IoT Data Privacy via Blockchains and IPFS

Muhammad Salek Ali
University of Bologna / FBK Create-Net, Trento, Italy
ms.ali@fbk.eu

Koustabh Dolui
FBK Create-Net, Trento, Italy
k.dolui@fbk.eu

Fabio Antonelli
FBK Create-Net
Trento, Italy
fantonelli@fbk.eu

ABSTRACT

Blockchain, the underlying technology of cryptocurrency networks like Bitcoin, can prove to be essential towards realizing the vision of a decentralized, secure, and open Internet of Things (IoT) revolution. There is a growing interest in many research groups towards leveraging blockchains to provide IoT data privacy without the need for a centralized data access model. This paper aims to propose a decentralized access model for IoT data, using a network architecture that we call a modular consortium architecture for IoT and blockchains. The proposed architecture facilitates IoT communications on top of a software stack of blockchains and peer-to-peer data storage mechanisms. The architecture is aimed to have privacy built into it, and to be adaptable for various IoT use cases. To understand the feasibility and deployment considerations for implementing the proposed architecture, we conduct performance analysis of existing blockchain development platforms, Ethereum and Monax.

Author Keywords

Internet of Things; Privacy; Blockchain; Distributed Access Model; Computer Networks

ACM Classification Keywords

K.4.1. Public Policy Issues: Privacy; C.2.1. Network Architecture and Design: Network topology

INTRODUCTION

The Internet of Things (IoT) is a network of physical objects with computational capabilities, connected over the Internet. These devices monitor and collect information about their surroundings and communicate with one another to perform automated tasks and services. Despite the benefits of the IoT, the sheer amount of data generated by it can lead to privacy issues. IoT, as it is currently implemented, works on a centralized, client-server based access model in which user data is entrusted with centralized service providers. IoT devices and the data they generate can reveal personal information of the users including their behaviours and preferences. Centralized service providers can make illegitimate use of IoT data; mass-surveillance programs being one example [1].

To introduce privacy for users' data in IoT, the goal is to develop a decentralized IoT, that has privacy built into it by design. In a world where data-driven insight directly translates to wealth, democratizing IoT data and giving users full authority over their data will foster a revolutionary model in digital commerce [2]. Decentralized IoT data management will give users the choice of sharing or selling their sensor data with third party entities without intermediaries. The objective therefore, is to provide a decentralized data access model for IoT, which ensures that user-data is not entrusted to centralized entities or companies, but instead is made the property of the users themselves.

Blockchains and other peer-to-peer techniques can prove to be crucial in realizing this goal [3]. The blockchain is a peer-to-peer distributed data structure that represents an immutable ledger of 'transactions'. The transactions refer to data exchanges that occur in a network. In cryptocurrency networks, the transactions involve transfer of cryptocurrency. In a blockchain network, the entire ledger is distributed over all the nodes, and every node casts a vote over the validity of every new transaction that is added to the ledger. Since the same copy of the blockchain is maintained over all the nodes using peer-to-peer consensus algorithms, there is no central entity entrusted with maintaining transaction records. Therefore, blockchains create a 'trust-less' environment with accountability built into it.

Towards a decentralized data access model for IoT data privacy, we propose an IoT network architecture that we call a 'modular consortium network'. In this architecture, we use a software stack of blockchain and IPFS (the peer-to-peer 'Interplanetary file system') to provide decentralized access control. Furthermore, the blockchain provides an immutable log of all IoT data operations including sensor data creation and IoT data access. The IoT devices being used for any singular use case are grouped together into private blockchains, or 'sidechains'. Users can own multiple sidechains, and each sidechain network is made responsible for maintaining a secure log of the IoT data operations that occur within it. The sidechain networks are connected in a modular fashion to a larger, decentralized peer-to-peer 'consortium' network, which runs its own blockchain. The consortium blockchain is responsible for securely logging any incoming access requests for any user's IoT data, and

IoT 2017, October 22–25, 2017, Linz, Austria

DOI: **https://doi.org/10.1145/3131542.3131563**

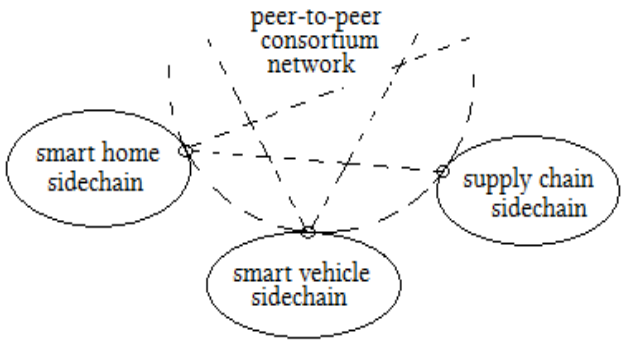

Figure 1. Modular consortium network, a decentralized peer-to-peer network made up of isolated sidechain networks

performing access control for those incoming requests. Figure 1 shows a high-level depiction of our modular consortium network. Here, modularity allows individuals or groups to seclude themselves, or information about themselves, and thereby express themselves selectively.

Current blockchain development platforms are continuously under development, and deployment considerations need to be made when applying blockchain based applications to IoT. To better understand these deployment considerations, we conducted a performance analysis on existing blockchain platforms, Ethereum and Monax. The reason for this choice is the difference in the consensus mechanisms that these platforms employ.

STATE OF THE ART

IoT data privacy continues to be a research challenge because of the lack of standardization in IoT, the sheer scale of IoT networks, and centralized access models for IoT data. Numerous research contributions have been made for enabling secure data accessing in the traditional 'client-server'-based access control models [4,5]. IoT service providers also use proprietary authorization techniques, where they act as centralized authorizing entities. However, centralized IoT data management and access control models lead to scalability issues in IoT, and force users to place their trust in centralized third-party intermediaries to manage their data, thus compromising user data privacy as well as end-to-end security [13]. As such, research focus is being drawn towards developing a decentralized security and privacy model for IoT, using blockchains along with peer-to-peer data storage mechanisms. The reason for the research popularity of blockchains in the IoT domain stems from their proven potential of providing security to large scale distributed networks, as seen in Bitcoin, and other cryptocurrency networks.

In [6], the authors describe the capability of blockchains to maintain an immutable log of data exchanges as well as to perform access control. The access control element comes from creating access polices around the public key infrastructure (PKI) of blockchain networks [7,8]. Authors in [9,10] highlight the benefits of ensuring users' ownership of IoT data via blockchains. They discuss the potential of blockchains for facilitating an economy for sensor data, and users with full ownership of their private data can choose to sell their data to third-party entities. Authors in [11] describe a data-centric log of storing and accessing data, and [12] proposes a framework for implementing blockchains to provide access control while maintaining the privacy of user data, by adding programmable access control mechanisms to the blockchain. In [13], the authors leverage the programmability of the blockchain development platform Ethereum to provide IoT device management in a fine-grained fashion.

IBM Adept [14], a collaboration between IBM and Samsung, aims to harness the blockchain and develop a decentralized platform for IoT. Adept uses TeleHash for peer-to-peer messaging, and the Ethereum blockchain development platform on top of BitTorrent for peer-to-peer file sharing. The issues IBM Adept faces in implementing a blockchain based solution for decentralizing IoT are the poor scalability of blockchains and the inherent latency in blockchain consensus. The authors in [6] suggest dividing the IoT blockchain network into smaller sub-networks, since a single blockchain cannot scale well enough to perform consensus based validation of transactions coming from a constantly growing large scale networks [15]. The authors of [16] propose a layered architecture for blockchains in IoT, with one overlay blockchain network connecting to sub-networks of local blockchains. They provide access control by writing access policies into the blockchain headers. Data retrieval from this network is done by sending IoT data to the requester, stored on either local storage or third-party cloud storage services.

The Internet of Things, as it exists today, is a distributed network of intelligent objects with software and data management being provided by centralized third party entities. Not only does this client-server model add to the limitations of the network, it also forces users to place their trust on third-party entities to manage their data and not misuse it. Therefore, the research problem to be solved is leveraging blockchains and peer-to-peer data storage techniques for IoT data privacy, *where each user has complete authority over their data without trusting any third-party entities to manage IoT software or data*.

MODULAR CONSORTIUM ARCHITECTURE FOR BLOCKCHAINS IN IOT

To provide decentralized access model for IoT data privacy, we propose an IoT network architecture we call the 'modular consortium' architecture. With this architecture, we aim to provide IoT data privacy via blockchains, and address the challenges associated with implementing blockchains to IoT networks, as discussed in the state of the art. One of the challenges is the scalability of blockchains in IoT. Since every entry on the blockchain requires consensus among the network nodes, a single blockchain responsible for logging every IoT data operation would not scale well. To address

this issue, we break the network down into smaller, private blockchains, or 'sidechains'. These sidechain networks are connected together to form one decentralized, peer-to-peer 'consortium' network. The benefit of implementing private sidechains is that the nodes participating in one sidechain's consensus algorithm don't have to validate or verify transactions occurring in an entirely different sidechain. Users can own and manage one or multiple sidechains as per the users' requirements. Each sidechain network is used separately for different IoT use cases, and can be added or removed from the consortium network in a modular fashion.

In order to access the IoT data of one sidechain, the requester would have to become a member of the peer-to-peer consortium network and make a request for access. The consortium network runs its own blockchain which we refer to as the consortium blockchain. The consortium blockchain performs access control and prevents any unauthorized access of IoT data from one sidechain to another member of the consortium network. While each sidechain is tasked with logging sensor data creation events within the sidechain networks, the consortium blockchain is responsible for maintaining a log of successful and unsuccessful data access requests that are made from one member of the consortium to another.

Keeping the logging responsibilities of the sidechains and consortium blockchain separate helps mitigate the issue of public visibility of the blockchain contents. If a single blockchain were to log all IoT data operations, it would compromise privacy, since the log contents of the blockchain are accessible by all members of the blockchain network. However, in the case of our proposed architecture, any member of the consortium network will only be able to access the records of IoT data access requests, while logs of IoT data creation events remain private within the sidechains.

Accountability for Requesters and Requestees

Allowing the members of the consortium network access to an immutable log of all successful and unsuccessful access requests provides accountability to both requesters and requestees. Consider the scenario where a user sells his sensor data to a marketing company. The user agrees to allow the company access for one month. If the user revokes access rights on the sidechain level before the month is up, the company's request access will be unsuccessful; and as evidence of misconduct by the sidechain owner, the company can produce timestamped logs of its unsuccessful access requests.

Blockchain and IPFS Software Stack

A decentralized access model, with privacy built into it by design, is made possible by a software stack of blockchains working on top of a decentralized peer-to-peer file system, IPFS. IoT data can be grouped together and stored in IPFS, while the blockchain itself only needs to hold the hash of the IPFS files containing the IoT data. The programmability and PKI properties of blockchains enable secure access control, and IPFS eliminates the need for centralized IoT data management.

Enabling Technology: Blockchain

The blockchain is a peer-to-peer ledger that records transactions in a network. The transactions represent transfer of data; Bitcoin is just one use-case of blockchains where the transactions represent transfer of crypto-currency. Public key cryptography is used to issue transactions; nodes sign transactions with their private key and address it to other nodes using their public keys. The transactions are relayed on to the blockchain network, and the peers responsible for validating transactions (also called 'miners') group the transactions together into candidate blocks. The validating peer that successfully solves a complex mathematical puzzle involving the hash of the block, is selected to add its candidate block to the blockchain. Each new added block contains the hash of the previous block, as shown in Figure 2. Linking the blocks this way makes the contents of the blockchain immutable, since any changes in a previous block are easily detectable. The remaining nodes verify that the new block is valid, and update their copies of the blockchain. Every node carries the same copy of the blockchain and 'trust-free' distributed networking is established.

Alternate Block Validation Methods

The method of validating new transactions into blocks by computational puzzle solving is called 'Proof-of-Work' (PoW). In PoW, the node adding a new block to the blockchain is chosen based on a race to find a nonce that when hashed with the rest of the block, produces a result with a specified number of leading zeroes. Thus, PoW-based consensus is computationally intensive and introduces latency, both of which are undesirable in IoT. An alternate consensus algorithm, 'Proof-of-Stake' (PoS) chooses the validating node for adding their candidate block in a pseudo-random fashion. The probability of a node being chosen is proportional to the share of the stake they hold in the network. The advantage of PoS is that it does not consume large quantities of computational power to validate new transactions and add them to the blockchain. However, PoS makes the network vulnerable to the 'nothing-at-stake' attack [17], where nodes can easily create forks in the chain without holding any stake in the network or expending a lot of computational power. Blockchain development platforms adapting PoS like Monax and Ethereum's Casper[18] deal

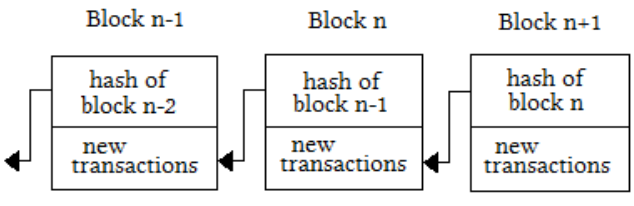

Figure 2. Blocks in a blockchain, with headers containing the hash to the previous block; the first block in the blockchain being the 'genesis' block

with nothing at stake attacks by slashing the stake of any nodes that create false blocks. PoS provides us with the opportunity to further explore the viability of blockchains in IoT, since low processing requirements are desirable for IoT networks.

Smart Contracts and Blockchain Programmability

'Smart contracts' bring programmability to the blockchain, in the sense that they facilitate the execution of transactions while meeting the terms of a contract written in code. Smart contracts are deployed in the blockchain with specific addresses, so in order to invoke a function written in a smart contract, transactions are signed off by nodes and addressed to the smart contracts themselves. While smart contracts enforce terms and conditions for transactions in financial applications, they can enforce access control policies in the modular consortium architecture.

Enabling Technology: IPFS

IPFS (Interplanetary File System) aims to be a fully decentralized file system for the Internet [19]. IPFS currently makes up a subsystem of the Internet as one single peer-to-peer swarm, making file exchange over the Internet secure and open. IPFS files are content-addressed and are identified by their hashes. This makes them cache-friendly, and IPFS file hashes can be easily stored within the blockchain. IPFS is therefore, an ideal file storage and sharing platform for developing a decentralized access control model for IoT.

WORKING PRINCIPLES OF THE MODULAR CONSORTIUM ARCHITECTURE FOR IOT

Private Sidechains for IoT Users

The private sidechains maintain logs of all IoT data operations that occur within a private IoT network. The private IoT network consists of IoT devices and one validator node running the sidechain. The IoT devices are given unique public and private keys, which they use to send encrypted sensor readings to the validator node. The validator node logs any data received with authorized key encryption as data creation events. The validator node adds new blocks to the sidechain, and has higher computational power as well as storage space. A smart contract is deployed within the sidechain to perform the following functions:

- Storing a dictionary of each authorized smart device's public key and the hash of the IPFS file storing the smart device's data.
- Ensuring that only the data incoming from authorized smart devices are able to communicate with the sidechain validator.
- Storing a dictionary of public keys of requesters in the consortium network with access privileges, and the public keys of the smart devices whose data the requesters have access to.
- Performing access control on incoming access request transactions.

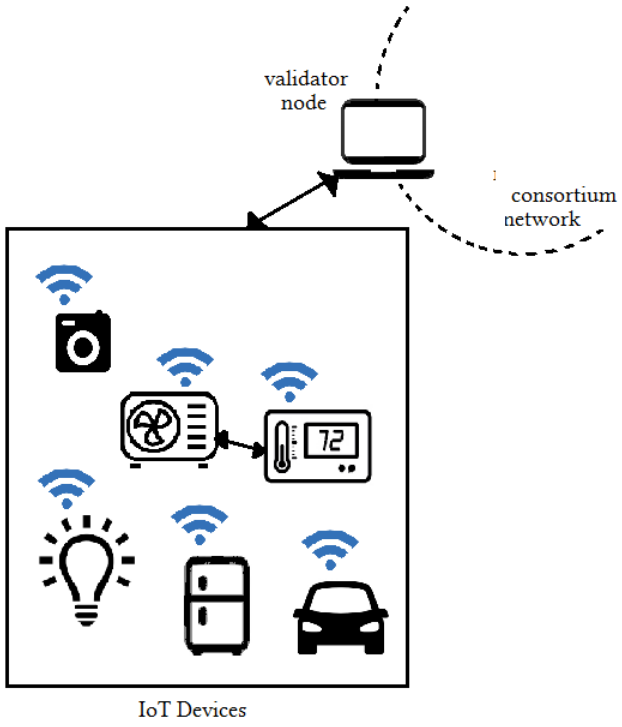

Figure 3. Sidechain Network, with IoT devices, and the validator node connecting the sidechain to the consortium

To add more IoT devices to the network, each new device will be given a public key, which will be stored onto the smart contract in the sidechain. Maintaining a list of authorized smart devices in the validator node ensures no unauthorized device can flood the validator node with fake incoming data.

Storing IoT Data

Whenever a smart device generates data, it encrypts it and sends it to the validator node. The validator node identifies any unauthorized device attempting to participate in the sidechain. Subsequently, the validator node blocks all incoming data packets from the unauthorized device. Upon receiving data from an authorized device, the validator node logs a 'creation of IoT data' transaction in a new block. The validator node then updates the IPFS file, as well as its hash stored in the smart contract.

Consortium Blockchain Network

The validator nodes in each sidechain are connected to form the decentralized consortium network, running a global consortium blockchain. The consortium blockchain carries within itself a smart contract that stores a list of the public keys of devices authorized to make access requests for IoT data. While the private sidechains are responsible for securely storing IoT data and keeping a log of all the local IoT events, the consortium blockchain is responsible for allowing access to the IoT data to external requesters who are given access privileges. The access policies written in the consortium blockchain's smart contract, together with the

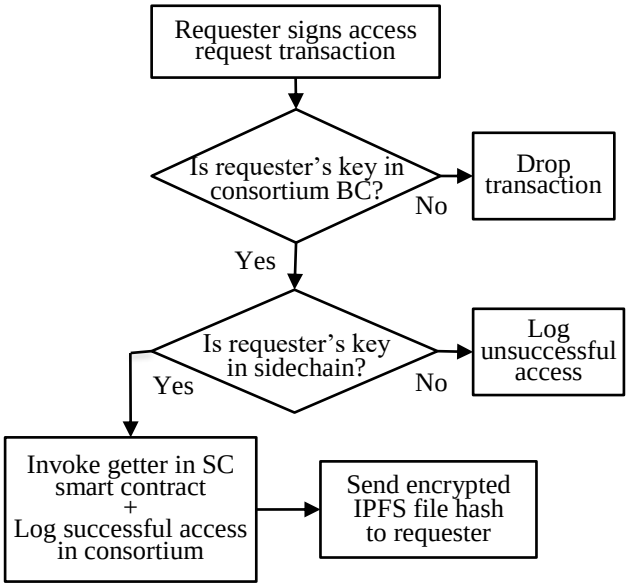

Figure 4. Flowchart of steps for access control following an access request transaction

access policies written in the sidechain's smart contract, provide fine-grained access control so as to limit requesters to only be able to access the data of specific smart devices within specific sidechains.

Accessing IoT Data

To access user data, a requester joins the consortium blockchain network via a client application as a participating peer, i.e., the requester can make requests for the data, but doesn't add blocks to the consortium blockchain. In the case of either IoT data sharing or selling, the sidechain owner adds the requester's public key to the sidechain's smart contract along with the amount of time the requester has access privileges. At this point, the validator node adds this same public key to the list of authorized requesters in the consortium blockchain. This is to ensure that the requester is only able to make access requests on the consortium level, if a sidechain owner vouches for them.

To gain access to a user's IoT data, the requester signs an access request transaction with their private key. Figure 4 shows a flowchart of the steps taken in response to an incoming access request on the consortium network

Upon receiving the encrypted IPFS file hash, the requester can decrypt it using their private key, and access the file containing the smart device's data via HTTP using www.ipfs.io/ipfs/<FileHash>. The ability to modify access policies gives the user the choice to share their IoT data with whomever they choose.

The smart contract on the consortium blockchain also prevents requesters from flooding the consortium blockchain with unauthorized request transactions. After a requester makes a specified number of consecutive unsuccessful requests, the smart contract removes the associated public key from the list of authorized requesters.

Transactions Per Minute	Ethereum	Monax
10	~97%	~0.5%
300	~98%	~0.6%

Table 1. CPU usage on core-i7 laptop as validator node

IMPLEMENTATION AND PERFORMANCE ANALYSIS

To gain insight about the feasibility of implementing the proposed architecture and the relevant deployment considerations, we conducted a performance analysis of existing blockchain application development platforms on both the sidechain and consortium level. We used Ethereum, the blockchain development platform, whose cryptocurrency Ether is second only to Bitcoin. Other than Ethereum, we used Monax, the blockchain application development platform for business ecosystems. Ethereum achieves consensus via the PoW algorithm. Monax achieves consensus using the Tendermint [20] consensus engine, which employs PoS.

Our testbed was built on a consortium of five validator nodes, each receiving incoming data from five smart devices. The performance metrics we used for our analysis were processing overhead, network traffic overhead and block processing times.

Processing Overhead

On the sidechain level, we conducted tests on CPU usage when validating new blocks with Ethereum and Monax. We compared processing overhead by comparing CPU usage of the blockchain client's specific process ID. Considering five smart devices are connected to each validator node, we carried out tests with varying number of incoming transactions from within the sidechain network. Throughout all variations in the incoming transactions, the processing overhead remained unchanged with both platforms. Therefore, in Table1, we display the processing overhead only for the lowest and highest transaction rates that we tested.

Network Traffic Overhead

In blockchain applications, there is network traffic overhead that comes from the nodes of the network participating in the consensus algorithm. We measured traffic overhead for the consortium blockchain, because the sidechains only involve one validator node. For this experiment, the rate of access requests transactions is assumed to be less than the data creation transactions within the sidechains.

As observed from the network traffic overhead coming from Ethereum and Monax, while Ethereum traffic does increase with increasing number of nodes in the consortium, it is significantly smaller than the traffic overhead of Monax. The

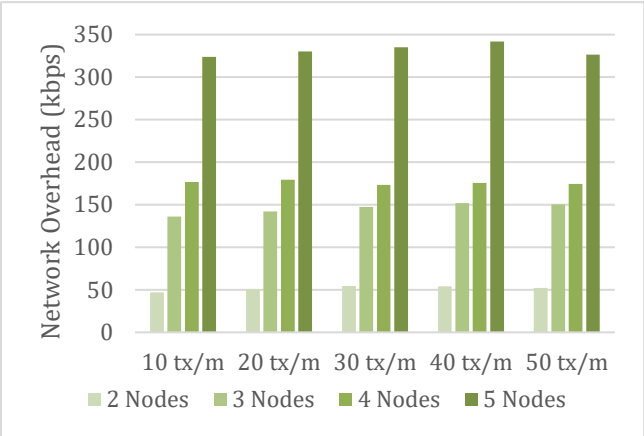

Figure 5. Network Traffic Overhead in Monax

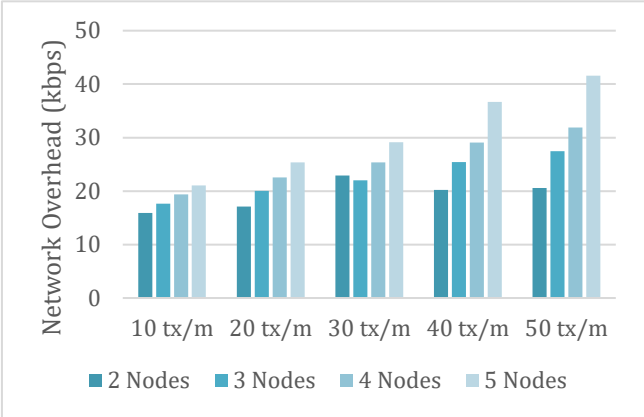

Figure 6. Network Traffic Overhead in Ethereum

high network overhead in Monax is due to the fact the the Tendermint consensus engine sends out empty blocks as heartbeat to check if a peer is up. Monax was developed for business applications and it was not meant to be used in a scalable public network, the way the consortium network aims to be. In this experiment, we measured network traffic with varying number of nodes in the consortium network, and varying amount of access request transactions incoming per minute. The observations we gathered from this experiment are illustrated in Figures 5 and 6.

Block Processing Time

In its current stage of development, Monax does not group different transactions into one block, and in our tests, it processed blocks consistently at ~1 block per second, with one transaction per block. On the other hand, Ethereum grouped transactions together in blocks. Figure 7 shows the block processing time information we obtained from the Ethereum client's log file.

We ran the experiment with Ethereum for 60 minutes, and set the 'difficulty' in the genesis block to 200000. Difficulty relates to the hashing power required for processing. In this experiment, we observed a trend of rising difficulty and processing time with increasing number of blocks.

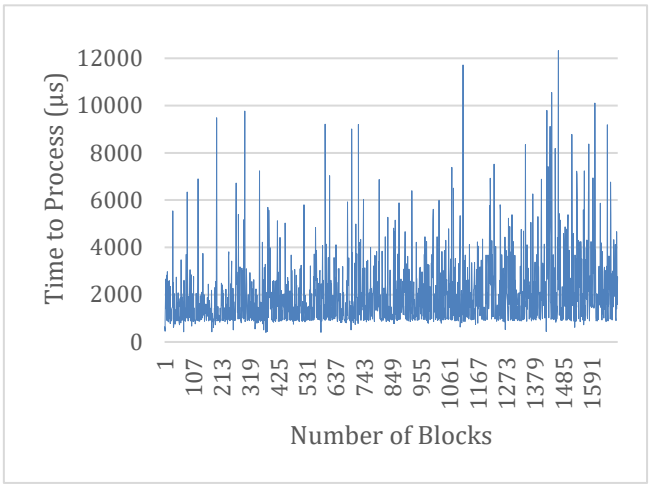

Figure 7. Block Processing Time in Ethereum

We have made the log files of our experiments available via IPFS and can be accessed here:

http://ipfs.io/ipfs/QmPNUYLFu2To9b5YYgrpYqQeREphwqN9iBe5WyFnGrzQEp

DEPLOYMENT CONSIDERATIONS

While using Ethereum and Monax to implement the modular consortium architecture, there are certain deployment considerations that need to be made. Firstly, Monax blockchain development platform was built for private business applications, and judging from the network traffic overhead, it does not scale well on the consortium level. When used on the sidechain level, Monax outperforms Ethereum in terms of processing overhead. However, from the observations we made of Monax's block processing time, creation of data from different streams or IoT devices will need to be bundled together natively in the validator node, and logged together as a single transaction in Monax.

On the consortium level, Ethereum performs well in terms of network traffic overhead, however, to validate new blocks on an Ethereum based consortium blockchain, every validator node will be faced with processing overhead. To alleviate the processing overhead from the validator nodes, separate dedicated validators can be set up on the consortium level. In that case, while validation of new blocks will not be fully decentralized, the members of the consortium network will be able to sign access request transactions and access the consortium blockchain. Even if one of the dedicated validator nodes is compromised, the sidechain owners' IoT data remains private, since access privileges are set at the sidechain level.

FUTURE WORK

We aim to continue our work with the modular consortium architecture by stress testing it against different use cases and with larger scale networks. We intend to implement other blockchain development platforms like Hyperledger and Parity to assess their suitability for our architecture, as well

as design customized blockchains to optimize them specifically for IoT. Further work needs to be done on how to advertise the availability of certain data in a sidechain, either via broker or by keeping a list within the consortium smart contract. As of now, the architecture facilitates fetching data from specific time periods. More work will be carried out to maintain a data stream for valid requests via IPFS. We intend to explore blockchain pruning to combat the trend of rising block processing time, as well as conserving storage space on validator nodes.

CONCLUSION

In this paper, we propose a network architecture for providing IoT data privacy via blockchains and IPFS. In the proposed 'modular consortium' architecture, blockchain smart contracts perform access control, while providing accountability for both the data owners and the third parties whom the users allow access to. We built implementations of the architecture using two existing blockchain application platforms. Performance analysis of the blockchain platforms provided insights into the architecture's feasibility and further considerations for deploying a usable implementation. The main contribution of this paper is to implement a software stack of blockchain smart contracts and peer-to-peer file storage, to give IoT users authority over their data, and to eliminate the need for centralized IoT data management.

REFERENCES

1. NSA Prism program taps in to user data of Apple, Google and others. Retrieved May 2, 2017 from https://goo.gl/2RCCQB
2. Yu Zhang and Jiangtao Wen. *Peer-to-Peer Netw. Appl.* (2017) 10: 983. doi:10.1007/s12083-016-0456-1
3. Marco Conoscenti, Antonio Vetro and Juan C. D. Martin: Blockchain for the Internet of Things: A Systematic Literature Review. *In Proceeding of The Third International Symposium on Internet of Things: Systems, Management and Security (IOTSMS-2016)*.
4. Yves-Alexandre de Montjoye, Erez Shmueli, Samuel S Wang, and Alex Sandy Pentland. 2014. openpds: Protecting the privacy of metadata through safeanswers. *PloS one* 9, 7 (2014), e98790.
5. Vijay Sivaraman, Hassan Habibi Gharakheili, Arun Vishwanath, Roksana Boreli, and Olivier Mehani. 2015. Network-level security and privacy control for smart-home IoT devices. *In Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on. IEEE*, 163–167
6. Zyskind, Guy, Oz Nathan, and Alex Pentland. "Enigma: Decentralized computation platform with guaranteed privacy." arXiv:1506.03471 (2015).
7. Loise Axon. 2015. Privacy-awareness in Blockchain-based PKI. Retrieved April 12, 2017 from http://goo.gl/3Nv2oK
8. Fromknecht Conner, Dragos Velicanu, and Sophia Yakoubov. CertCoin: A NameCoin Based Decentralized Authentication System. Retrieved March 29th, 2017 from https://courses.csail.mit.edu/6.857/2014/files/19-fromknecht-velicann-yakoubov-certcoin.pdf
9. Yu Zhang and Jiangtao Wen. 2015. An IoT electric business model based on the protocol of bitcoin. *ICIN. IEEE*, pp. 184–191
10. Wörner Dominic, and Thomas von Bomhard. 2014. When your sensor earns money: exchanging data for cash with Bitcoin. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 295–298
11. Shafagh Hossein, Anwar Hithnawi, and Simon Duquennoy. 2017. Towards Blockchain-based Auditable Storage and Sharing of IoT Data. *arXiv preprint* arXiv:1705.08230 (2017)
12. Guy Zyskind, Oz Nathan and Alex Pentland. 2015. Decentralizing Privacy: Using Blockchain to Protect Personal Data. In *Proceeding of 2015 IEEE Security and Privacy Workshops (SPW)*, DOI: 10.1109/SPW.2015.27
13. Ouaddah Aafaf, Anas Abou Elkalam, and Abdellah Ait Ouahman. 2017. Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology in IoT. *Europe and MENA Cooperation Advances in Information and Communication Technologies. Springer International Publishing, 2017*. 523-533.
14. Device Democracy: Saving the Future of the Internet of Things. Retrieved May 10, 2017 from https://goo.gl/18Y16F
15. Barber, Simon, Xavier Boyen, Elaine 7Shi, and Ersin Uzun. 2012. Bitter to better—how to make bitcoin a better currency. In *International Conference on Financial Cryptography and Data Security*, Springer Berlin Heidelberg, (2012), 399-414.
16. Ali Dorri, Salil S. Kanhere, and Raja Jurdak. 2017. Towards an Optimized BlockChain for IoT. In *Proceedings of the Second International Conference on IoT Design and Implementation. ACM, 2017*
17. Ethereum Wiki: Problems. Retrieved May 17, 2017 from https://github.com/ethereum/wiki/wiki/Problems
18. Ethereum Wiki: Proof of Stake FAQ. Retrieved May 17, 2017 from https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ
19. IPFS – The Permanent Web. Retrieved May 18, 2017 from https://github.com/ipfs/ipfs
20. Kwon, Jae. 2014. Tendermint: Consensus without mining. Retrieved May 18, 2017 from http://tendermint. com/docs/tendermint {_} v04. pdf