MaRSChain: Framework for a Fair Manuscript Review System Based on Permissioned Blockchain

Nitesh Emmadi(✉), Lakshmi Padmaja Maddali, and Sumanta Sarkar

TCS Innovation Labs, Hyderabad, India
{nitesh.emmadi1,lakshmipadmaja.maddali,sumanta.sarkar1}@tcs.com

Abstract. Current Manuscript Review Systems (Conference/Journal) rely on a centralized services (like EasyChair, iChair, HotCRP or EDAS), which manage the whole process that starts with manuscript submissions to notification of the results. As these review systems are centralized, the trust is based on a single entity. The fairness of the system hinges on the honesty of the central controlling authority. This dependency can be avoided by decentralizing the source of the trust. Bitcoin has shown the power of decentralization and shared database through blockchain technology, and currently is being studied for its immense impact on FinTech. We leverage blockchain to address the above concern and present a decentralized manuscript review system that provides trust and fairness. We call this system MaRSChain. As a proof of concept, we develop a prototype of MaRSChain system on top of Hyperledger Fabric platform. To the best of our knowledge, this is the first ever decentralized manuscript review system based on Blockchain.

Keywords: Manuscript review system · Blockchain
Consensus · Fairness · Trust · Smart contract

1 Introduction

Ever since the cryptocurrency Bitcoin [16] has shown the application of blockchain technology that rules out the need of central authority, it has drawn attention from both the industry and academia. Blockchain enables mutually distrusting parties form a peer to peer distributed network and maintain a common transaction ledger. A typical blockchain as in Bitcoin does not need verified identity of a peer, i.e., it is an open enrollment system. In other words it is a permission-less blockchain. Research has been carried out to embrace Blockchain's decentralization feature in several applications ranging from finance [10], supply chain [9], IoT [18], and to many other business use cases. A report by world economic forum predicted that 10% of global GDP would be stored on blockchain technology by 2025 [1]. The idea of permission-less blockchain may not be suitable for many enterprise applications, like banks, which require

G. Mencagli et al. (Eds.): Euro-Par 2018 Workshops, LNCS 11339, pp. 355–366, 2019.
https://doi.org/10.1007/978-3-030-10549-5_28

their users to be verified. To cater to this kind of applications, we need a permissioned model of blockchain, for example, Hyperledger Fabric [5]. With this permissioned blockchain many centralized services can be decentralized, which is now being explored. In this paper we focus on the applicability of permissioned blockchain to build a decentralized conference management system.

A conference management system (for example, EasyChair [2], EDAS [3], HotCRP [4] or iChair [7])[1] handles the life-cycle of a conference from manuscript submissions to acceptance/rejection notification. A conference program committee, first invites manuscripts for reviews and assigns the manuscripts to reviewers for evaluation. Based on the evaluations submitted by the reviewers, the conference program chairs decide on the manuscript acceptance. The accepted papers are invited for publication in the conference proceedings. Current systems are flexible, easy to use and have many features that make them powerful event managers for conducting international conferences. However, they are centralized services, thereby giving the hosting entity full control of the system. A malicious party in control of the system can manipulate decisions and results impacting fairness of the system. For instance, the controlling entity can assign papers to reviewers of his choice and hamper the fairness of reviews, or can change the results in the system. To address the above challenges, we propose a decentralized framework for fair manuscript review system based on permissioned blockchain. A conference review system is an application operating in controlled environment that employs parties with verified identities i.e., authors, reviewers, program chairs. Hence, a permissioned model of blockchain is suitable in case of applications where the distrusting parties involved have verified identities [15].

1.1 Related Work

Apart from the widely used conference management systems (such as EasyChair, EDAS, HotCRP or iChair), other notable systems are ConfiChair [13], P3ERS [12] and CryptSubmit [14]. ConfiChair proposes an architecture to build conference management systems in a privacy preserving manner in order to protect the privacy of entities (authors/reviewers/Program Chairs(PCs)) against untrusted cloud service providers. It preserves privacy and confidentiality using encryption mechanisms with key translations and mixes. P3ERS (Privacy Preserving Peer Review System) is a distributed peer review system with several group managers. P3ERS preserves privacy of all the users in the system with an improved group signature scheme. P3ERS considers an untrusted cloud service provider and actors within the system as potential adversaries and proposes a distributed architecture to host different services on different servers. It ensures privacy of authors and reviewers from PCs by creating separate services for them. These systems are still centralized and address privacy concerns within the conference system.

[1] EasyChair, EDAS and HotCRP are third party services whereas iChair is an open-source software that can be hosted by any of the program chairs of a forum.

*MaRSChain is listed in Hyperledger's inventory of usecases [6].

CryptSubmit proposes a manuscript review system with timestamped submissions and reviews. In this system, the hash of the submissions and reviews are timestamped on the public Bitcoin blockchain that lies outside the actual review system, still keeping the actual review system centralized. The manuscripts are timestamped outside the system and the review system does not guarantee proof of manuscript or review submissions into the system.

Centralized systems do not guarantee security against single point of failure and hence there is a need for decentralized manuscript review system. In this regard, we propose MaRSChain, a decentralized solution where the actual review process is done on the blockchain. Decentralization ensures that a malicious entity can not corrupt the system to modify/remove submissions and reviews from the system. Our solution aims to improve trust in the system by leveraging blockchain to decentralize the system. This is the first ever manuscript review system based on blockchain.

1.2 Our Contribution

In this paper, we propose MaRSChain, a framework to build a manuscript review system based on a permissioned blockchain. We leverage Hyperledger Fabric [5], a permissioned blockchain platform, to build our system. MaRSChain can be built on top of any permissioned blockchain platform which provides features described in this paper. We employ several smart contracts to handle submissions and reviews, validation of submissions and consolidation of reviews. In the usual centralized conference management systems, the PCs have immense power and control over the system. A malicious PC can manipulate reviewer assignments or can modify/remove reviews and etc.

MaRSChain promises:

- **Security against manipulation of manuscript reviewers assignment:** In a centralized system, a malicious PC can assign a manuscript to reviewers of his choice, in order to hamper the fairness of reviews. A decentralized solution ensures that a malicious PC can not manipulate the reviewers assignment.
- **Security against manipulation of manuscript reviews:** In a centralized system, a malicious PC can manipulate the reviews to influence the acceptance/rejection of a manuscript. Decentralization guarantees that a malicious party in the system can not manipulate reviews.
- **Confidentiality and privacy of manuscript submissions and reviews:** A permissioned blockchain employs encryption and pseudonymous identities along with access controls to better preserve confidentiality and privacy of the authors/reviewers.

2 System Overview

A conference/journal forum forms a peer-to-peer network of blockchain nodes. We refer to this blockchain network as Conference Blockchain (CBC). The entities in a CBC are listed below:

- Authors
- Reviewers
- Program Chairs

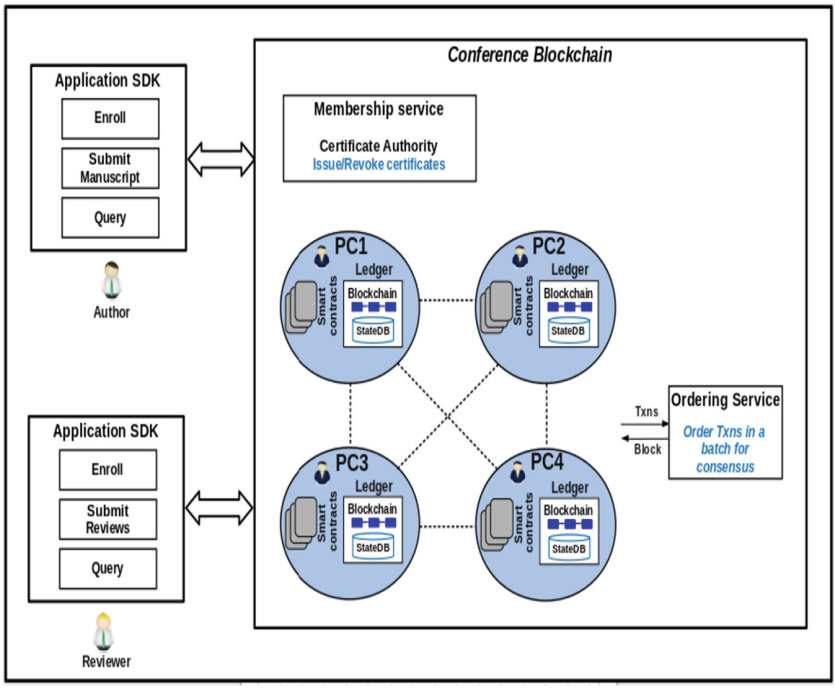

Fig. 1. Conference blockchain (CBC)

All the entities in CBC Blockchain are registered into the network by Membership Service. The Membership Service hosts a Certificate Authority that is responsible for issuing/revoking certificates to the entities. These certificates are identities of the entities and are used to transact on the blockchain ledger. Authors and Reviewers are end-users in the blockchain network. Users submitting manuscripts to the conference are authors. Reviewers review manuscripts submitted by the authors. PCs are the validating entities that are responsible for manuscript validation, reviewers assignment and reviews consolidation.

Operational Flow: A conference blockchain can be visualized as in Fig. 1. Below, we describe the operational flow of our system.

1. Manuscript Submission. All the inputs to the blockchain systems are signed transactions which are recorded in the blockchain. Authors submit their manuscripts to a conference as signed transactions. On receiving the manuscript transactions, the PC nodes in a CBC validate the transactions against submission policy (semantics, duplicates, signatures). If the transactions are valid, then the manuscripts are accepted by the conference for review and are recorded onto the blockchain ledger.

2. Reviewers Assignment. Once the manuscript acceptance window is closed, the PCs reach consensus on assigning the submitted manuscripts to reviewers for evaluation. For ease of implementation, we assume that this consensus is an offline process and all the PCs have agreed to the decision on reviewers assignment. A PC initiates the reviewer assignment transaction based on their decision for each of the manuscripts. All these transactions are validated by the PCs and their consensus reflects their agreement on reviewers assignments.

3. Review Submission. Reviewers evaluate the assigned manuscripts and submit their evaluations to the system in the form of transactions. Reviewers' evaluation transactions are validated by the PC nodes and the evaluations are updated in the system. An evaluation is a score awarded to the manuscript by the reviewer along with justifications (comments).

4. Results Declaration. Once the review window is closed and the reviews are received, the PCs consolidate the evaluation scores of each of the submissions and updates the results of the submissions. These scores reflect the decision of the conference. These scores and results are then notified to the authors.

3 Security of Our System

In this section, we describe security guarantees of our system.

3.1 Security Against Manipulation of Reviewers Assignment

In a centralized system, a malicious PC can affect the fairness of the reviews by assigning the manuscripts to reviewers of his choice in order to get his desired evaluation result. Consider a malicious PC submits a reviewers assignment transaction with the manipulated reviewers assignment to the system. The transaction submitted by malicious PC goes through consensus where each of the other PCs validate the transaction. A malicious assignment can easily be detected by the other PCs and the transaction is rejected during the consensus. This ensures protection against unfair reviewers assignment. The consensus algorithm ensures that a malicious PC can not influence reviewers assignment at his will without corrupting some other PCs.

3.2 Security Against Forgery or System Corruption

We assume that a malicious PC member is trying to manipulate review evaluations to impact the decision on a manuscript. This can be done either by forging review transactions of the reviewers or by corrupting all the PC nodes to exclude certain review transactions.

Consider a malicious PC member as an adversary trying to manipulate the review of a manuscript. To do so, the adversary has to do one of the following: forge signatures of reviewers or exclude reviews from ledger. The reviewers submit a review transaction digitally signed by reviewer's private key which is securely stored at his end. Security of a digital signature scheme ensures that a signature is very hard to be forged. Hence, protection against forgery is ensured. Excluding the review of a manuscript from the ledger either by denying the reviewers' transaction or by re-writing the ledger is one of the ways for the adversary to manipulate the review. Decentralized system coupled with consensus ensures that a single malicious party can not influence the system. The consensus algorithm ensures that a malicious PC can not force reviews exclusion without corrupting other PCs.

The digital signature schemes and secure consensus mechanisms ensure protection against malicious PCs. Hence, security against manipulation of reviews is guaranteed.

3.3 Privacy and Confidentiality of Authors and Reviewers

All the users in the blockchain network submit transactions to the system which are validated and recorded into a common ledger. In MaRSChain, an author can monitor transactions from other authors or reviewers and their review assignments. However, for fair reviews, the forums encourage anonymous submissions and blind reviews. Lack of information about the authors limits unwarranted behavior of reviewers in evaluation of manuscripts. Hence, privacy and confidential of data on blockchain is necessary.

Privacy of the users in blockchain can be viewed in two forms: Anonymity and Unlinkability. Anonymity of transaction refers to hiding of the a user's identity in an anonymity set of all the users i.e., an identity on the ledger should not directly be associated to particular user. Unlinkability refers to the association of multiple transactions of a single user i.e., two different transactions from a same user should not be related to each other. For anonymity and unlinkability, MaRSChain provides one-time pseudonymous identities to the users. All the transactions submitted to the blockchain are under pseudonymous identities. Thus, a transaction on the blockchain ledger can neither be linked to the user directly nor to other transaction by the same user. Only authorized parties (PCs) with the knowledge of a secret have the ability to link pseudonymous identity on the blockchain to the actual identity of the user.

To enable confidentiality of the transactions, a MaRSChain encrypts transaction payloads with one-time symmetric keys. The symmetric keys are only available to the users themselves and other authorized parties in the network

(PC nodes). Hence, an unauthorized adversary can not decrypt the transactions without the knowledge of secret key. Furthermore, access-control mechanisms that restrict access to transaction payloads provide another layer of security for the transactions.

Therefore, MaRSChain guarantees privacy and confidentiality of transactions.

4 Implementation

We develop a prototype of our MaRSChain system on top of Hyperledger Fabric platform (version 1.1.0-preview). We simulate various steps of a conference management system to illustrate the system operational flow Fig. 2. We remark that a MaRSChain system can be built on top of any permissioned blockchain platform. Note that our implementation assumes the program chairs decide offline on reviewers assignment. However, the final decision is recorded on the blockchain based on consensus. If the reviewer assignment transaction differs from the actually decided offline agreement, then the program chairs can easily detect and reject the transaction. This is to enable ease of implementation. This process can be made online, yet note that, this is off-chain process and only the final decision is recorded on the blockchain. Our blockchain system is instantiated with four PC nodes and the consensus requires majority of the nodes i.e., 3 out of 4 nodes, to endorse a transaction (refer Appendix B for more details).

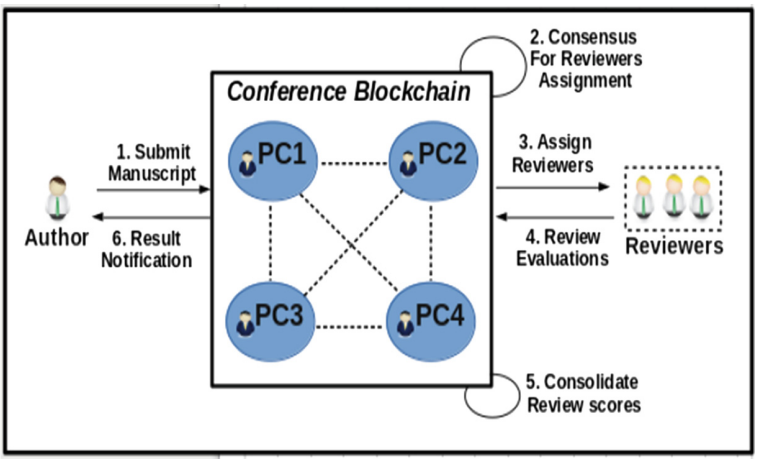

Fig. 2. CBC implementation flow

4.1 CBC Chaincode

In Hyperledger Fabric, the smart contracts are realized in chaincode that is deployed on the blockchain by peers. In this section, we describe various functionalities in our chaincode:

- ***submit_paper:*** Authors submit manuscripts by using client application to invoke *submit_paper*.

1: Manuscript submission to forum

Transaction: CBC_Submit_Txn=(*manuscript*, *author pseudonyms*)
Validation: Check if manuscript is already submitted to this forum
if *manuscript is present in the ledger* **then**
| *duplicate submission*; *rejected*
else
| *considered for submission*; *submitted*
end

- ***assign_reviewers:*** Once the manuscript submission window is closed, PCs decide on the reviewers assignment, one of the PCs invoke the *assign_reviewers* for each of the manuscripts to assign reviewers.

2: Reviewers Assignment

Transaction: CBC_Update_ReviewersList_Txn=(*manuscript_id*, *reviewers_list*)
Validation: Check if reviewers list is acceptable(as decided by program committee)
if *manuscript reviewers list is acceptable* **then**
| *update reviewers list for manuscript in CBC*
else
| *reject transaction*
end

- ***submit_review:*** Reviewers evaluate the assigned manuscripts and submit reviews by invoking *submit_review* through client application.

3: Review submission to forum

Transaction: CBC_Reviewer_Txn=(*manuscript_id*, *review_pseudonym_id*, *review*, *score*)
Validation: Check if review is submitted by assigned reviewer
if *reviewer in reviewers list* **then**
| *update review and score for manuscript*
else
| *reject transaction*
end

- ***make_decision:*** Once the review window is closed and the reviews are received, the PCs consolidate the reviews for each of the manuscripts by invoking *make_decision*.

4: Consolidate reviews of manuscripts

Transaction: CBC_Reviews_Consolidate_Txn=(*manuscripts*)
Validation: Check if manuscript status is "under-review"
for *all reviewedmanuscripts in CBC* **do**
  paper_final_score = (reviewer1_score + reviewer2_score + reviewer3_score)/3
  update manuscript final score in CBC;
  if *paper_final_score>3* **then**
    status updated to "accepted" in CBC
  else
    status updated to "rejected" in CBC
  end
end

- ***querySubmittedPaperInfo:*** Details of a submitted manuscript can be queried using *querySubmittedPaperInfo*.

- ***queryAllPaperIDs:*** List of all the manuscripts can be viewed using *queryAllPaperIDs*.

- ***queryPaperStatus:*** Status of a manuscript can be queried with *queryPaperStatus*.

The transactions corresponding to all the above functionalities can be seen in Appendix .

Our current implementation does not handle manuscript/review updations that are common in currently available review systems. Also, it is a common practice to re-consider acceptance/rejection of manuscripts with borderline threshold reviews. These features can be handled by introducing update transactions that can be linked to already submitted manuscript/review/decision.

Our system will be available in Hyperledger Fabric's Inventory of Usecases [6]. A detailed document describing the whole setup and other instructions will be available along with the implementation.

5 Conclusions and Discussions

We have proposed a framework to build a fair decentralized manuscript review system based on blockchain (MaRSChain). Our decentralized system ensures that a malicious party can not corrupt the system to hamper fairness of review process. As part of our future work, we plan to integrate a reputation system to strengthen MaRSChain further.

Another important problem in current conference management systems is to detect *double submissions* (plagiarism) and *concurrent submissions* (submission of a manuscript to multiple forums concurrently, during the review period) [17]. The current systems rely on some trusted third party services, for example iThenticate [8], to detect double submissions. iThenticate has access to publication content from several different publishers. It performs a "Similarity Check" [11]

to check for plagiarism of the submissions i.e., double submission detection. To do this, a submission is compared against public domain data and a database of current and archived publications from publishing houses. As these services are centralized, the root of trust hinges solely on them. Therefore, double submissions can only be detected effectively as long as this third party is honest. Moreover, as the content of a submitted manuscript in a conference is not public, it is not possible to check concurrent submissions.

The nature of these problems is similar to "double spending" problem in cryptocurrencies. Cryptocurrencies handle double spending by enforcing a common ledger for all the miners to check against. This idea can be invoked in the conference systems too to detect double and concurrent submissions, i.e., we can enforce a common database of all publications from all the publishing houses. This database also includes the publications currently under review at all the conferences associated with the member publishing houses. Hence, whenever a manuscript is submitted to a conference, it can be compared with the list of manuscripts in this database to detect double and concurrent submissions. This effectively eliminates dependence on third party services for detecting double submission and also detects concurrent submission. However, the burden of maintaining huge database of publications makes this solution difficult to realize.

Acknowledgements. We would like to thank Vigneswaran R for his inputs towards the development of our system.

Appendix A CBC Transactions

Hyperledger Fabric supports chaincode execution through "*query*" and "*invoke*". A *query* is a chaincode execution which reads from the ledger but does not write into the ledger. Whereas, an *invoke* is capable of both, reading and writing. *invoke* transactions will be captured as transactions on blockchain. Here is the list of invoke and query transactions from our MaRSChain implementation:

- ***C*BC_Submit_Txn:**
peer chaincode invoke -o 127.0.0.1:7050 -C CBC_Channel -n CBC_CC -c '{"Args":["submit_paper", "author1","author2", "author3","attach_01"]}'
- ***C*BC_Reviewer_Assignment_Txn:**
peer chaincode invoke -o 127.0.0.1:7050 -C CBC_Channel -n CBC_CC -c '{"Args":["assign_reviewers", "paper_id","reviewer1", "reviewer2","reviewer3"]}'
- ***C*BC_Review_Txn:**
peer chaincode invoke -o 127.0.0.1:7050 -C CBC_Channel -n CBC_CC -c '{"Args":["submit_review", "reviewer1","paper_id","rating"]}'
- ***C*BC_Reviewes_Consolidate_Txn:**
peer chaincode invoke -o 127.0.0.1:7050 -C CBC_Channel -n CBC_CC -c '{"Args":["make_decision"]}'
- ***C*BC_Query_Manuscript_Txn:**
peer chaincode query -o 127.0.0.1:7050 -C CBC_Channel -n CBC_CC -c '{"Args": ["querySubmittedPaperInfo", "paper_id"]}'
- ***C*BC_Query_Manuscripts_List_Txn:**
peer chaincode query -o 127.0.0.1:7050 -C CBC_Channel -n CBC_CC -c '{"Args": ["queryAllPaperIDs"]}'
- ***C*BC_Query_Manuscript_Status_Txn:**
peer chaincode query -o 127.0.0.1:7050 -C CBC_Channel -n CBC_CC -c '{"Args": ["queryPaperStatus", "paper_id"]}'

Appendix B Endorsement Policy

The classical blockchain systems relied on *order-execute architecture*, where the ordered transactions are executed by the peers sequentially. This has a drawback of decreased throughput. The current Hyperledger Fabric (v1.0.0+) employs *execute-order-validate architecture* to parallelize the validation of transactions by the peers. A transaction is executed by the peers in parallel and the result of the execution is provided as an endorsement to the user. The user collects and sends all the endorsements to the committers through an orderer. The committers validate the endorsements based on an endorsement policy(m out of n signatures) before committing the transactions into the ledger. Our implementation assumes a total of 4 PCs(3t+1), tolerating 1 malicious PC. Hence, our endorsement policy mandates 3 out of 4 endorsements for any transaction. The endorsement policy from our implementation can be seen below:

5: CBC Endorsement Policy

```
var epolicy = {
identities: [
{ role: { name: "member", mspId: "PC1MSP" }},
{ role: { name: "member", mspId: "PC2MSP" }},
{ role: { name: "member", mspId: "PC3MSP" }},
{ role: { name: "member", mspId: "PC4MSP" }}
],
policy: {
"3-of": [{ "signed-by": 0 }, { "signed-by": 1 }, { "signed-by": 2 }, { "signed-by": 3 }]
}
};
```

References

1. Deep Shift- Technology Tipping Points and Societal Impact. http://www3.weforum.org/docs/WEF_GAC15_Technological_Tipping_Points_report_2015.pdf
2. EasyChair. http://easychair.org/
3. EDAS. https://edas.info/
4. HotCRP. https://hotcrp.com/
5. Hyperledger Fabric. https://www.hyperledger.org/projects/fabric
6. Hyperledger Use Case Inventory. https://wiki.hyperledger.org/groups/requirements/use-case-inventory
7. iChair. https://www.baigneres.net/ichair/
8. iThenticate. http://www.ithenticate.com/
9. Provenance. https://www.provenance.org/whitepaper
10. Ripple. https://ripple.com/
11. Similarity Check. https://www.crossref.org/services/similarity-check/
12. Aïmeur, E., Brassard, G., Gambs, S., Schönfeld, D.: P3ers: privacy-preserving peer review system. Trans. Data Privacy, pp. 553–578 (2012). http://dl.acm.org/citation.cfm?id=2423656.2423659

13. Arapinis, M., Bursuc, S., Ryan, M.: Privacy supporting cloud computing: confichair, a case study. In: Proceedings of the First International Conference on Principles of Security and Trust. pp. 89–108 (2012). https://doi.org/10.1007/978-3-642-28641-4_6
14. Gipp, B., Breitinger, C., Meuschke, N., Beel, J.: Cryptsubmit: introducing securely timestamped manuscript submission and peer review feedback using the blockchain. In: Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries. pp. 273–276 (2017). http://dl.acm.org/citation.cfm?id=3200334.3200370
15. Karl, W., Arthur, G.: Do you need a blockchain. https://eprint.iacr.org/2017/375.pdf
16. Nakmoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
17. Schulzrinne, H.: Double submissions: publishing misconduct or just effective dissemination? SIGCOMM Comput. Commun. Rev. pp. 40–42 (2009). https://doi.org/10.1145/1568613.1568622
18. Shafagh, H., Burkhalter, L., Hithnawi, A., Duquennoy, S.: Towards blockchain-based auditable storage and sharing of iot data. In: Proceedings of the 2017 on Cloud Computing Security Workshop, pp. 45–50 (2017). https://doi.org/10.1145/3140649.3140656