Towards an end-to-end IoT data privacy-preserving framework using blockchain technology

Faiza Loukil, Chirine Ghedira, Khouloud Boukadi, Benharkat Aïcha-Nabila

Towards an end-to-end IoT data privacy-preserving framework using blockchain technology

Faiza Loukil[1], Chirine Ghedira-Guegan[2], Khouloud Boukadi[3], and Aïcha Nabila Benharkat[4]

[1] University of Lyon, University Jean Moulin Lyon 3, CNRS, LIRIS, FRANCE
faiza.loukil@liris.cnrs.fr
[2] University of Lyon, University Jean Moulin Lyon 3, iaelyon school of Management, CNRS, LIRIS, FRANCE; chirine.ghedira-guegan@liris.cnrs.fr
[3] Mir@cl Laboratory, Sfax University, Tunisia
khouloud.boukadi@fsegs.usf.tn
[4] University of Lyon, INSALyon, CNRS, LIRIS, FRANCE
nabila.benharkat@liris.cnrs.fr

Abstract. Internet of Things-based environments collect and generate huge amounts of data about users, their activities, and their surroundings, which can disclose some sensitive information and threaten their privacy. Hence, the user's collected and handled data by IoT-based applications need to be exploited and secured in an appropriate way to protect personal data and user's privacy. Therefore, we aim at improving the data ownership, transparency, and auditability for users. To this end, we propose an end-to-end privacy-preserving framework for the IoT data using blockchain technology. The smart contract use in our framework will hence enforce the privacy requirement compliance according to the user's (i.e., data owner) privacy preferences and end-user's (i.e., data consumer) requests. To do so, we detail the design of the system architecture by introducing its core components and functionalities and highlight through an example of how it operates in a real-world use-case.

1 Introduction

Internet of Things (IoT) consists of devices that collect, exchange, store, and process large amount of fine-granularity and high-frequency data in every aspect of life. Such detailed data improve delivering advanced services in a wide range of application domains. Indeed, service providers gather the IoT data and use them to personalize services, optimize decision-making process, and predict future trends. However, the IoT data raise security and privacy concerns. In fact, the users have a little or no control over the collected data about themselves [10].

Furthermore, due to the distributed nature of the IoT networks, security and privacy are recognized to be among the major challenges of the IoT domain. Well-known security and privacy techniques, such as encryption, authentication, and role-based access control which are used in the context of conventional

information systems failed to protect IoT data due to the variety of hardware platforms and limited computing resources [1]. For instance, well-known encryption protocols and privacy-preserving methods, such as RSA, fully homomorphic encryption, and differential privacy proved to be very expensive when running on devices with limited computing capabilities in the IoT domain [16].

In recent years, the blockchain emerged as a new technology. The first system based on this technology was Bitcoin [12], which enables users to securely transfer the currency (bitcoins) while eliminating the need to trust a centralized regulator. Ethereum [3] is another blockchain-based system that can also be used for the cryptocurrency. Unlike Bitcoin, Ethereum has the ability to use a smart contract, which is a common agreement between two or more parties. It stores information, processes inputs and writes outputs thanks to its predefined functions [3]. Ethereum requires paying currency to run smart contract to prevent infinitely runs. Since then, other projects demonstrated how the blockchain technology could be used to address other domains, like the IoT data privacy-preserving. Although several blockchain-based solutions [2] [5] [7] [13] address the privacy issue in the IoT domain, they assumed that the IoT resources had sufficient resources to solve the Proof-Of-Work, which may not always be true as well as the others, did not address the whole IoT data lifecycle. Moreover, existing solutions did not consider all the privacy requirements, such as the purpose, retention duration, disclosure limitation, etc. that are defined by the privacy standard [8] and legislation [15] to preserve user privacy. In our work, the privacy requirements should cover the obligations that must be fulfilled by all the involved parties to preserve the privacy during the whole IoT data lifecycle.

Motivated by the legal rights imposed by the European General Data Protection Regulation (GDPR) [15], we focus on the privacy requirement enforcement to preserve privacy during the whole IoT data lifecycle. The objective of our work is to guarantee that the privacy requirements will be enforced while handling the shared IoT data that are collected by IoT resources. To this end, we propose an end-to-end privacy-preserving framework for the IoT based on the blockchain technology and more specifically on smart contract. The main purpose of the smart contract use is to enable our framework to express privacy-preserving policies. A policy is a set of conditions that the consumer needs to fulfill in order to handle a specific shared IoT data. Thus, the use of smart contract will prevent any privacy violation attempts. By protecting the shared data, these data can be only handled by invoking functions defined on the hosted smart contract in the blockchain. This implies that there are no parties that can get hold of the smart contract once hosted in the blockchain. Thus, smart contract enforces the data owner's privacy preferences, then the shared data will be handled as expected.

This paper is organized as follows. Section 2 discusses the existing researchers who studied blockchain and IoT technology integration to preserve privacy. Section 3 presents an overview of our proposed framework. Section 4 identifies its core components. Section 5 explains the framework's main functionalities. Section 6 validates our solution in a healthcare scenario. Section 7 concludes the paper and presents some future endeavors.

2 Related Work

IoT and blockchain combination generates resilient, peer-to-peer systems, and the ability to interact with peers in a trustless and auditable manner [4]. Many researchers have studied the integration of blockchain and IoT technology.

Biswas and Muthukkumarasamy [2] for example proposed a blockchain based security framework to enable secure data communication in a smart city. However, their proposal is at a high level of abstraction and they do not provide any system design to prove the feasibility of their framework. On the other hand, Dorri et al. [5] proposed a lightweight and optimized blockchain for resource-constrained devices and applied it in a smart house scenario. This work focused on data store and access use cases by IoT resources. However, the system design included a centralized control node, which can be considered as a single point of failure that could damage availability. For their part, Hashemi et al. [7] proposed a decentralized solution for the sharing of data in the IoT environment, which consists of a distributed data storage system. However, the authors assumed that the IoT have sufficient resources to solve the Proof-Of-Work which may not always be true. In fact, solving the POW for Bitcoin requires a very sophisticated hardware. Ouaddah et al. [13] proposed FairAccess, a blockchain-based framework for access control. The authors relied on Bitcoin system and introduced new transaction types. The transactions are used to provide access control, and the blockchain is used to store and read the permissions. However, this work addressed the access control issue but did not address the whole data lifecycle.

To sum up, it can be said that most existing blockchain-based solutions concentrate on addressing the access control issue in the IoT field. Moreover, they are only concerned with one phase, and do not address the whole data lifecycle.

3 Proposed Solution Overview

Considering the legal rights imposed by the GDPR [15], it is necessary to ensure the privacy requirement compliance to preserve privacy during the whole data lifecycle, covering the collection, transmission, storage and processing phases. In our work, we focus on how to enforce these privacy requirements and obligations for the IoT environment. To this end, we propose PrivBlockchain, an end-to-end privacy-preserving framework for the IoT data. PrivBlockchain is based on these principles, namely (i) user-driven and transparency, (ii) distributed architecture and central authority lack, and (iii) fine-granularity privacy policies.

Figure 1 depicts the proposed architecture, which includes two types of network, namely private IoT network, which can be a smart home, smart building, etc. This network includes the IoT resources owned by a data owner, which can be an individual or an organization. The second network is the public IoT network, which represents the external domain of the private IoT network. Moreover, we distinguish three types of IoT network nodes, namely private, public, and storage nodes. Both public and storage IoT network nodes belong to the public network. The private node (i.e., gateway node or private IoT resource) is an IoT node that belongs to both the private and public IoT networks.

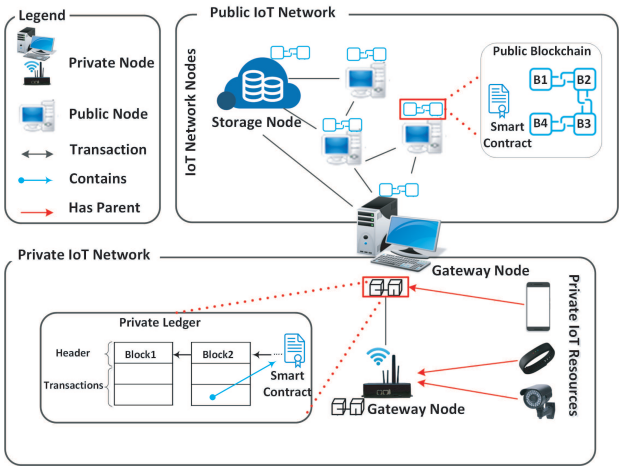

Fig. 1. PrivBlockchain Architecture

In the private IoT network, each data owner has one or more high resource devices, known as the "gateway node", which is responsible for the other owned IoT resources. The communication between the owned IoT resources by the data owner (i.e., the private IoT resources) is stored in a private blockchain called the "private ledger". The communication between the private IoT nodes and the other nodes of the public IoT network is stored in a "public blockchain".

The reason behind using the blockchain technology to preserve privacy in the IoT domain is that the blockchain is an immutable public record of data secured by a network of peer-to-peer participants that use addresses as pseudonyms. Such user identity management improves anonymity and pseudonymity in an IoT network. In our case, we use a pair of public and private keys to ensure the anonymity and pseudonymity privacy properties. Moreover, using a different key pair in each transaction enforces unlinkability. For instance, the gateway node can use a different key pair in each transaction with the external nodes that belong to the public IoT network. Besides the privacy properties, namely anonymity, pseudonymity, and unlinkability[14] that are ensured by the blockchain, this final has the potential to enforce privacy requirement compliance. To this end, a set of privacy requirements is chosen that we agree critical in the area of IoT based on an extensive literature review, the ISO standard [8], and the GDPR [15].

We outline the proposed framework core components in the following section.

4 PrivBlockchain Core components

Table 1 shows the main blockchain-based solution components. Indeed, the PrivBlockchain framework consists of nine core components, such as: smart contracts, transactions, private IoT networks, private ledgers, gateway nodes, local storage, public IoT networks, public blockchain, and storage nodes.

Table 1. PrivBlockchain core components description

Component	Component description
Smart contract	It is a common agreement that is hosted within the blockchain. We propose three smart contracts, namely *PrivacyPermissionSetting*, *Ownership*, and *SubscriptionPrivacyPolicy*. The two first smart contracts are published in the private ledger. The third one is published in the public blockchain.
Transaction	Communication between IoT resources and network nodes is known as a transaction. We define a set of transaction types, namely T_{Add}, T_{Remove}, $T_{LocalStore}$, T_{Store}, T_{Access}, $T_{Monitor}$, $T_{GetPermission}$, $T_{GrantPermission}$, and $T_{GetSharedResource}$.
Private IoT network	It is an area, like a smart home or a smart building, where its owner can control a set of private IoT resources.
Private ledger	It is a local private blockchain that enables the data owner to control his own IoT resources.
Gateway node	It is a device with a high memory and storage capabilities. Each gateway node is responsible for a set of private IoT resources, generates there keys and adds them to the IoT network.
Local storage	It is a device used to store data locally. It saves the collected data by IoT resources for long-term storage before sending them to the external storage center, which is the storage node.
Public IoT network	It is a peer-to-peer network that contains several nodes with different memory and storage capabilities.
Public blockchain	It can be seen as the history of all the transactions that are sent by the public nodes to access or share IoT data in the public IoT network. In fact, it can ensure auditing functions.
Storage node	It is a public IoT network node that offers a storing service for both public blockchain and data collected by the IoT resources.

5 PrivBlockchain functionalities

Based on the proposed smart contracts, our end-to-end privacy-preserving framework includes the following functionalities: (i) adding a new IoT resource to the *Ownership* smart contract, (ii) storing the collected data by a private IoT resource, and (iii) sharing IoT resource output with data consumers. The dynamic aspect of PrivBlockchain relies on those functionalities that we detail hereafter:

5.1 IoT resource add transaction protocol

Each gateway node publishes an *Ownership* smart contract that includes its own IoT resource addresses in the private ledger. For each IoT resource, a set of outputs can be added. A *PrivacyPermissionSetting* smart contract is associated with each IoT resource output. This smart contract enforces the data owner's privacy preferences about how the owned IoT resources must behave.
Figure 2 depicts the business process of adding a new IoT resource. We assume that the gateway node has created its *Ownership* smart contract and generated a pair of keys for an IoT resource.

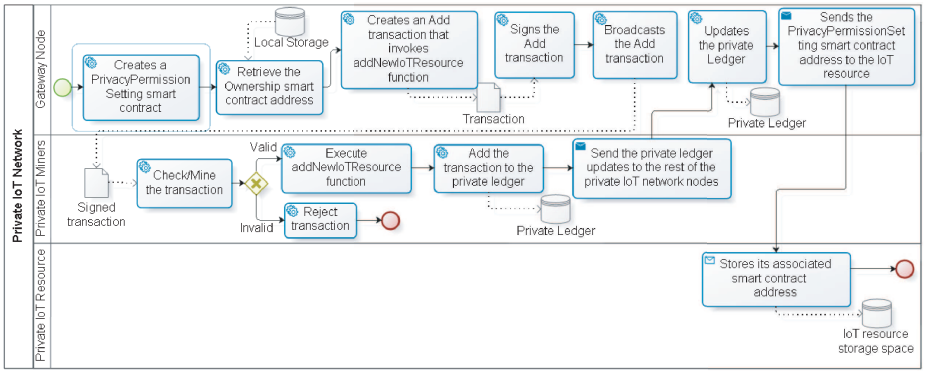

Fig. 2. Adding a new IoT resource business process notated in BPMN

5.2 IoT resource output store transaction protocol

IoT resource periodically sends a $T_{LocalStore}$ transaction to the gateway node in order to locally store the collected data.
We assume that the IoT resource has an address and it knows the address of its associated smart contract and the gateway node address.
1. The IoT resource calculates the data hash, encrypts the collected data using the lightweight AES Encryption [9], and sends the encrypted data to the gateway node through a secure channel.
2. The gateway node decrypts the received data and calculates the data hash.
3. The IoT resource creates a new $T_{LocalStore}$ transaction that invokes the LocalStore function of its smart contract. The transaction includes the data type and hash. The $T_{LocalStore}$ transaction is sent to the appropriate gateway node because only the parent has the appropriate private key to sign the transaction.
4. When the gateway receives the transaction, it verifies the data integrity by comparing the data hash stored on the transaction and the calculated data hash in Step 2. In case of a match, the gateway signs the transaction and broadcasts it to the private IoT network. Otherwise, the transaction and data are rejected.
5. The miners validate and then execute the appropriate LocalStore function, which enables the IoT resource permission verification. The included data type on the transaction must belong to the allowed IoT resource outputs. If it belongs, the associated *PrivacyPermissionSetting* smart contract address of the IoT resource output is retrieved and compared with the included smart contract address on the transaction. If both addresses are the same and the storage permission is Permit, then the transaction is added to the ledger and the gateway node stores the data on the local storage. Otherwise, the received data are rejected.

5.3 Process of sharing IoT resource output

The consumer's subscription establishes a relationship between one node and an IoT resource output.

1. The consumer creates a subscription request, which contains the requested data, why, when, where, how, to whom, and for how long the data are needed. This request is sent to the data owner address in a $T_{GetPermission}$ transaction.
2. The Matching Manager is included in the gateway node and matches between the data owner's privacy preferences and the consumer's subscription request. First, the Matching Manager evaluates the privacy requirements of the output privacy rule and the consumer's subscription request. In case of a match, the Matching Manager verifies if there is an already published file with the same requested data to retrieve the associated *SubscriptionPrivacyPolicy* smart contract address. Otherwise, it creates a new file that contains the result of the requested data. Then, it invokes the constructor of the *SubscriptionPrivacyPolicy* smart contract to create a new one. It specifies the hash of the file content and the address of the node that stores the shared file. After that, it sends the transaction to the gateway node to be signed and propagated to the network. Once the smart contract is created, the Matching Manager receives its address.
3. Once the Matching Manager gets the appropriate *SubscriptionPrivacyPolicy* smart contract address, it creates a $T_{GrantPermission}$ transaction that invokes the addConsumer function of this contract with the appropriate privacy permissions. It sends the transaction to the gateway node to be signed and propagated to the network. This transaction enables to add a new consumer to the list of the allowed consumers of the shared file.
4. When the data consumer receives the *SubscriptionPrivacyPolicy* smart contract address, it can send a $T_{GetSharedResource}$ transaction that invokes a set of the smart contract functions to handle the data. Before executing each function, the set of the consumer's permissions is verified to enforce the data owner's privacy preferences. For instance, if the consumer has a permission to disclose the file content and the retention duration is not finished yet, it can invoke the addConsumer function in order to add a new consumer to the file but with read-only permission and limited retention duration. We refer the reader to [1] for further details about the Business Process Models of the defined protocols in sections 5.1, 5.2, and 5.3.

6 Prototype and validation

We implemented our proposed smart contracts using the Solidity language [6] and deployed it to the Ethereum test network, which is based on the go ethereum (geth) implementation of the protocol released by the Ethereum foundation. The test network is identical to the production network except that the Ether has no real-world value. Given that our system does not rely on the currency transfer, the test network works like the real Ethereum network. MyEtherWallet [11] is used to access the network node information and control the public IoT network. Because synchronizing with the entire Ethereum test network would take too much time and resources, we decided to mount a local private Ethereum blockchain.

[1] https://sites.google.com/view/privblockchain-protocols

Insofar as PrivBlockchain is generic and could be used for a variety of IoT application domains, such as smart home, smart grid, etc., we applied it to the following scenario from the healthcare domain to validate our solution:

A patient named Bob needs to follow a healthcare protocol, which consists in practicing some sport activities and eating healthy meals. Bob owns a private IoT resource, which is a wearable device that collects the user's heart rate. This IoT resource can be connected to several IoT resources, such as smart equipment. Bob goes regularly to a modern gym, which uses fitness smart equipment. This smart equipment collects data from the different users' wearable devices and sends them to the gym data center to be stored. These stored data are analyzed to propose personalized recommendations for users, detect the most used equipment, propose group training programs, etc. Moreover, the gym offers a "healthy eat application" that proposes a set of healthy meals according to the needed calories for each specific user. Bob wants to connect his wearable device to a smart treadmill in the gym to monitor his performance by consistently measuring his heart rate and walking or running power. Hence, a need for a break or the water notifications could be sent to Bob when necessary. However, Bob is afraid that the gym center uses the collected data by his wearable device to monitor his activities or to disclosure them to third parties without his consent.

In order to reassure its clients, the gym center needs to opt for a solution that preserves the IoT data privacy and improves transparency. Indeed, the gym center relies on the PrivBlockchain solution in order to (i) gain the users' trust, (ii) guarantee the users' right to control their personal data while benefiting from personalized services, and (iii) be compliant with the privacy legislation [15].

Figure 3 shows the major components and the interactions between them. It is worth noting that all the components are identified by Ethereum addresses and the interactions are shown as transactions in Figure 4.

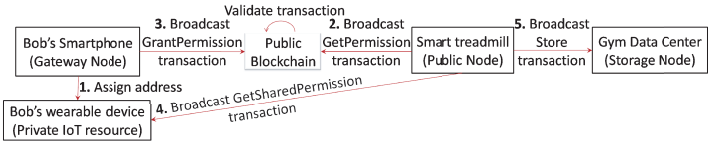

Fig. 3. Implementation components

Figure 4 depicts both adding an IoT resource and sharing its output protocols. Transactions shown in the middle succeed to add Bob's wearable device and the smart treadmill as a heart rate consumer. The figure left side details the addNewIoTResource function that is invoked by the gateway node to add Bob's wearable device and associate a *PrivacyPermissionSetting* smart contract to the output. The right side details the addConsumer function, which is defined in the *SubscriptionPrivacyPolicy* and invoked by a transaction sent by the smart treadmill. This transaction is validated and added to the public blockchain because the smart treadmill has the disclosure permission, otherwise, it is rejected.

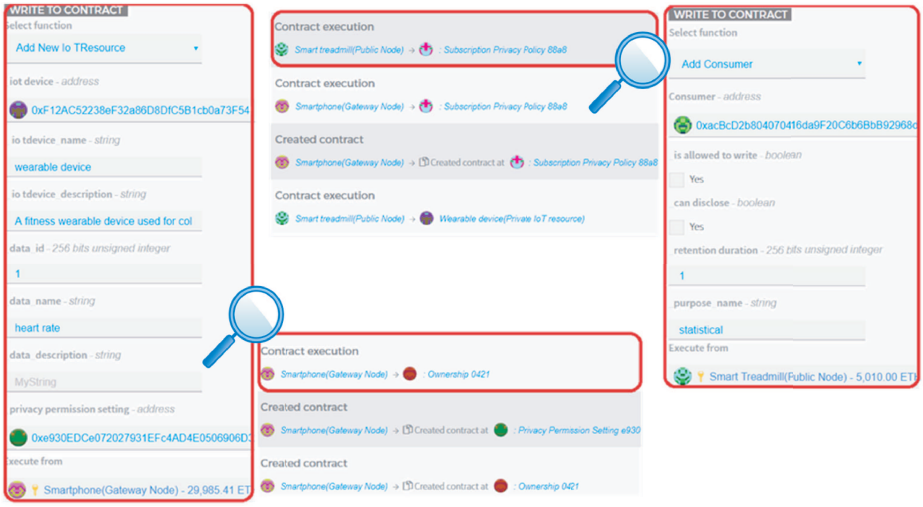

Fig. 4. Illustration of addNewIoTResource and addConsumer function invocations

7 Conclusion

In recent years, several researchers have agreed that the combination of blockchain and IoT generates resilient, truly distributed peer-to-peer systems and the ability to interact with peers in a trustless and auditable manner. However, few proposed solutions have dealt with the privacy issue in the IoT domain. Despite the increasing legislation pressure, the privacy requirements, such as consent and choice, purpose specification, and collection limitation, have been less addressed in the IoT domain. For these reasons, we have focused on the privacy requirement enforcement to preserve privacy during the whole IoT data lifecycle using the blockchain technology. To this end, we have proposed an end-to-end privacy-preserving framework for the IoT data based on smart contracts to enforce preserving privacy in the IoT domain. As 'proof of concept', we have established an initial implementation of PrivBlockchain with limited number of nodes. In our future work, we intend to experiment our framework in a large IoT network with multiple involved parties. We plain also to broadcast a large amount of transactions that ask and grant permissions, then analyze the PrivBlockchain's behavior over time. Moreover, we intend to perform comparative performance analysis by comparing the estimated computational costs of PrivBlockchain and some of the existing works using the *gas* as a unit of measure.

Blockchain analysis can possibly reveal the frequency of visiting a place or practicing an activity by a specific node. To overcome this problem, our framework enables the several addresses use for the same IoT resource. Besides, we intend to incorporate the use of differential privacy, a rigorous privacy model that preserves data privacy while maintaining utility in our framework. In fact, by adding some noise to the transactions, we can prevent blockchain analysis.

References

1. Bertino, E.: Data security and privacy in the iot. In: EDBT. vol. 2016, pp. 1–3 (2016)
2. Biswas, K., Muthukkumarasamy, V.: Securing smart cities using blockchain technology. In: High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on. pp. 1392–1393. IEEE (2016)
3. Buterin, V., et al.: A next-generation smart contract and decentralized application platform. white paper (2014)
4. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. IEEE Access **4**, 2292–2303 (2016)
5. Dorri, A., Kanhere, S.S., Jurdak, R.: Towards an optimized blockchain for iot. In: Proceedings of the Second International Conference on Internet-of-Things Design and Implementation. pp. 173–178. ACM (2017)
6. Ethereum project's Solidity Team: Solidity language. http://solidity.readthedocs.io/en/develop/ (2016), last accessed 10 Aug 2018
7. Hashemi, S.H., Faghri, F., Rausch, P., Campbell, R.H.: World of empowered iot users. In: Internet-of-Things Design and Implementation (IoTDI), 2016 IEEE First International Conference on. pp. 13–24. IEEE (2016)
8. International Organization for Standardization: Information technology security techniques privacy framework, ISO/IEC 29100 (2011)
9. Landman, D.: Arduino Library for AES Encryption. https://github.com/DavyLandman/AESLib (2017), last accessed 10 Aug 2018
10. Maddox, T.: The dark side of wearables: How they're secretly jeopardizing your security and privacy. https://www.techrepublic.com/article/the-dark-side-of-wearables-how-theyre-secretly-jeopardizing-your-security-and-privacy/ (2015), last accessed 10 Aug 2018
11. MyEtherWallet Team: Myetherwallet. https://www.myetherwallet.com/ (2015), last accessed 10 Aug 2018
12. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
13. Ouaddah, A., Abou Elkalam, A., Ait Ouahman, A.: Fairaccess: a new blockchain-based access control framework for the internet of things. Security and Communication Networks **9**(18), 5943–5964 (2016)
14. Pfitzmann, A., Hansen, M.: Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology (2005)
15. Regulation, General Data Protection: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46. Official Journal of the European Union (OJ) **59**, 1–88 (2016)
16. Singla, A., Mudgerikar, A., Papapanagiotou, I., Yavuz, A.A.: Haa: Hardware-accelerated authentication for internet of things in mission critical vehicular networks. In: Military Communications Conference, MILCOM 2015-2015 IEEE. pp. 1298–1304. IEEE (2015)