trust by informing users about the operating principles of the network as well as to request the acceptance of a consent form by them. This paper presents the design and implementation of a Forms of Consent application, a Distributed Application that interacts with a set of Smart Contracts deployed on a private Ethereum network. The application is being developed as part of the GHOST project.**

***Index Terms*—blockchain, ethereum, smart contracts, cyber security, user trust**

I. Introduction

Internet of Things (IoT) or Internet of Objects as it is commonly referred as, aims to bridge the gap between the physical and the digital world. It is a network of interconnected devices, including sensors and other embedded hardware, that capture, collect and distribute data, between each other or with other entities, via a local network and the Internet. IoT technologies had a tremendous growth during the past few years. According to Gartner, it is estimated that the IoT sector will include 26 billion units installed by 2020, up from 0.9 billion in 2009. By that time, IoT product and service suppliers will generate an incremental revenue exceeding US$300 billion, mostly in services [1]. Based on a different forecast by Businessinsider [2], by 2020 IoT devices will account for 24 billion out of the 34 billion devices connected to the internet. The same forecast predicts that more than US$6 trillion will be spent on IoT solutions between 2015 and 2020.

However, IoT networks are extremely vulnerable to external threats and attacks by malicious users. As discussed in [3], it is easy to gain physical access to individual devices since they are often isolated and there is no administrator to manage them; IoT devices usually communicate with each other and with a gateway using different wireless communication protocols, making eavesdropping very easy; and most devices have low processing capabilities and thus it is difficult to implement complex security schemes on a per device basis. Moreover, IoT devices generally fail to provide basic security; many times default passwords are used, there is a possibility of account enumeration attacks and other trivial issues still exist. Due to their inherent lack of security and the fact that often IoT devices capture and collect sensitive user data, there are raising concerns over the past years regarding the privacy of their users.

For the growth of the IoT environments to continue to increase, it is imperative for the users involved to trust that their privacy is maintained and that the security and safety of the installed devices is up to higher standards. On the other hand though, the security and privacy issues mentioned above, remain two of the main challenges in IoT environments, threatening user trust. The decentralized deployment, high connectivity, diversity and heterogeneity of IoT devices results in a number of security and privacy challenges, that in turn induce trust requirements. According to [4], these requirements are divided into device, entity, and data trust. Device trust is confidence on the proper function of a device, a confidence though that cannot always be established, entity trust refers to the expected behavior of participants such as persons or services, while data trust is certainty on the authenticity of data independently on the originating device or the medium through which they were communicated.

GHOST is a H2020 European research project that aims to develop a cyber-security layer on top of IoT smart home installations. The proposed system analyses packet level data flows for building patterns of communications between IoT devices and external entities. The project aims to address different security and privacy concerns of IoT device installations in smart homes in combination with the exploitation of the Blockchain technology, using a number of solution vectors, including an IP blacklisting scheme, a software integrity

978-1-5386-5150-6/18/$31.00

mechanism as well as a consent form implementation.

This paper explores the design and implementation of a Forms of Consent application for the GHOST project. It is developed in order to enhance the trust perception of the users in an IoT environment, has a modular architecture and enables interoperability with future applications developed as part of the GHOST project. The application is built on top of Ethereum, a distributed ledger Blockchain protocol that forms the backbone of the GHOST Blockchain network and is implemented based on Smart Contracts as well as a number of different web application standards, including HTML, CSS and Javascript.

The rest of the paper is organized as follows. Related work regarding the use of Blockchain in IoT environments is summarized in Section II. Section III gives an overview of the GHOST project, discusses the concepts of Blockchains and Smart Contracts and details the design and implementation of the back-end and front-end modules for the Forms of Consent application in GHOST. Concluding remarks and future work are discussed in Section IV.

II. Related Work

The Blockchain technology uses public-key cryptography to create an immutable chain of blocks of transactions. Its inherently secure nature can be used to strengthen the security and privacy of IoT device networks. There is ongoing research on the technology itself and also on its integration with the IoT domain.

A 2016 literature review [5] identified various research papers that use the Blockchain technology in other areas beyond cryptocurrencies, including data storage management, data trading, identity management and others. In all cases, data exchanged between IoT devices are stored as unique transactions within the Blockchain network and are subsequently distributed among the participating nodes, ensuring the integrity and security of the communication between them.

The security enhancements that can be achieved in IoT with the use of Blockchain are given in [6]. The role of Blockchain is examined through four challenges; costs and capacity constraints, architecture, unavailability of services and susceptibility to manipulation. The authors conclude that given the decentralized and consensus-driven structure of Blockchain, more secure IoT ecosystems can be provided as the network size increases.

The use of Blockchain in a smart home installation of IoT devices was discussed in [7]. In this research it was shown that a proposed Blockchain-based framework promoted the security of smart homes in terms of confidentiality, integrity, and availability with only a comparatively low overhead introduction.

In [8], the subject of IoT devices firmware upgrade process was detailed and it was shown how this process can easily be hijacked by a malicious user resulting to the compromise of the security of the network. Then, a new firmware upgrade scheme was proposed that combines a Blockchain with a BitTorrent peer-to-peer network to ensure that the devices' firmware is always up-to-date while not tampered.

A security framework that integrates the Blockchain technology with smart devices to provide a secure communication platform in a smart city was presented in [9].

There is also ongoing research on the use of Distributed Applications (Dapps) on top of the Ethereum Blockchain network. An identity management system built on top of the Ethereum blockchain platform that provides a recovery mechanism for user identity in a robust way was discussed in [10]. A decentralized record management system to handle electronic medical records and perform permission management with the use of Ethereum, Smart Contracts and Dapps was extensively analyzed in [11]. In that research paper was shown the benefits of this approach in terms of auditability, interoperability and accessibility. Finally, a platform for transactive IoT blockchain applications with repeatable testing that can be used to develop, test, and analyze fault-tolerant IoT Blockchain applications was demonstrated in [12].

III. Design and Implementation

A. Overview

As part of the operation of a smart home installation of IoT devices, sensitive user data can be captured, such as presence information, surveillance cameras' video or records from medical devices. For this reason, it is important to inform participating users, such as home residents, employees or patients, about the operating principles of the network, thus promoting user trust. Moreover, it is essential for the proper functioning of the network to request the acceptance of said principles by the users. This procedure can be performed by digital signing a consent form through a Dapp that is connected to a set of Smart Contracts deployed to the GHOST Blockchain network. The introduction of a digital Form of Consent over a traditional on-paper approach has the following benefits:

- *Automation*: The users are required to sign the form upon their first connection to the network and additionally every time the terms of the network are modified. The devices of the users are only allowed to operate on the GHOST network after the user has signed the most recently issued Form of Consent.
- *Non-repudiation*: Every user signature is stored digitally on the Blockchain as part of a transaction block and is eventually distributed to each node of the network. That way, it is ensured that no party can deny in the future that such a signature took place.

This Section presents details on the design and implementation of the Forms of Consent application.

B. The GHOST project

The GHOST project aims to deploy a highly usable and effective security framework for smart home residents. The project applies behavioural design principles for the elaboration of a novel reference architecture for user-centric cyber security in smart home environments.

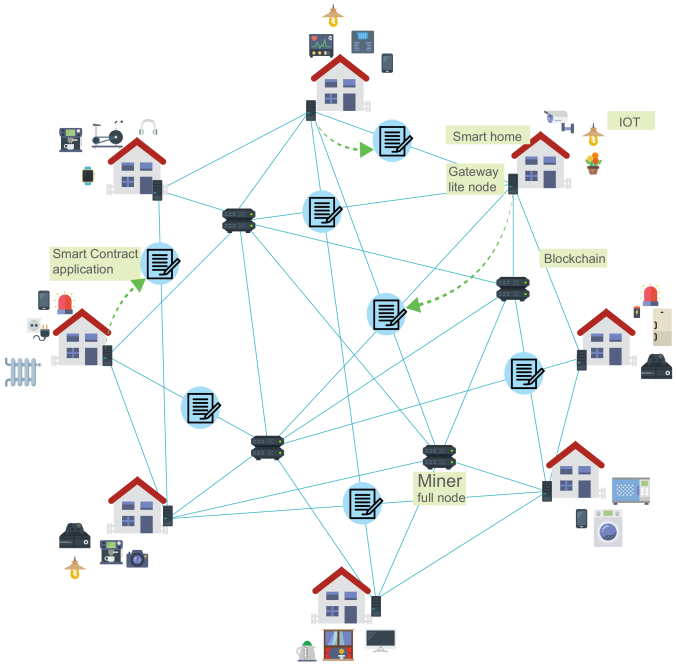

Fig. 1. The GHOST Blockchain architecture

GHOST includes the Data Interception and Inspection layer that is responsible to gather, aggregate and analyze data; the Contextual Profiling layer that builds behaviour trees for devices and data-flows and provides current state of data identification and related behaviour; the Risk Assessment layer that gathers information about the current risks and analyzes in real-time current network traffic flows; and the Control and Monitoring layer that presents a graphical interface to the end user.

Moreover different components are utilized, including the GHOST Blockchain Defence Infrastructure component that uses the Blockchain technology and Smart Contracts to ensure data integrity in the process of distributed decision making, as well as additional security countermeasures; the Cross Layer Anomaly Detection Framework where traditional cyber security features are exploited, extended and adapted for the needs of the smart home environment; the Cyber Security Knowledge Base that consists of a cloud-based knowledge repository to collect anonymized security intelligence and insights to enhance the automatic decision making and improve end-user visual experience within the Control and Monitoring layer; and the Shared Data Storage, a single-storage framework. A detailed analysis of the GHOST architecture can be found in [13].

The GHOST network consists of a full-scale deployment of a private Blockchain on top of an inter-connected grid of smart home installations of IoT devices. Each smart home features a device that will act as the smart home gateway middleware and simultaneously as a Blockchain node. Due to their limited processing power, this type of devices act as light nodes, therefore additional, full nodes with adequate processing power are required to act as miners, making the private Ethereum private blockchain functional. The GHOST Blockchain architecture is depicted in Figure 1.

C. Blockchain and Smart Contracts

A Blockchain is a linked list of records, called blocks, with each block being secured via encryption. It was originally developed as part of the Bitcoin cryptocurrency in order to maintain a financial publicly verifiable ledger. The Blockchain can be thought of as a singleton state-machine that can transition between states via cryptographically-secured transactions, as detailed in [11]. When generating a new state-machine, the nodes encode logic which defines valid state transitions and upload it onto the Blockchain. Then, the blocks journal a series of valid transactions that, when incrementally executed with the previous block's state, morph the state-machine into the current state.

The double spending problem is an inherent flaw of digital currencies, where the the same token can be spent more than one time and can only be solved with the use of a trusted central authority that validates every transaction. This problem is solved by the use of a Blockchain in a decentralized way by using the concept of consensus, with which all participating nodes in the network reach an agreement over the validity of the transactions even though they don't trust each other. A consensus protocol is used to enforce that the network users follow its rules and to ensure that the transactions are validated in the right order. It is also used to make sure that the information within a block is correct and that the nodes (miners) get a fair compensation for their transaction processing. A major difficulty with Blockchain is to make sure that the consensus protocol is reached by the participants of the entire network [14].

Backlund [15] argues that one way of ensuring authenticity is to let each user within the network get one vote and let all users vote which transaction should be included in the next block. The number of votes decides which set of transactions should be included. This kind of consensus-process is vulnerable to Sybil attacks though, where one user could create multiple accounts and get a higher influence within the network.

Bitcoin solved this issue of influence by adding a cost to the vote. Each user's amount of influence is based on the computing power of that user. The more computing power, the higher the needed energy and the higher the hardware costs. Bitcoin uses a Proof of Work consensus protocol, a mechanism to confirm a person's validity by assigning to them a certain work, which has a certain computational cost associated to it and its correctness can be easily verified. With this protocol, the network collects all the transactions made during a set period into a block. The nodes' task is to confirm those transactions and write them into the Blockchain, while hashing the information as well, to protect it from potentially malicious users. The nodes get economic incentives to keep mining and hashing, since more Bitcoin tokens are received as additional blocks are created.

A Smart Contract is a self-executing script deployed decentralized in a Blockchain network that supports it, such as Ethereum, and is publicly visible to all the users of the network. It can be defined as 'a computerized transaction protocol that executes the terms of a contract'. A contract can encode any set of rules represented in its programming language with its correctness enforced by the consensus protocol of the Blockchain [16]. Each contract is identified by its address, computed in a deterministic way from the address and the transaction count (nonce) of its creator as well as its Application Binary Interface (ABI). Every user of the Blockchain that has this information can submit transactions to the Smart Contract. Then, as detailed in [17], all miners of the network execute the contract code with the current state of the Blockchain and the transaction payloads as inputs. The network then agrees on the output and the next state of the contract by participating in a consensus protocol.

D. Digital Consent and personal data

As is evident from the Section above, Smart Contracts lack transactional privacy and this has to be kept in mind when designing Dapps. All actions and transactions taken in a Smart Contract are propagated to the miners across the network and can be recorded on the Blockchain, making them publicly visible. For this reason, no personal data are stored in the Blockchain that could potentially be used to uniquely identify an individual and their actions. As such, no actions are required in cases of data breach, or in cases pertaining user actions and private data such as the right by the users to obtain the erasure of personal data concerning them.

This is in agreement with the European General Data Protection Regulation (GDPR) process, a regulation on privacy and data protection for all individuals within the European Union. GDPR applies to the processing of personal data in the context of the activities of an establishment of a controller or a processor in the European Union, regardless of whether the processing takes place in the Union or not [18].

E. Forms of Consent Implementation

The implementation for the Forms of Consent is developed as two separate modules; a back-end and a front-end module. The back-end consists of a set of stateful Smart Contracts and is built on top of Ethereum, a Turing-complete smart contract platform. The front-end is a Dapp, a web application that runs on the user browser and interacts directly with the back-end on the Blockchain.

The back-end module is written in the Solidity programming language, a contract-oriented, high-level language for implementing Smart Contracts that targets the Ethereum Virtual Machine (EVM). It consists of a set of inter-connected Smart Contracts that implement the functionality of the Forms of Consent.

The system consists of three different types of contracts, "Platform", "Users" and "Support". The contracts, that are inter-connected via the use of *external* functions and through inheritance, are detailed below. A Class Diagram for the Forms of Consent implementation can be seen in Figure 2.

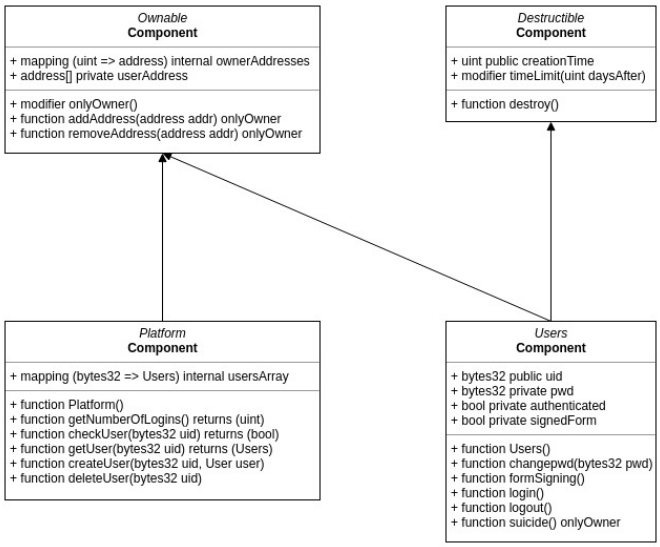

Fig. 2. Class Diagram of the Forms of Consent implementation

1) *Platform*: A single Smart Contract that acts as the central, administrative node of the Forms of Consent deployment. The contract is deployed first as it consists the node through which multiple users of the GHOST network are tracked. To store information on the users, the contract keeps a *mapping* array that contains pairs of user IDs *uid* and contracts *Users*. The following functions are available:
   - *Platform* is the constructor function of the Contract that adds the address of the real owner of the Contract to the *ownerAddresses* array inherited by the Ownable Contract.
   - *getNumberOfLogins* returns the number of currently logged in users.
   - *checkUser* returns whether a user with an ID *uid* is matched to a contract *Users*.
   - *getUser* returns the Blockchain address of the Contract *Users* matched to a user with an ID *uid*.
   - *createUser* is used to create a user by pairing an ID *uid* to the Blockchain address of a Contract *Users*. It is actually not called by the *Platform* Contract, but from the constructor of the *Users* Contract instead.
   - *deleteUser* removes the user with ID *uid* from the *usersArray* array. It is also call by the *Users* Contract instead.
2) *Users*: For every smart home of the GHOST network, a different Smart Contract of this type is created and connected to the "Platform" Smart Contract. The contract stores non-sensitive information of the smart home it is assigned to, such as the GHOST ID, a field that indicates whether the user has accepted the Forms of Consent as well as additional fields connected to

different functionality of the system including the list of smart devices deployed on the smart home and their firmware version.

After creating the "Platform" Smart Contract, one or more "Users" contracts can be deployed to the network by providing as an argument the ID *uid* of a user, their password *pwd* and the address of "Platform" on the Blockchain of the GHOST network. This information is stored through the Contract constructor to the relevant *uid* and *pwd* variables and subsequently the *createUser* function of the Platform Smart Contract is called to create an entry to the *usersArray* array using the newly created address of the "Users" Smart Contract. "Users" provides access to the following functions:

- *Users* is the constructor function of the Contract that calls the *createUser* function of the Platform Contract.
- *changepwd* is used to change the password of the user that owns the specific Users Contract.
- *formSigning* is used to set the *signedForm* variable when the user accepts the terms of the Form of Consent.
- *login* is used by the user to login to the Platform.
- *logout* is used by the user to logout from the Platform.
- *suicide* calls the *deleteUser* function of the Platform Contract and then using the *destroy* function inherited by the Destructible contract to remove the Contract from the Blockchain network.

3) *Support*: The support contracts, called "Ownable" and "Destructible" provide a set of utilities to both the "Platform" and "Users" contracts. "Ownable" is used to determine the owner of a contract and to subsequently restrict access to the "Platform" contract to just the administrator of the GHOST network and to each "Users" contract by the appropriate owner of a smart home installation. The "Destructible" contract on the other hand inherits to "Platform" and "Users" the functionality to remove a contract from the Blockchain after a specified time limit or whenever the administrator deems appropriate. The support contracts consist of the following functions:

- The *ownerAddresses* array can be used to store multiple Blockchain addresses that belong to the same user. It is manipulated with the use of the *addAddress* and *removeAddress* functions.
- The *destroy* function can be used to delete a Smart Contract from the Blockchain network.
- The *timeLimit* modifier can be used to automatically call the *destroy* function and thus delete a Contract, when a user-defined time period had elapsed.

The front-end Dapp module is in many ways similar to a standard HTML/CSS/Javascript web service that instead of a traditional database back-end it is running on top of the EVM. This interoperability between the back-end and the front-end is realized via the web3.js Ethereum JavaScript API.

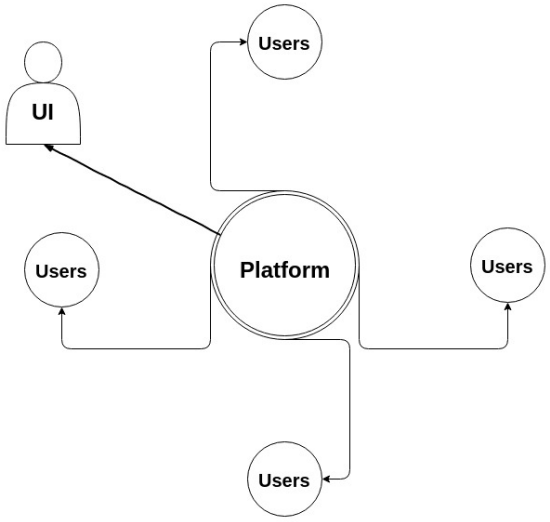

Fig. 3. Architecture of the Forms of Consent implementation

To read the terms of usage for the GHOST network and sign the Form of Consent, the user assigned to each smart home is presented with a simple web interface written using a combination of HTML, to describe the structure of the interface and CSS to style the interface.

The interface defines the following areas:

1) *Terms*: A text area that holds the terms that the user must read and accept.
2) *User information*: A table that shows user information retrieved from the Blockchain through the "Users" contract. This information includes the user's full name, their Blochchain ID as well as whether the latest version of the Forms of Consent has been already signed.
3) *Signature button*: The button which the user must press to indicate that they accept the terms of usage of the GHOST network.

Under the hood, the web interface uses Javascript and the jQuery library to parse the user's actions and to pass to the HTLM different information depending on whether the user has already signed the latest version of the consent form or their signature is pending. The actual connection to the Blockchain and the appropriate "Users" contract is performed with the use of the web3.js Ethereum JavaScript API. web3.js is implementing the JSON RPC API, a stateless, light-weight remote procedure call (RPC) protocol.

Events are a way for the back-end module to notify the front-end of a particular occurrence or action taken. When an event is triggered, a signal is emitted through the Web3 API that in turn can react if it was watching for that particular signal. The Forms of Consent implementation uses events extensively to promote the interoperability between the Smart Contracts and the graphical interface.

Figure 3 presents the architecture of the Forms of Consent implementation.

IV. Conclusions and Future Work

This paper presented the design and development of a Forms of Consent application for the GHOST network. The application is developed as a front-end Dapp module that leverages web technologies such as HTML/CSS and Javascript and uses the web3.js Ethereum JavaScript API to interconnect and communicate with a Smart Contract back-end deployed on a private Ethereum Blockchain network. It was demonstrated how the trust of smart home users can be promoted, by informing the users of GHOST for the principles and terms of the network.

Further work will include the implementation of additional applications for the GHOST network to promote security and user trust even further. These include a software integrity application that utilizes a new firmware update scheme, based on a synergy between the Blockchain network and a BitTorrent network to validate and update the firmware of IoT devices on smart homes, as well as an IP blocklisting application that involves the use of a Knowledge Base and an Risk Assessment module to flag IP addresses as malicious and subsequently block them.

Acknowledgment

This work is partially funded by the European Union's Horizon 2020 Research and Innovation Programme through GHOST project (https://www.ghost-iot.eu/) under Grant Agreement No. 740923

References

[1] PRWIRE, "Gartner says the internet of things will transform the data centre." [Online]. Available: https://prwire.com.au/pr/42679/gartner-says-the-internet-of-things-will-transform-the-data-centre

[2] Businessinsider, "How the internet of things will impact consumers, businesses, and governments in 2016 and beyond." [Online]. Available: http://www.businessinsider.com/how-the-internet-of-things-market-will-grow-2014-10

[3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] E. Vasilomanolakis, J. Daubert, M. Luthra, V. Gazis, A. Wiesmaier, and P. Kikiras, "On the security and privacy of internet of things architectures and systems," in *2015 International Workshop on Secure Internet of Things*, 2015, pp. 49–57.

[5] M. Conoscenti, A. Vetr, and J. C. D. Martin, "Blockchain for the internet of things: A systematic literature review," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, Nov 2016, pp. 1–6.

[6] N. Kshetri, "Can blockchain strengthen the internet of things?" *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.

[7] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2017, pp. 618–623.

[8] B. Lee and J.-H. Lee, "Blockchain-based secure firmware update for embedded devices in an internet of things environment," *The Journal of Supercomputing*, vol. 73, no. 3, pp. 1152–1167, 2017.

[9] K. Biswas and V. Muthukkumarasamy, "Securing smart cities using blockchain technology," in *18th International Conference on High Performance Computing and Communications; 14th International Conference on Smart City; 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2016, pp. 1392–1393.

[10] W. Lee, J.-H. Jin, and M.-J. Lee, "A block chain-based identity management service supporting robust identity recovery," 2016.

[11] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *Open and Big Data (OBD), International Conference on*. IEEE, 2016, pp. 25–30.

[12] M. A. Walker, A. Dubey, A. Laszka, and D. C. Schmidt, "Platibart: a platform for transactive iot blockchain applications with repeatable testing," in *Proceedings of the 4th Workshop on Middleware and Applications for the Internet of Things*. ACM, 2017, pp. 17–22.

[13] A. Collen, N. Nijdam, J. Augusto-Gonzalez, S. Katsikas, K. Giannoutakis, G. Spathoulas, E. Gelenbe, K. Votis, D. Tzovaras, N. Ghavami, M. Volkamer, P. Haller, A. Sanchez, and M. Dimas, "Ghost - safeguarding home iot environments with personalised real-time risk control," in *International Symposium on Computer and Information Sciences (ISCIS) Security Workshop*, 2018.

[14] D. Kraft, "Difficulty control for blockchain-based consensus systems," *Springer Peer-to-Peer Networking and Applications*, vol. 9, no. 2, pp. 397–413, 2016.

[15] L. Backlund, "A technical overview of distributed ledger technologies in the nordic capital market," Ph.D. dissertation, Uppsala University, 2016.

[16] N. Szabo, "The idea of smart contracts, 1997," *http://szabo. best. vwh. net/smart_contracts_idea. html*.

[17] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 254–269.

[18] EU, "Regulation (eu) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data." [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679