PATRIoT: A Data Sharing Platform for IoT Using a Service-Oriented Approach Based on Blockchain

Faiza Loukil[1(✉)], Chirine Ghedira-Guegan[2], and Aïcha-Nabila Benharkat[3]

[1] University of Lyon, University Jean Moulin Lyon 3, CNRS, LIRIS, Lyon, France
faiza.loukil@liris.cnrs.fr
[2] iaelyon School of Management, University of Lyon, University Jean Moulin Lyon 3, CNRS, LIRIS, Lyon, France
chirine.ghedira-guegan@univ-lyon3.fr
[3] University of Lyon, INSALyon, CNRS, LIRIS, Lyon, France
nabila.benharkat@insa-lyon.fr

Abstract. The Internet of Things has emerged as a paradigm in a variety of application domains where several parties share data to tackle specific tasks. However, these IoT data can be sensitive and the data subject wish not share them with other competitor organizations without retaining some level of control. Thus, a privacy-preserving, user-centric, and transparent solution is needed to deal with the challenges of IoT data sharing, such as the loss of control over the shared data, the trust need in data consumer infrastructure, and the lack of transparency in terms of data handling. Therefore, we propose PATRIoT, a privacy-preserving PlATfoRm for IoT data sharing using a service-oriented approach. The latter is proposed based on the blockchain technology, which enforces privacy requirement compliance according to the General Data Protection Regulation. For validation purposes, we deploy the proposed solution on a private Ethereum blockchain and give the performance evaluation.

Keywords: Privacy · IoT · Blockchain technology · Data sharing

1 Introduction

The Internet of Things (IoT) is a paradigm that improves delivering advanced services in a wide range of application domains. Indeed, multiple devices collect, exchange, store, and process a large amount of fine-granularity and high-frequency data in every aspect of life [4]. However, these smart devices have limited memory and storage capabilities, so they collect and send IoT data to consumers' external platforms to be stored and analyzed. Indeed, these external platforms receive these IoT data and use them to personalize services, optimize decision-making processes, and predict future trends. However, the produced IoT data are generally rich in sensitive data and their analysis allows data consumers

E. Kafeza et al. (Eds.): ICSOC 2020, LNCS 12571, pp. 121–129, 2020.
https://doi.org/10.1007/978-3-030-65310-1_10

to deduce personal behaviors, habits and preferences of data subjects.[1] Indeed, collecting data in IoT applications increases the data subject's worries about the potential uses of these data. Hence, centralizing the storage and analysis of a huge amount of data poses significant issues in terms of data subject privacy, such as the loss of control over the externalized data, the need to trust the consumer platforms, and the lack of data handling transparency. To overcome the aforementioned issues, various legislative bodies have enacted privacy legislation, such as the General Data Protection Regulation (GDPR) [9] in Europe that gives data subjects rights to be informed how their personal information are handled by consumers. However, a user-centric and transparent solution for ensuring that these rights are respected in the IoT domain is still missing.

Motivated by the limited computing capabilities of smart devices, the sensitive feature of IoT data, and the increasing privacy legislation pressure, we propose PATRIoT, a preserving privacy PlATfoRm for IoT data sharing while adapting a service-oriented approach based on the blockchain technology. PATRIoT provides a set of services, generic enough to be applied to a large variety of IoT applications. These services can be deployed over a given architecture to make applications aware of users' privacy requirements, such as data purpose, disclosure, and retention. The components of PATRIoT are built around the semantic description of data (e.g., data sensitivity level, data purpose, etc.) without storing personal data. Furthermore, the reason behind the blockchain technology use is its immutable nature secured by a peer-to-peer network. It hosts smart contracts which contain conditions to trigger and actions to execute if the conditions are satisfied. In our case, the conditions represent the preferences and requirements of the data subjects concerning their IoT data privacy that need to be respected by consumers. Thus, the smart contract use prevents any attempt to violate privacy by ensuring that the shared data are handled as expected during their lifecycle.

This paper is organized as follows. Section 2 analyses existing solutions that studied the privacy-preserving issue in the IoT domain. Section 3 describes the proposed system model. Experiments and results are detailed in Sect. 4. Finally, Sect. 5 concludes the paper and presents some future endeavors.

2 Related Work

Early attempts to incorporate blockchain technology into IoT proposed new blockchain systems. For instance, Dorri et al. [8] proposed a custom blockchain, where the home gateways hold the role of the miners. Such a solution is hard to be deployed since they require a "critical mass". As it seems relevant to new IoT solutions, it is worth building on existing technologies to be compatible with already available libraries and wallets. More recent attempts are using blockchain and smart contracts to provide security and access control for IoT. Novo [11] proposed an IoT access control system with gateway nodes, which are responsible for handling resource requests by taking into consideration the policies stored in the

[1] Data subject: any person whose personal data are being collected, held or processed.

blockchain. For their part, Zhang et al. [12] proposed a smart contract-based access control system while an IoT gateway handles resource requests. These solutions encoded statically in smart contracts the actions a specific consumer can perform to a particular IoT device/data. Furthermore, the blockchain technology is also used in the healthcare field. For instance, Dagher et al. [7] proposed a blockchain-based framework for secure access to medical records by several parties, while preserving the patients' privacy. However, this work cannot perform data erasure, since it stored some personal data in the blockchain. To overcome this issue, an off-chain distributed database can be used to store the shared IoT data to guarantee data subjects' right to be forgotten as required by the GDPR.

3 System Model Overview

Despite the increasing legislation pressure, several privacy requirements have been less addressed in the IoT domain. Using a service-oriented approach to address the GDPR compliance makes it easier to the data consumers to build new applications or change existing applications while ensuring the enforcement of the data subject privacy thanks to smart contracts. For this purpose, we propose the system model that is depicted in Fig. 1. It consists of five involved parties, namely data producer, blockchain and smart contract, PATRIoT platform, distributed database, and data consumer. The PATRIoT platform aims at providing an environment that allows data subjects to easily exercise their rights defined by GDPR and assisting data consumers to meet the GDPR requirements using a privacy ontology and the blockchain technology. To this end, PATRIoT has been designed to be an IoT data sharing platform while adapting a service-oriented approach that provides several components including the privacy preference matching service and the privacy policy compliance service.

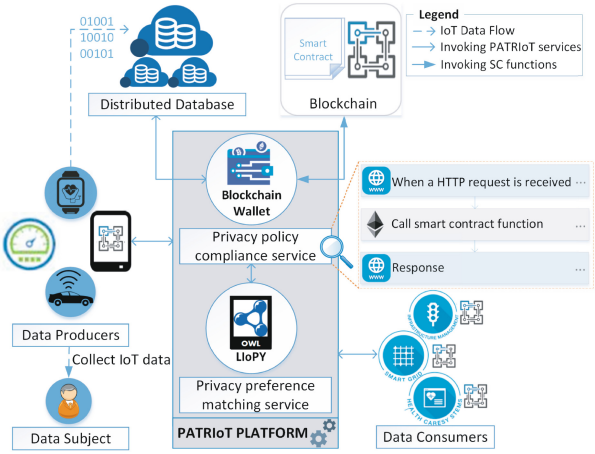


Fig. 1. System model

In the rest of this section, we first outline the proposed model core components, then describe the IoT data sharing process.

3.1 Core Components Description

Figure 1 depicts the model components, which we describe hereafter:

Data producer: it is an IoT device equipped with sensing and communication capabilities that allow it to collect data, communicate with other devices, or connect to the Internet. In this work, a mobile phone can provide a user-friendly environment for data subjects in order to control their shared data and manage their privacy preferences thanks to the privacy preference matching service.

Privacy preference matching service: it is responsible for matching the data subjects' and data consumers' privacy requirements that are served as inputs, then generating a common privacy policy, as an output. This privacy policy consists of several rules that specify why, when, how, to whom and for how long the requested IoT data are handled. To ensure privacy preference matching, the PATRIoT platform adopted the data privacy ontology, called LIoPY and the reasoning process that we have previously proposed in [10]. Indeed, LIoPY ontology models the privacy requirements in the IoT environment and the common privacy policy that will be enforced by the blockchain and smart contracts.

Blockchain and smart contract: blockchain is responsible for transparency, integrity, non-repudiation, and validity of the data handling operations. Moreover, it hosts a privacy policy as a set of self-enforcing and machine-readable rules using smart contracts [6]. Therefore, we propose IoTDataSharing, a smart contract that aims at addressing the data subject's control enforcement over the shared data and assisting the data consumers to meet the fundamental GDPR requirements. The predefined smart contract's functions can be invoked by the data producers and consumers by means of the privacy policy compliance service.

Privacy policy compliance service: it is responsible for exposing the functionality of the deployed smart contract as an application REST interface for simpler external application interaction with the blockchain. Both POST and GET methods are provided to push transactions and query for transactions on the blockchain. These methods can invoke the smart contract's functions. Indeed, this service ensures that the data sharing management works properly according to the access authorizations defined in the smart contract while eliminating the data producers' needs to interact directly with the blockchain due to their limited memory and storage capabilities. Moreover, it verifies the consumer's permissions before allowing access to the shared data that are stored in distributed databases.

Distributed database: it is an off-chain database, used to store the IoT data. It is a peer-to-peer storage system used to overcome the expensive cost of storing IoT data on the blockchain and to guarantee the right to be forgotten defined by the GDPR. Thus, only the hash pointer of the data location is exchanged by the blockchain's transactions between the data producers and the data consumers.

Data consumer: it can be a medical application, an energy substation, or a traffic routing station that can use the privacy preference matching service

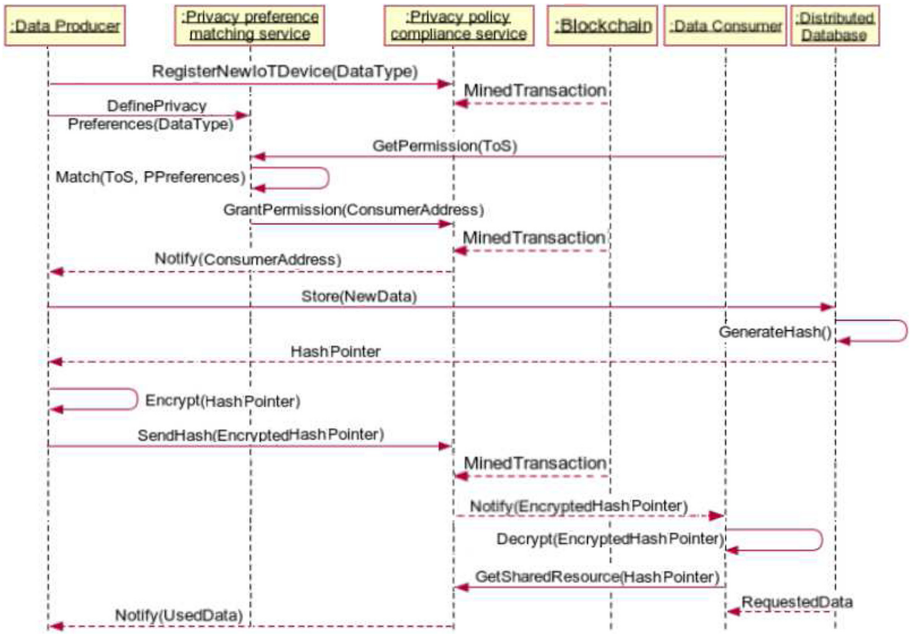


Fig. 2. The process for IoT data sharing using the PATRIoT platform. Assume that actors have established a blockchain address prior to this process.

to request data subjects' consents and the privacy policy compliance service to handle the requested data transparently and unambiguously.

For more details, we refer the reader to a full description that is available on.[2]

3.2 Blockchain-Based IoT Data Sharing Process

Figure 2 depicts the process of sharing IoT data between data producers and consumers using the PATRIoT platform while logging the established communication on the blockchain. This process begins by registering a new data producer using the RegisterNewIoTDevice function to store the producer's blockchain address and its sensed data type on the IoTDataSharing smart contract. Once the privacy preferences are defined, the data consumer asks for getting permission by specifying its terms of service, such as the requested data type, why, to whom, and for how long the data are used. Then, the privacy preference matching service matches the received terms of service with the data subject's privacy preferences, off-chain. In case of a match, the service uses the GrantPermission function to add the consumer's blockchain address to the authorized consumers' list stored in the smart contract that notifies the producer of the new consumer. When the data producer collects new data, it sends them to the distributed database that generates a hash pointer of the file location and

[2] https://www.dropbox.com/s/levxfzid1s3o50b/PATRIoT.pdf?dl=0.

returns it to the producer. The latter encrypts the received hash pointer using the consumer's public key and sends it to the consumer. Once mined, the consumer receives the transaction, retrieves the encrypted hash pointer, and uses its private key to decrypt the hash pointer. When the consumer obtains the hash pointer of the file location, it uses the GetSharedResource function provided by the privacy policy compliance service to retrieve the data from the distributed database or invoke one of the IoTDataSharing smart contract functions to handle the data. By using the privacy policy compliance service, the data subject is periodically informed how the data are handled and can easily add or revoke authorization to the data consumers.

4 Experiments and Result Analysis

Due to a lack of space, we only show in this section the proposal feasibility by implementing smart contracts, but the entire proposal is designed for a service-oriented architecture deployment. As Ethereum is currently the most common blockchain platform for the development of smart contracts [6], we implemented our smart contract using the Solidity language [1] and deployed it to the Ethereum test network using Ganache [2]. Therefore, we created a test system using Truffle development framework [3], used InterPlanetary File System (IPFS) [5] as an off-chain distributed database, and deployed the PATRIoT services to Swarm that is a Docker orchestration tool.

4.1 Computation Time Cost

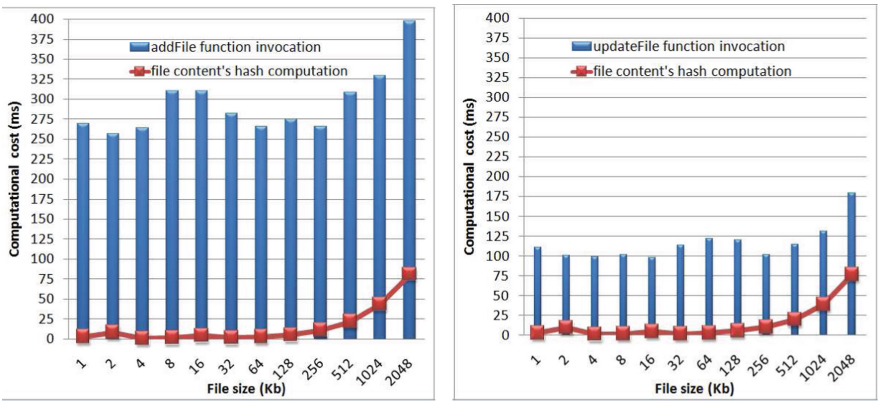
In order to measure the performance of our solution, we conducted some experiments to compute the computational time cost of both addFile and updateFile functions defined on the IoTDataSharing smart contract. Thus, we performed add file operation by adding random file contents for 100 repetitions. We measured the required time to off-chain compute the file content's hash and execute the addFile function (see Fig. 3a) and the updateFile function (see Fig. 3b) by making several tests while increasing the file size from 1KB to 2MB. We observe that the processing time of updating an existing file that varies from 95 to 180 ms is less than the processing time of adding a new file that varies from 257 to 390 ms.

4.2 Cost Overhead

To evaluate the PATRIoT efficiency, we conducted an experiment to measure the *gas*[3] used by a transaction to invoke one of the IoTDataSharing smart contract's functions, namely addFile, updateFile, addConsumer, and removeConsumer.

Table 1 illustrates the average gas usage and cost per invoked function. We observe that the *gas* used by a transaction changed depending on the function. This can be explained by the fact that functions that require more computational resources cost more *gas* than functions that require few computational resources.

[3] gas: it is a measure unit of the cost necessary to perform a transaction on the network.



(a) **addFile** function (b) **updateFile** function

Fig. 3. Average processing time of two functions with different file size

Table 1. Cost overhead

Invoked function	Average *Gas* Usage (*gas*)		Average *Gas* Cost (USD)
	File size = 1KB	File size = 2MB	
addFile	471959	471959	3,00
updateFile	28946	28946	0,18
addConsumer	332429	332429	2,11
removeConsumer	23456	23456	0,15

Moreover, we used in this experiment two file sizes, namely 1KB and 2MB. We deduce that the *gas* used by the transactions is independent of the file size. This can be explained by the fact that the functions only used the file content's hash whose bit length is fixed and equal to 32 bits. Thus, the file size has no impact on the cost overhead of our proposal. Indeed, this latter can be used in case of files with a huge amount of data without increasing the cost overhead.

Table 1 also illustrates the average gas cost of the four smart contract's functions. Currently, 1 *gas* costs about 20 Gwei (i.e., $20 * 10^{-9}$ Ether) and the exchange rate is about 318 USD for 1 Ether at the time of writing. Thus, we compute the *gas* cost by multiplying the used *gas* by the *gas* price for each function. Therefore, we can deduce that our solution is not a cost-expensive one.

After evaluating the performance, we analyze below the legal compliance.

4.3 PATRIoT Platform in Legislation Context: GDPR Compliance

PATRIoT aims at achieving the GDPR compliance by meeting several privacy requirements. First, PATRIoT meets the consent requirement by using the IoTDataSharing smart contract that offers to the data subjects the ability to manage

their consents by easily adding, modifying, and revoking authorizations. Moreover, PATRIoT establishes accountability and transparency of data sharing process. Thus, the defined blockchain-based solution helps data consumers to automate compliance checks and allows for a comprehensible record for auditing. Furthermore, PATRIoT meets the notification obligation by logging all the transactions that prove who has handled data. Thus, any privacy violation attempts can be detected. Finally, PATRIoT meets the erasure requirement by using the off-chain data store and only storing the hash of the IoT data on the blockchain.

By meeting the aforementioned privacy requirements, PATRIoT addresses areas associated with GDPR compliance. On one hand, it enforces the data subject's ownership and control over the shared data. On the other hand, it can be seen as a consumer's proof of legislation compliance thanks to both transparency and auditability characteristics.

5 Conclusion

In recent years, several researchers have agreed that the blockchain technology can be used to improve the data subject privacy in the IoT domain while being GDPR compliant. In this context, we proposed PATRIoT, an IoT data sharing platform for preserving privacy using a service-oriented approach based on blockchain. Indeed, we proposed a smart contract that ensures that the shared data will be handled as expected. Besides, we relied on off-chain database use to store the shared IoT data. In our future work, we aim at continuing research in the use of blockchain to meet other privacy legislative standards by deploying a Blockchain as a Service to be available for all actors in different domains.

References

1. Solidity language (2014). https://solidity.readthedocs.io/en/develop/. Accessed 20 Aug 2020
2. Ganache: Personal blockchain for ethereum development (2016). https://www.trufflesuite.com/ganache. Accessed 20 Aug 2020
3. Truffle: Ethereum development framework (2016). https://github.com/trufflesuite/truffle. Accessed 20 Aug 2020
4. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
5. Benet, J.: IPFS-content addressed, versioned, P2P file system (2014)
6. Buterin, V., et al.: A next-generation smart contract and decentralized application platform. White paper (2014)
7. Dagher, G.G., Mohler, J., Milojkovic, M., Marella, P.B.: Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustain. Cities Soc. **39**, 283–297 (2018)
8. Dorri, A., Kanhere, S.S., Jurdak, R., Gauravaram, P.: Blockchain for IoT security and privacy: the case study of a smart home, pp. 618–623 (2017)

9. GDPR: 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. Off. J. Eur. Union (OJ) **59**, 1–88 (2016)
10. Loukil, F., Ghedira-Guegan, C., Boukadi, K., Benharkat, A.N.: LIoPY: a legal compliant ontology to preserve privacy for the Internet of Things. In: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), pp. 701–706. IEEE (2018)
11. Novo, O.: Blockchain meets IoT: an architecture for scalable access management in IoT. IEEE Internet of Things J. **5**(2), 1184–1195 (2018)
12. Zhang, Y., Kasahara, S., Shen, Y., Jiang, X., Wan, J.: Smart contract-based access control for the Internet of things. IEEE Internet of Things J. **6**(2), 1594–1605 (2018)