Blockchain Technologies

Dhiren Patel · Sukumar Nandi ·
B. K. Mishra · Deven Shah ·
Chirag N. Modi · Kamal Shah ·
Rajesh S. Bansode *Editors*

IC-BCT 2019

Proceedings of the International Conference on Blockchain Technology

Springer

Blockchain Technologies

This book series aims to provide details of blockchain implementation in technology and interdisciplinary fields such as Medical Science, Applied Mathematics, Environmental Science, Business Management, and Computer Science. It covers an in-depth knowledge of blockchain technology for advance and emerging future technologies. It focuses on the Magnitude: scope, scale & frequency, Risk: security, reliability trust, and accuracy, Time: latency & timelines, utilization and implementation details of blockchain technologies. While Bitcoin and cryptocurrency might have been the first widely known uses of blockchain technology, but today, it has far many applications. In fact, blockchain is revolutionizing almost every industry. Blockchain has emerged as a disruptive technology, which has not only laid the foundation for all crypto-currencies, but also provides beneficial solutions in other fields of technologies. The features of blockchain technology include decentralized and distributed secure ledgers, recording transactions across a peer-to-peer network, creating the potential to remove unintended errors by providing transparency as well as accountability. This could affect not only the finance technology (crypto-currencies) sector, but also other fields such as:

Crypto-economics Blockchain
Enterprise Blockchain
Blockchain Travel Industry
Embedded Privacy Blockchain
Blockchain Industry 4.0
Blockchain Smart Cities,
Blockchain Future technologies,
Blockchain Fake news Detection,
Blockchain Technology and It's Future Applications
Implications of Blockchain technology
Blockchain Privacy
Blockchain Mining and Use cases
Blockchain Network Applications
Blockchain Smart Contract
Blockchain Architecture
Blockchain Business Models
Blockchain Consensus
Bitcoin and Crypto currencies, and related fields

The initiatives in which the technology is used to distribute and trace the communication start point, provide and manage privacy, and create trustworthy environment, are just a few examples of the utility of blockchain technology, which also highlight the risks, such as privacy protection. Opinion on the utility of blockchain technology has a mixed conception. Some are enthusiastic; others believe that it is merely hyped. Blockchain has also entered the sphere of humanitarian and development aids e.g. supply chain management, digital identity, smart contracts and many more. This book series provides clear concepts and applications of Blockchain technology and invites experts from research centers, academia, industry and government to contribute to it.

If you are interested in contributing to this series, please contact msingh@endicott.ac.kr OR loyola.dsilva@springer.com

More information about this series at http://www.springer.com/series/16276

Dhiren Patel · Sukumar Nandi · B. K. Mishra ·
Deven Shah · Chirag N. Modi · Kamal Shah ·
Rajesh S. Bansode
Editors

IC-BCT 2019

Proceedings of the International Conference on Blockchain Technology

Springer

Editors
Dhiren Patel
Veermata Jijabai Technological Institute
Mumbai, Maharashtra, India

B. K. Mishra
Thakur College of Engineering
and Technology
Mumbai, Maharashtra, India

Chirag N. Modi
National Institute of Technology Goa
Goa, India

Rajesh S. Bansode
Thakur College of Engineering
and Technology
Mumbai, Maharashtra, India

Sukumar Nandi
Department of Computer Science
Engineering
Indian Institute of Technology Guwahati
Guwahati, Assam, India

Deven Shah
Thakur College of Engineering
and Technology
Mumbai, Maharashtra, India

Kamal Shah
Thakur College of Engineering
and Technology
Mumbai, Maharashtra, India

ISSN 2661-8338 ISSN 2661-8346 (electronic)
Blockchain Technologies
ISBN 978-981-15-4541-2 ISBN 978-981-15-4542-9 (eBook)
https://doi.org/10.1007/978-981-15-4542-9

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Organization Committee

General Chair
Dhiren Patel, VJTI, Mumbai, India

Organizing Program Chair
B. K. Mishra, TCET, Mumbai, India

Technical Program Chair
Sukumar Nandi, IIT Guwahati, India

Publicity Chair
Rajesh Bansode, TCET, Mumbai, India

Organizing Program Co-chair
Kamal Shah, TCET, Mumbai, India

Technical Program Co-chair
Deven Shah, TCET, Mumbai, India

Publicity Co-chair
Anil Vasoya, TCET, Mumbai, India

Workshop Tutorial Chair
Kamal Shah, TCET, Mumbai, India

Workshop Tutorial Co-chair
Mr. Pravin Patil, Member, IET Mumbai Local Network

Technical Expert Committee

Ramki Thurimella, CRISP, Colorado, USA
Stefan Junestrand, European Blockchain Observatory, Spain
Sung-Mo Steve Kang, JBSE, University of California, USA
Devesh C. Jinwala, NIT, Surat, India

Alka Mahajan, Nirma University, Ahmedabad, India
Tanish Zaveri, Nirma University, Ahmedabad, India
Mayank Lau, NASSCOM, India
Pablo Geovanny, Universidad de Las Americas, Quito, Ecuador
Chirag Modi, NIT, Goa, India
Rajib kumar Jena, Bajaj Auto Limited, Pune, India
Sanjay Gandhe, SITRC, Nashik, India
Vaishali Khairnar, TEC, Navi Mumbai, India
Mayank Agrawal, Telekom Innovation Laboratories, Israel
Subha Hari, Watson Supply Chain IBM, Bangalore, India
Dipesh Sharma, Watson Supply Chain IBM, Bangalore, India
Raghavendra Krishna Murthy, Watson Supply Chain IBM, Bangalore, India
Srinivas Kakaraparti, IBM, Bangalore, India
Uday Kothari, Blockchain DLT Geeks Pune, India
Syam Sunder, Requital Technology, Hyderabad, India
Darshit Parmar, MCME, Ahmedabad, India
Deepak Garg, SEAS, Greater Noida, India
Aman Soni, Global Audit and Assurance Trainee KNAV International Ltd.
Ankit Sharma, Upscale Consultancy Services Pvt. Ltd, Germany
Anupam Tiwari, Defence Intelligence Agency, Delhi, India
Gupta Boda, NABARD, Mumbai, India
N. M. Pandurang, Embiot Technologies R&D, Bangalore, India
Sachin Sadare, Digital Dojo Pvt. Ltd, Mumbai, India
Deepak Jain, Finlaw
Mohan Mishra, Finlaw
Tashish Rai Singhani, Sofocle Technologies, Noida, India
Nitash Juyal, Sofocle Technologies, Dubai
Ravi Awasti, Sofocle Technologies, Dubai

Program Committee

General Chair

Dr. Dhiren Patel, Director, VJTI, Mumbai, India

Organizing Program Chair

Dr. B. K. Mishra, Principal, TCET, Mumbai

Technical Program Chair

Dr. Sukumar Nandi, Professor and Head Centre for Linguistic Science and Technology, IIT Guwahati, India

Publicity Chair

Dr. Rajesh Bansode, Professor, TCET, Mumbai

Organizing Program Co-chair
Dr. Kamal Shah, Professor and Dean R&D, TCET, Mumbai

Technical Program Co-chair
Dr. Deven Shah, Professor and Vice Principal, TCET, Mumbai

Publicity Co-chair
Mr. Anil Vasoya, Assistant Professor, TCET, Mumbai

Workshop Tutorial Chair
Dr. Kamal Shah, Professor and Dean R&D, TCET, Mumbai

Technical Expert Committee

Dr. Ramki Thurimella, Director, Colorado Research Institute for Security and Privacy, University of Denver, Colorado, USA
Dr. Stefan Junestrand, Researcher—Media Company CEO/ Ph.D. Architect, European Blockchain Observatory, Spain
Dr. Sung-Mo Steve Kang, Professor, Jack Baskin School of Engineering, University of California, US
Dr. Devesh C. Jinwala, Professor, NIT, Surat
Alka Mahajan, Director, Nirma University, Ahmedabad, India
Dr. Tanish Zaveri, Professor, Nirma University, Ahmedabad, India
Mayank Lau, Principal Consultant/Researcher—under NASSCOM, India
Dr. Pablo Geovanny, Professor, Universidad de Las Americas, Quito, Ecuador
Dr. Chirag Modi, Assistant Professor, National Institute of Technology, Goa, India
Rajib Kumar Jena, General Manager, Bajaj Auto Limited, Pune
Dr. Sanjay T. Gandhe, Principal, Sandip Institute of Technology and Research Centre, Nashik
Dr. Vaishali Khairnar, Associate Professor, Terna Engineering College, Navi Mumbai, India
Dr. Mayank Agrawal, Associate Professor and Post-doctoral Researcher, Telekom Innovation Laboratories, Ben-Gurion, University of Negev, Beer-Sheva, Israel
Subha Hari, Performance Architect, Watson Supply Chain IBM, Bangalore
Dipesh Sharma, Project Executive, Watson Supply Chain IBM, Bangalore
Raghavendra Krishna Murthy, Delivery Lead, Watson Supply Chain IBM, Bangalore
Srinivas Kakaraparti, IBM Consultant, Sr. Manager, SaaS Customer Programs and Production Enhancements, IBM Watson Customer Engagement-Watson Supply Chain, India Software Laboratory IBM, Bangalore
Uday Kothari, Blockchain DLT Geeks Pune, India
Syam Sunder, Founder and CEO Requital Technology, Hyderabad, India, and Founder at Marlin Protocol

Darshit Parmar, Member of Blockchain Council/IBM Blockchain Essentials/CMO, M-Connect Media and Encrybit, Ahmedabad
Dr. Deepak Garg, Professor, Head—Computer Science Engineering, School of Engineering and Applied Science, Bennet University, Greater Noida
Aman Soni, Global Audit and Assurance Trainee KNAV International Ltd.
Aditya Nalge, Blockchain Engineer, Decentralized Finance Laboratories, Palo Alto, California, USA
Ankit Sharma, Director, Upscale Consultancy Services Pvt. Ltd, Riverstreams Communications Pvt. Ltd, Code Mentor for Mifos X, Munich, Bavaria, Germany
Anupam Tiwari, Researcher, InfoSec and Cyber Security Professional, Defence Intelligence Agency, Delhi
Gupta Boda, Chief Technology Advisor, NABARD, Mumbai
N. M. Pandurang, SSE, Technical Trainer, Embiot Technologies R&D, Bangalore, India
Sachin Sadare, Founder Director, Digital Dojo Pvt. Ltd, Mumbai
Deepak Jain, Consultant, Finlaw
Mohan Mishra, Consultant, Finlaw
Tashish Rai Singhani, Consultant, Sofocle Technologies, Noida
Nitash Juyal, Director, Sofocle Technologies
Dr. Ravi Awasti, Consultant, Sofocle Technologies, Dubai

Preface

This volume contains the proceedings of the International Conference on Blockchain Technology 2019, held in Mumbai, India, during 29–30 March 2019.

The International Conference on Blockchain Technology 2019 (IC-BCT 2019) was jointly organized by TCET and VJTI, Mumbai. The objective of the conference was to bring delegates to share new ideas, experiences and knowledge in Blockchain technology. Blockchain being a disruptive technology provides immense opportunities to the researchers and industry practitioners. Scalability, sustainability, security, besides consensus mechanisms, latency, limited privacy, storage constraints, wasted resources, etc., are various challenging areas in Blockchain. Also, several business case studies are devised to take advantage of Blockchain features such as immutability and verifiable ledger. A truly global platform to report latest research and development in Blockchain technology and showcase ideas linked to Blockchain use cases was extended to researchers, academia and industry practitioners through IC-BCT 2019.

Research papers in IC-BCT 2019 included theoretical as well as real-world case studies. IC-BCT 2019 received 60 submissions from five different countries including Bangladesh, India, Japan, South Korea and Spain. Based on the extensive review by 41 experts in the Program Committee, 15 full-length papers and four short-length papers were selected for presentation and inclusion in the proceedings. Full-length papers were worth the inclusion in the proceedings, whereas acceptance of short-length papers was based on the belief that the paper would contribute either in introducing new concepts or will have a high impact. It thus resulted in a highly competitive call and 26.78% acceptance rate for full technical papers.

Presentations by academic and industry experts in the proceedings focused on the fields of performance optimization, decentralization schemes, Blockchain-based applications, smart contracts and distributed ledgers.

In IC-BCT 2019, we had accompanying Blockchain workshop related to the basics of Bitcoin and Ethereum along with the case study. This allows technical exposure to young blood.

All the parties involved have contributed immensely to this international conference. We would like to thank Program Committee members and Technical

Expert Committee members for timely and in-depth review of the submitted papers. We are also thankful to all the members of the Organizing Committee. We also thank the EasyChair conference system, Springer, IET team with Bhushan Nemade and Neerajkumari Khairwal for proceedings preparation. We hope that the proceedings paves a way ahead for research in Blockchain technology.

Mumbai, India	Dhiren Patel
Guwahati, India	Sukumar Nandi
Mumbai, India	Deven Shah
Mumbai, India	Kamal Shah
March 2019	

Contents

Preserving Location Privacy Using Blockchain

Rishipal Yadav, Sumedh Nimkarde, Gaurav Jat, Udai Pratap Rao, and Dilay Parmar

Abstract With the advancement of technology and enhanced techniques of the global positioning system, the use of location-based services has significantly increased in the last decade. With the increase in the use of these services, there is also a rise in concern for the preservation of location privacy. There have been some cases where location data was disclosed, which even led to some serious crimes. Preservation of location privacy becomes a must in these situations. There are various techniques for preserving location privacy. Some use an anonymizer in between location-based services (LBS) and user, while other uses a distributed architecture for preserving location privacy. In this paper, a blockchain-based decentralized architecture for preserving location privacy is proposed. Earlier users had to trust either the anonymizer or the LBS for retrieving the query results, but with this proposed solution, advancement toward zero trust model would be possible.

Keywords Location privacy · Blockchain · Location-based services · Decentralization · Zero trust privacy model

1 Introduction

Technology has greatly simplified our lives. We have mobile applications for everything we want to do. There are applications that can tell us who is around us, what is happening around us by using our location. For better result retrievals and enhanced user experience, every kind of service which is provided online needs access to the user location. These services are known as 'location-based services' which continuously ask for user locations. There have been some instances in the past where adversaries tracked user location and used this information for malicious intent, sometimes even for criminal activity. So, a user must preserve his location information. While accessing location-based services, a user sends his location data

R. Yadav (✉) · S. Nimkarde · G. Jat · U. P. Rao · D. Parmar
Department of Computer Engineering, S. V. National Institute of Technology, Surat, Gujarat, India
e-mail: yadav.rishipal001@gmail.com

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_1

and query to a location-based service provider, and the service provider returns the result. The location needs to be preserved in this entire communication.

The rest of the paper is organized as follows. In Sect. 2, a brief overview of preserving location privacy techniques to date has been given. In Sect. 3, the proposed approach is presented. Section 4 concludes the paper and presents the future scope of the work.

2 Related Work

There has been much research going on in preserving location privacy. The existing defense mechanisms are based on either of two distinct architectures given for preserving location privacy [6]: (1) centralized architecture or TTP-based architecture, (2) decentralized architecture or TTP-free architecture.

In centralized architecture (TTP-based), there is an anonymizer between user and LBS provider which anonymizes the user data. Grutser and Grunwald [4] have vastly discussed in their paper about spatial cloaking, temporal cloaking, and interval cloaking. Mokbel et al. [13] presented Casper Cloaking. Gedik and Liu [3] suggested a clique scheme for forming ASR regions and preserving location data. Hilbert curves are the principal premise of Hilbert cloaking mechanism, as suggested by Kalnis et al. [9]. Authors in [1] presented an idea of mixing zones for alias formation to preserve the identity of user. Protecting privacy through dummy nodes has been proposed in [11, 12].

The decentralized architecture (TTP-free) is the one where the user directly requests service from the LBS provider. Now, for this approach, peer-to-peer spatial cloaking has been proposed by [2] . The concept of peer-to-peer spatial cloaking is the same as TTP-based spatial cloaking, but there is no third party involved, and users collaboratively work to send the queries to LBS. However, in the peer-to-peer spatial cloaking, there is no guarantee that the peers are trusted. This is just an assumption. What if these peers have malicious intent? To overcome this, authors in [5] proposed a trust-based approach known as CAST mechanism. Gupta and Rao [7] have proposed a hybrid model for mobile LBS using homomorphic encryption and Gaussian noise. Geometric transform techniques have also been used to hide the location coordinates by the authors in [8].

Now, with all these pre-existing solutions, new kinds of the solution in the decentralized architecture are being proposed. There are few solutions where blockchain is used to preserve privacy, but these solutions are application-specific. Authors in [14] presented a blockhain-based solution for preserving location privacy in crowdsensing systems. Kanza and Safra [10] presented a solution for preserving privacy, psuedonymity, and trust in ride-hailing system using blockchain.

To the best of our knowledge, not much work has been done in preserving location privacy using blockchain. Significant work has been done in preserving data privacy, but its applicability in preserving location privacy is in the nascent stage.

3 Proposed Approach

Figure 1 shows the block diagram of our proposed approach. Now, in the blockchain, there are predefined smart contracts deployed which are the basis of our model. A smart contract is just a predefined logic between two digital entities which is executed when they agree to the terms of the contract. This contract is written and cannot be modified as it will be stored in the blockchain. Smart contracts are written in such a way that these restrict the nodes to access specific data from the transaction and will allow the usage of certain data. The smart contract in this model which is to be deployed in the blockchain does the following two things—(1) contains functions which govern access rules to the certain data defined. (2) Restricts the identity access to the LBS.

The node which requests service, calls the function in the smart contract and the contract is executed. A transaction including node identity data and its location data is stored in the block. The function called in the smart contract is responsible for collecting this data from the node and stores it in the blockchain. Now, LBS, which is also a part of the blockchain, knows that a smart contract is executed. It will check the block, and because of smart contract functionality, it will only be able to access the location data and not the actual identity of the node. LBS takes the location data, and the query further fetches the result from the information database which it already has through API and sends the respective result to the address of the smart contract (smart contract is identified by an address in the blockchain). Once

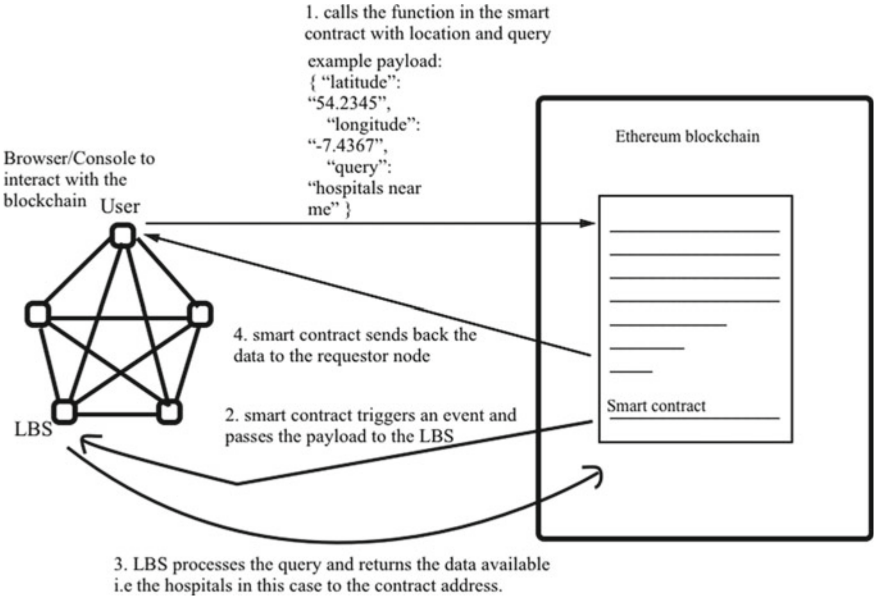
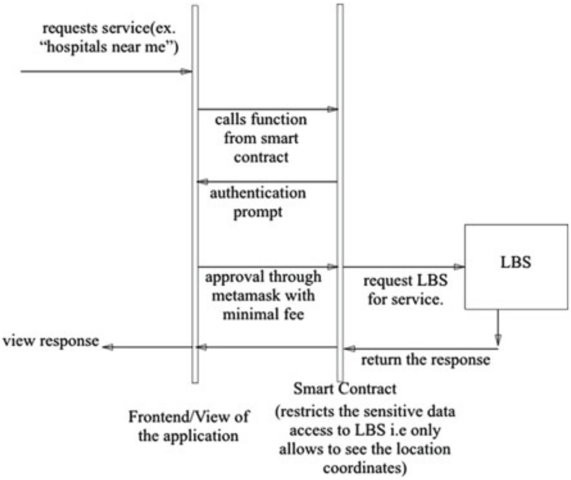

1. calls the function in the smart contract with location and query
example payload:
{ "latitude": "54.2345",
"longitude": "-7.4367",
"query": "hospitals near me" }
Browser/Console to interact with the blockchain
User
Ethereum blockchain
4. smart contract sends back the data to the requestor node
2. smart contract triggers an event and passes the payload to the LBS
LBS
Smart contract
3. LBS processes the query and returns the data available i.e the hospitals in this case to the contract address.

Fig. 1 Our proposed approach

Fig. 2 Sequence diagram


requests service(ex. "hospitals near me")
calls function from smart contract
authentication prompt
approval through metamask with minimal fee
request LBS for service.
LBS
view response
return the response
Smart Contract (restricts the sensitive data access to LBS i.e only allows to see the location coordinates)
Frontend/View of the application

the data is obtained, a smart contract will return it to the requestor node. Figure 2 shows the sequence diagram for this model. Ethereum blockchain can be used for the implementation of this approach.

3.1 How Privacy Is Preserved?

There are various attack surfaces where the location privacy of a user can be breached. It could be on the user side, or the LBS side. On the user side, privacy can be breached by the revelation of a user's real identity. In the LBS, it can get the location and identity information of a user. In this approach, privacy is preserved in the following ways:

In the blockchain, each node is identified by its public key instead of the real user identity. So, the pseudonym is used instead of the user name. It is difficult to identify the real user identity from this pseudonym as there is no correlation between a pseudonym (public key) and user identity.

Smart contracts act as the middlemen between the user and the LBS. They are the only one who knows the user's pseudonym and location query, but these are programmed logic. Hence, there is no malicious threat from these smart contracts.

LBS only knows the location and not the identity. It knows the address of the smart contract but cannot trace who is the query requestor. Here, the breaking of linkage between user identity and location information helps in protecting the privacy of users. By achieving sender anonymity, location privacy is also preserved.

4 Conclusions and Future Work

This paper proposed an idea for preserving location privacy using blockchain. In this model, advancement toward a zero trust model would be possible. This proposed approach is the first step toward the implementation of this work, and it can be extended by measuring its efficiency and comparing it with the existing models and further improving it.

References

1. Beresford AR, Stajano F (2003) Location privacy in pervasive computing. IEEE Pervasive Comput 1:46–55
2. Chow CY, Mokbel MF, Liu X (2006) A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In: Proceedings of the 14th annual ACM international symposium on advances in geographic information systems. ACM, pp 171–178
3. Gedik B, Liu L (2005) Location privacy in mobile systems: a personalized anonymization model. In: Proceedings of the 25th IEEE international conference on distributed computing systems (ICDCS 2005). IEEE, pp 620–629
4. Gruteser M, Grunwald D (2003) Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the 1st international conference on mobile systems, applications and services. ACM, pp 31–42
5. Gupta R, Rao UP (2017) Achieving location privacy through cast in location based services. J Commun Networks 19(3):239–249
6. Gupta R, Rao UP (2017) An exploration to location based service and its privacy preserving techniques: a survey. Wirel Pers Commun 96(2):1973–2007
7. Gupta R, Rao UP (2017) A hybrid location privacy solution for mobile lbs. Mob Inform Syst
8. Gupta R, Rao UP (2018) Privacy protection through hiding location coordinates using geometric transformation techniques in location-based services enabled mobiles. In: Cyber security: proceedings of CSI 2015. Springer, Berlin, pp 1–10
9. Kalnis P, Ghinita G, Mouratidis K, Papadias D (2007) Preventing location-based identity inference in anonymous spatial queries. IEEE Trans Knowl Data Eng 19(12):1719–1733
10. Kanza Y, Safra E (2018) Cryptotransport: blockchain-powered ride hailing while preserving privacy, pseudonymity and trust. In: Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems. ACM, pp 540–543
11. Kido H, Yanagisawa Y, Satoh T (2005) An anonymous communication technique using dummies for location-based services. In: Proceedings of international conference on pervasive services (ICPS'05). IEEE, pp 88–97
12. Miura K, Sato F (2013) Evaluation of a hybrid method of user location anonymization. In: Proceedings of eighth international conference on broadband and wireless computing, communication and applications (BWCCA). IEEE, pp 191–198
13. Mokbel MF, Chow CY, Aref WG (2006) The new casper: query processing for location services without compromising privacy. In: Proceedings of the 32nd international conference on very large data bases. VLDB Endowment, pp 763–774
14. Yang M, Zhu T, Liang K, Zhou W, Deng RH (2019) A blockchain-based location privacy-preserving crowdsensing system. Fut Gener Comput Syst 94:408–418

Decentralised Ecosystem for Journalism based on Blockchain

Vaibhav Agrawal, Aishwarya Agarwal, Shailja Shah, Dilay Parmar, and Udai Pratap Rao

Abstract Significant improvement and adaptation of the blockchain technology have led to various implementations and use cases, which utilise the power of decentralisation, immutability, scalability and secure the flow of data between two parties. Journalism is controlling a huge part of society and if manipulated can lead to disastrous results. In the recent past, it has been noted how journalism or media can influence the mass population and be used in favour or opposition of certain entities. The ecosystem we are proposing would tackle the problem of news authentication by a voting system. The traditional system used in journalism today uses a centralised platform. In this paper, we propose a novel decentralised platform to make the world of journalism democratic by giving power to the people to filter the authenticity of a news article through a majority consensus.

Keywords Blockchain · Ethereum · Journalism · Decentralised

1 Introduction

Based on a study by Data and Society and the Knight Foundation, "*The news is only what the majority wants to hear, its never the complete truth, and it may be false in some aspects. There is bias in the language. Its just the way media works*" [1].

V. Agrawal (✉) · A. Agarwal · S. Shah · D. Parmar · U. P. Rao
Department of Computer Engineering, S. V. National Institute of Technology, Surat, Gujarat, India
e-mail: vaibhav.a.cse@gmail.com

A. Agarwal
e-mail: aish.agarwal04@gmail.com

S. Shah
e-mail: shailjashah24@gmail.com

D. Parmar
e-mail: dilayparmar@gmail.com

U. P. Rao
e-mail: udaiprataprao@gmail.com

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_2

The main problem which needs to be addressed is making journalism transparent, thereby preventing the spread of fake news and ensuring authenticity of the literature for end readers. According to the 2018 Edelman Trust Barometer survey, 63% of people cannot distinguish good journalism from rumours, and 59% of people find it hard to distinguish whether news is coming from a respected news outlet or one that is not so trusted [2]. The fact that we now have access to millions of potential news sources is an incredibly positive thing for journalism as we can retrieve firsthand information from those directly involved in the news, making it more likely for the truth to be told. However, the lack of gatekeepers of mainstream media or journalism also means there is no consensus on who is telling the truth, and that means the situation is not improving. In addition to the difficulty of dealing with fake news, an even more prevalent issue is learning to deal with the social aspects of fake news such as the decrease in trust of the public regarding what to trust. With the decentralised news sharing platform, we aim to create far better signals for consumers to be able to know if a news is trustworthy and credible as it is made up by the people themselves, regardless of what other influential individuals may be saying. This essentially puts everyone at the same level. The goal is to provide a platform for authors with a strong will to reveal something anonymously and a proper crowd-based judging mechanism. It will help the readers gain true insights based on the rating of crowds rather than various altered stories, be it through mainstream journalism or social media.

The rest of the paper is organised as follows: In Sect. 2, the preliminaries are explained. Section 3 focuses on the proposed approach for the ecosystem, including the system architecture and the flow of system modules. Section 4 is concluding our proposed approach for this ecosystem.

2 Preliminaries

2.1 Journalism

The journalists and media houses, specifically the TV news is essentially reporting on oneself. It reports events in accordance with its own formats rather than understanding the facts and then communicating them in terms of the real-world scenario, considering the complexities and ambiguities associated with it [3]. During a time when there is clearly a declining trust in the news media, it is vital to create a platform to share news without any bias. Media is more prevalent today in our society than it ever was in our history, but journalism as it is today is failing to meet its expected standards of true accounts that are fact-based and validated by multiple sources. Mass media is corporate owned, and hence, most of it is scripted, spiced, and numbers driven [4]. This is the biggest disadvantage with journalism today. The public rarely gets a firsthand account of any news, i.e. direct access to events as they happened. Instead, they are exposed to rehashed second, or thirdhand accounts.

2.2 Blockchain

Blockchain is the technology used to update digital transaction records in the form of an incorruptible decentralised digital ledger in real-time [5, 6]. This allows maintenance of a permanent and tamper-proof record of transactional data [7]. Adding data to the chain and reviewing the data are possible by everyone, but changing the data on the blocks is not possible without 51% of the members agreeing to it which is almost computationally infeasible due to its proof of work mechanism [8, 9]. The technological advantages of blockchain are as follows:

- **Durability**—As they are distributed networks, there is no single point of failure. This distribution of risk among the whole blockchain makes it more durable than centralised systems.

- **Transparency**—Each node on the network maintains an identical copy of the whole blockchain. This level of transparency makes activities and operations highly visible, thereby reducing the need for trust [9].

- **Immutability**—Due to the consensus mechanism, there is a need for validation by other nodes and a way to trace the changes. This gives users the highest degree of confidence that the chain of data is accurate and unaltered [9].

- **Process Integrity**—As distributed protocols are executed as they are written because of the non-alteration property, users can trust the actions executed by the protocol. Moreover, there is no human intervention involved.

2.3 Miners

Adding transaction records to a public ledger containing the previous transactions is known as mining. The computation of the block hash is a difficult task and requires high computational power. Miners are those nodes that provide with the computational power to solve the mathematical problems of the proof of work (PoW). At every certain interval of time, they take a few of the transactions from the pending pool of transactions and start hashing. They are paid for in terms of digital currency as a return [10].

2.4 Ethereum

Ethereum is a platform for developing decentralised applications that run on smart contracts. These applications execute exactly as they are programmed without the possibility of third-party interference, service denial or censorship. Ethereums public blockchain enables developers to create decentralised applications without worrying about the infrastructure involved in creating the blockchain. It has its own digital currency ether associated with it for the transactions to occur [11, 12].

2.5 Smart Contracts

Smart contracts are built on the Ethereum platform. Smart contracts are essentially blocks of code or program snippets that are deployed on the nodes of the blockchain. They can be used to exchange anything of value such as money, shares, and property in a conflict-free way without the use of middlemen, i.e. banks in the case of money, brokers in the case of shares or property. They eliminate any third parties in a transaction. These transactions are traceable and irreversible. They must be deterministic, and therefore, each input should map to the same value or produce the same output every time. A non-deterministic smart contract implies that when it is triggered, it will result in random results being returned for every node on the network. Therefore, on its execution result, this would prevent the network from reaching a consensus. In addition, since smart contracts lie on the blockchain, they each have an address which is unique. A smart contract can be executed by using the address assigned to its transaction. Hence, it executes independently on each node in the network depending on the data in the transaction that triggered the node. A properly written smart contract should clearly list all the possibilities or outcomes of the contract. The smart contract of every node maintains the same state of variables, inputs, and outputs [6].

2.6 Ethereum Virtual Machine (EVM)

The EVM, or Ethereum Virtual Machine can solve any computational problem. Smart contracts are powered by EVM. Ethereum enables users to make their own operations and hence serves as a platform for decentralised blockchain applications. The EVM is used to read and execute the smart contract-specific programming language bytecode [11].

2.7 MetaMask

MetaMask is an extension available for Chrome, Firefox, and Brave which allows users to run decentralised applications (DApps) in their browser without running a full Ethereum node. DApps are essentially Ethereum decentralised applications. It includes a secure sign-in process with a friendly user interface that enables the proper management of identities on various sites and secure signature blockchain transactions. In essence, it is a self-hosted wallet to store, send, and receive ETH [13]. To use MetaMask, it must first be installed on a browser. After installation, an icon at the same level of the address bar would be visible. The icon then needs to be selected, and at this point, the user can proceed with logging in. Funding MetaMask with ether is done through the Ropsten Test Network. MetaMask does not support most synchronous Web3 API methods. Web3JS library and conditionals are used to manage the error states.

2.8 Web3JS

Web3JS is a group of modules each with different functionalities for the Ethereum ecosystem. It enables interaction between the user and a local or remote Ethereum node, with the help of an HTTP or an IPC connection. This is shown in Fig. 1

Blockchain applications can be developed with Ethereum using the following aspects:

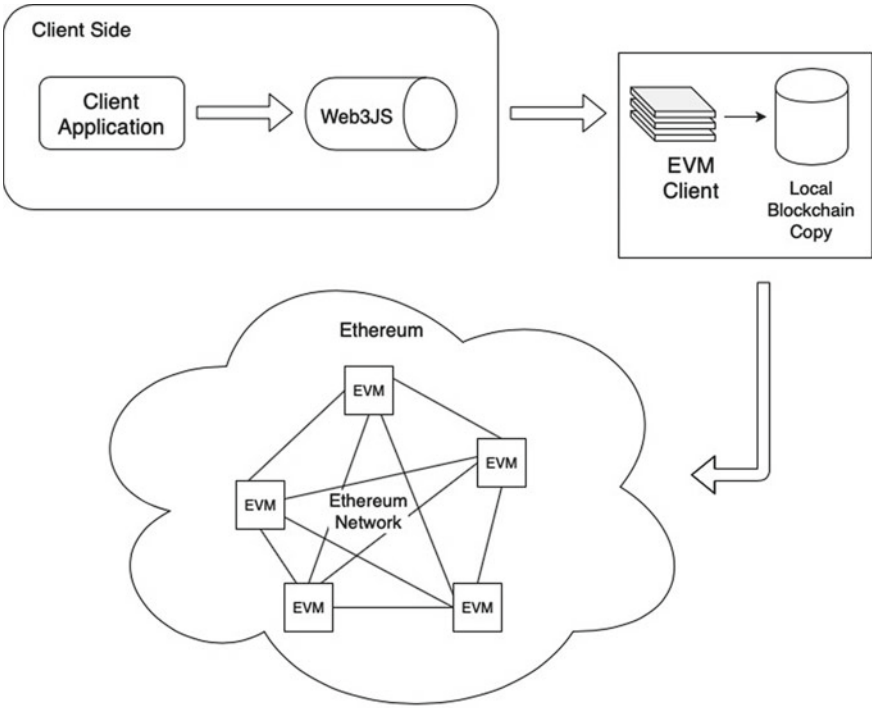


Fig. 1 Client Ethereum interaction using Web3JS

1. Solidity is used for developing smart contract code which is deployed on blockchain.
2. To develop clients to interact with blockchain to read and write data from the blockchain using smart contracts.

Web3JS allows the programmer to develop clients that will eventually interact with the Ethereum blockchain. It is also a library that provides user with the facility to transfer or transact the Ether between two different accounts. It also provides the facility to read/write from the smart contracts, or to create the smart contracts.

This communication takes place via JSON Remote Procedure Call protocol. Ethereum keeps a duplicate of the programs and data on the blockchain through a network of peer-to-peer nodes. JSON RPC is used to request an Ethereum node from Web3JS to read or write data to a blockchain network the same way jQuery is used to read or write data to a Web server by JSON API. [14].

To read data from smart contracts using Web3JS, the following things are necessary:

1. The smart contract, a JavaScript representation to interact with
2. A way to call the function on the smart contract during the process of reading the data

To retrieve the JavaScript representation of the smart contract, the function used is web3.eth.Contract(). Two arguments are passed to this function for the smart contract application binary interface (ABI) and for the smart contract address.

2.9 Blockchain Wallet

A software program which allows users to buy, sell, and check balance for their digital currency (or assets) is known as a blockchain wallet. It is used by users for exchange of cryptocurrencies, such as bitcoin and ether.

Blockchain wallets do not function as the way traditional wallets do, in that all the transactions are stored in the blockchain. Transactions in the blockchain involve the assigning of the cryptocurrency to the receiver's wallet address. In addition, these transactions are stored in the distributed ledger [11].

Blockchain transactions are based on asymmetric cryptography which uses two types of keys:

- Private key
- Public key

The public and private keys are non-identical pairs of large numbers, of which the public key may be shared and the private key may not be shared. The private key helps in uniquely identifying the user in the network and can be considered as a personal, digital signature and becomes invalid if the signature is altered.

For example, Person 1 sends some digital currency to Person 2; Person 2 is assigned as the owner of that currency to the address of Person 2's wallet. For verification of the transaction, Person 2's private key should match the public key assigned to the currency by Person 1. Once the transaction is verified, the transaction is combined with other transactions to form a block. The block is then added to the ledger, i.e. it cannot be altered, and the changes are reflected in the blockchain wallet.

3 Proposed Approach

3.1 Traditional Approach

Figure 2 is the centralised system for any news/thought sharing application available today. The organisation/developer of the application has complete control over user registration, data management, content management, etc. This system has certain drawbacks:

1. Single point of failure
2. Ownership of data
3. Concentrated rules and regulation
4. Organisation is always correct
5. All profit going to the central authority
6. Fake news being publicised without any measures taken.

3.2 Proposed Approach

The proposed approach involves deploying a smart contract on the Ethereum blockchain rather than deploying a code to a central server. The smart contract would

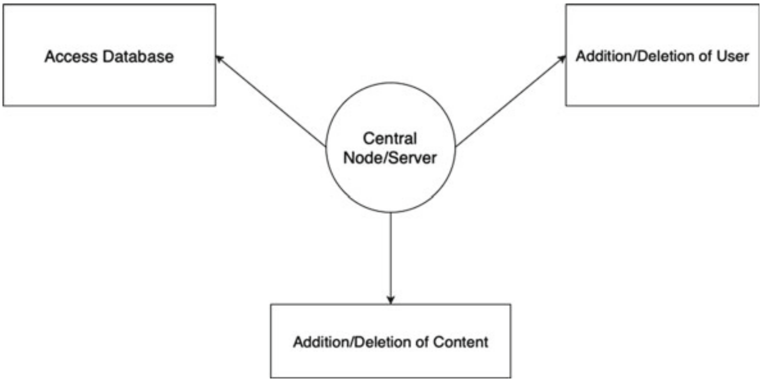


Fig. 2 Traditional system

include the required functionality of login, article creation, upvote/downvote module, and reputation module. When the smart contract is deployed, it would behave as one code being replicated and run on each node independently.

Fake news can be tackled by the upvote/downvote system by the community. Since people will be charged and they risk losing cryptocurrency, they will not be downplaying news without solid reason to. If they do, the local community who knows about the news being fake will downvote it. Crossing a threshold will mark it fake. In this way, the system will tackle the spreading of fake news.

3.3 System Architecture

System architecture essentially consists of the flow of the whole system and is as shown in Fig. 3:

1. Provide the user with user interface to interact with the smart contract.
2. The user would be able to login/logout with the Ethereum credentials.
3. Addition of news/article by any journalist would cost certain amount of ether.
4. This transaction would be broadcasted to every node on the network.

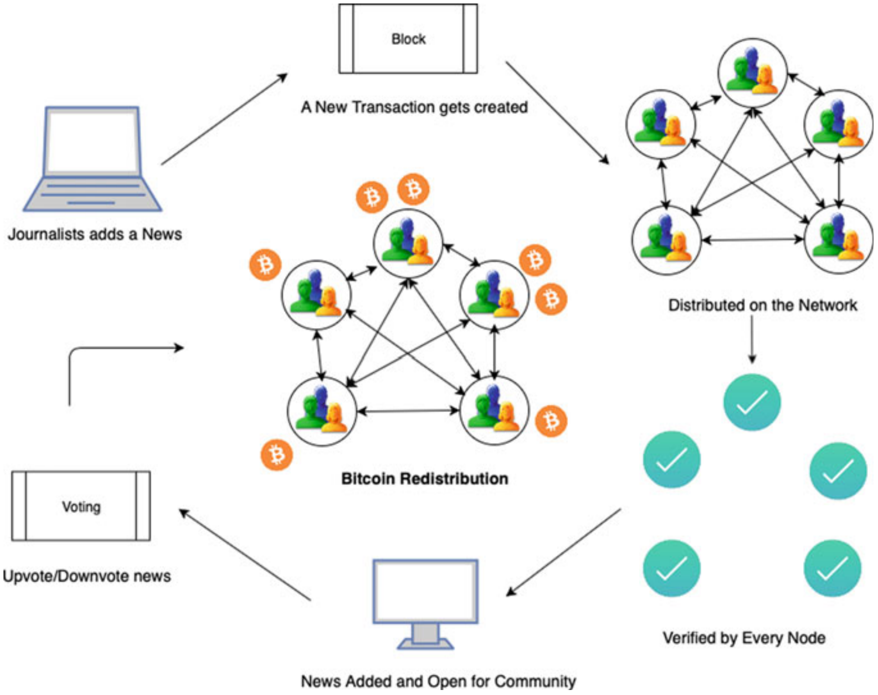


Fig. 3 System architecture

5. As soon as consensus is achieved, the news would become visible to all the users.
6. Users will be allowed to upvote/downvote the authenticity of news/article with a little cost.
7. Rewards the users based on their authenticity. The system can be divided into the following modules.

3.4 Post Management Module

This module will be used by the user to post a news on the platform. The sequence flow of events that occurs when a post is posted is shown in Fig. 4:

1. User creates a post message request.
2. Browser interacts with the Web3JS which is injected via MetaMask.

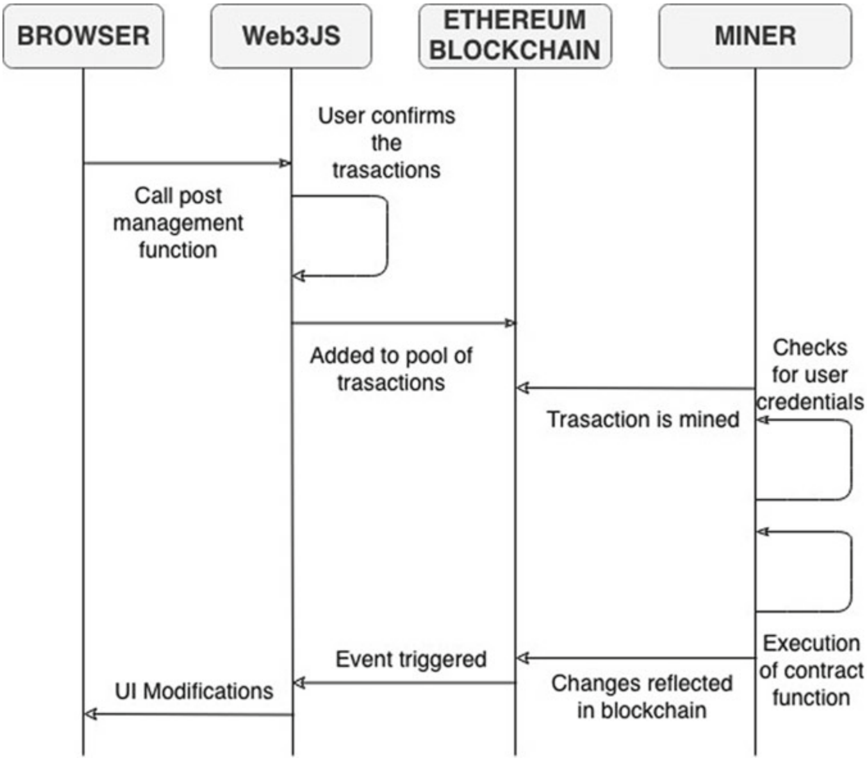

BROWSER
Web3JS
ETHEREUM BLOCKCHAIN
MINER
User confirms the trasactions
Call post management function
Added to pool of trasactions
Checks for user credentials
Trasaction is mined
Execution of contract function
Event triggered
Changes reflected in blockchain
UI Modifications

Fig. 4 Sequence diagram: post module

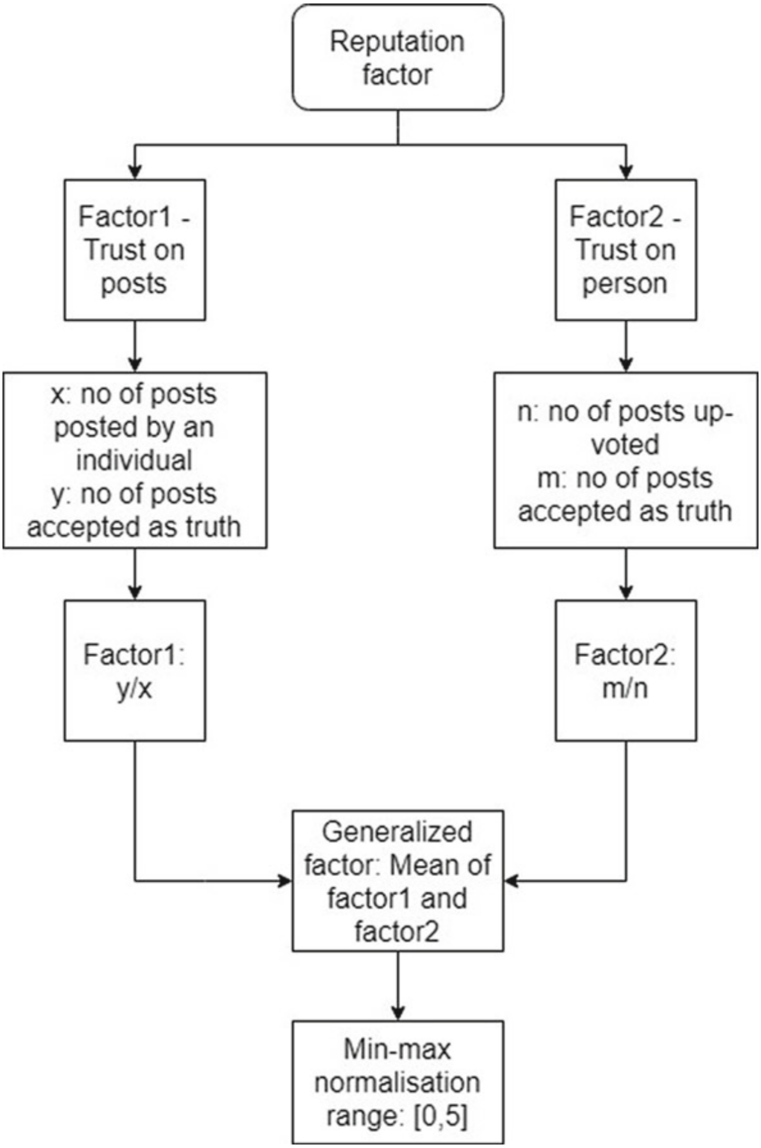

Reputation factor
Factor1 - Trust on posts
Factor2 - Trust on person
x: no of posts posted by an individual
y: no of posts accepted as truth
n: no of posts up-voted
m: no of posts accepted as truth
Factor1: y/x
Factor2: m/n
Generalized factor: Mean of factor1 and factor2
Min-max normalisation range: [0,5]

Fig. 5 Reputation factor calculation

3. Web3JS sends a transaction for execution of the post message function.
4. User signs the transaction using MetaMask, and the transaction is added to the pool of pending transactions.

3.5 Reputation Module

The sequence flow of events that occur when the reputation module is executed for each user after a post is accepted or declined by the system is shown in Fig. 5:

1. After all users upvote/downvote, reputation factor module is called.
2. Two reputation factors have to be calculated. Factor 1 represents the trust on an individuals post by calculating the probability of correctness of posts posted by an individual based on history of an individual on ecosystem. It is defined as the number of posts posted by an individual accepted. Factor 2 represents the trust on an individual by calculating the probability of acceptance of posts as truthful that were upvoted by an individual. It is defined as number of posts accepted for which the user upvoted.

$$\text{Reputation Factor} = (\text{Factor1} + \text{Factor2})/2 \tag{1}$$

$$\text{Normalised Reputaion Factor} = 5 * (\text{Reputation Factor}) \tag{2}$$

3. The generalised factor is calculated by taking the mean of both the factors and then normalised using min–max normalisation. The final range is [0,5].
4. The reputation factor for all users is updated in Ethereum blockchain.

A user is identified by their Ethereum credentials rather than the wallet addresses, so user having multiple wallet addresses cannot falsify the reputation factor as all the addresses will be mapped to a single user.

Also, if the attacker uses various Ethereum accounts to falsify the reputation factor, they risk losing all the ether they used to falsify the news. This would be a lot of ether as it is difficult to beat the community and ensure 51% advantage in attackers favour. Hence, using reputation factor is secure.

3.6 Upvote/Downvote Module

The sequence flow of events happening when user upvotes/downvotes a particular news post is explained in Fig. 6:

1. User upvotes/downvotes.
2. The interacting module, i.e. Web3JS, informs the blockchain about the transaction and initiates it.
3. As the transaction is added to the pool, the miner selects it and solves the mathematical problem while checking for user login status.
4. The block gets added to the Ethereum blockchain.

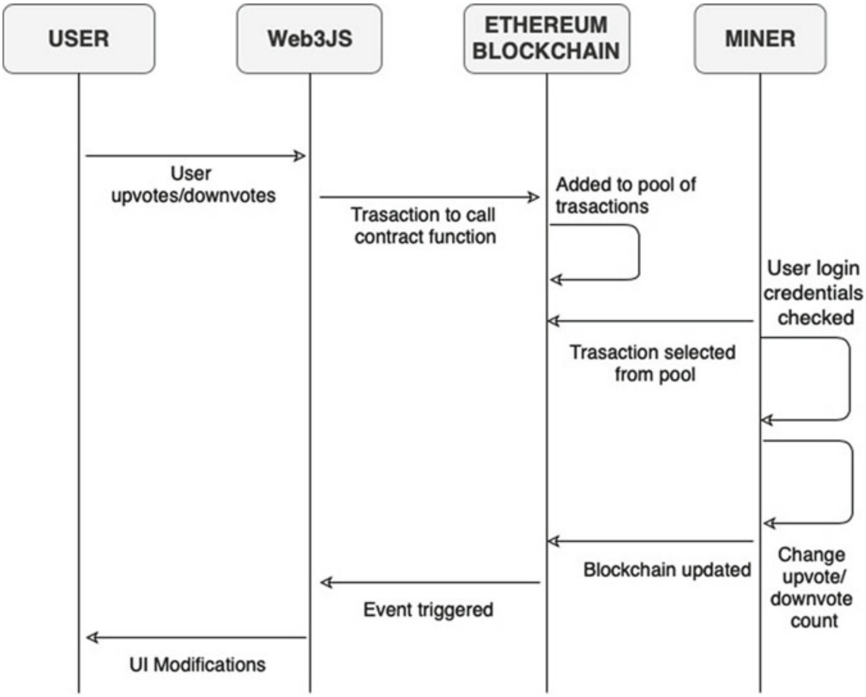

USER
Web3JS
ETHEREUM BLOCKCHAIN
MINER
User upvotes/downvotes
Trasaction to call contract function
Added to pool of trasactions
User login credentials checked
Trasaction selected from pool
Change upvote/ downvote count
Blockchain updated
Event triggered
UI Modifications

Fig. 6 Sequence diagram: upvote/downvote module

4 Conclusion

Through this application, we aim to create a novel solution to prevent news suppression. This would be the first step towards building a democratic ecosystem for journalism. The architecture described can be further implemented to achieve the goal of a democratic ecosystem. It would give power to the community rather than the media houses for writing and publishing stories. Readers would be able to gain true insight on news rather than the fabricated news that is often published these days to benefit the media houses. This would eliminate any dependency between journalists and publishers or journalists and ad agencies. Journalists would be able to exercise true freedom of press and be able to deliver and circulate content without the interference of a central authority. By using a blockchain ecosystem, no central authority would be able to filter articles based on personal bias, creating a platform where the right questions will be asked, and the right voices will be heard.

References

1. Surprise! Young People Don't Trust the Media. https://www.usatoday.com/story/college/2017/03/21/surprise-young-people-dont-trust-the-media/37429287/
2. People trust platforms less, trust journalism more, study says. https://www.cbsnews.com/news/2018-edelman-trust-barometer-survey-richard-edelman-interview/
3. Why does journalism need blockchain technology? Civil. https://blog.joincivil.com/why-does-journalism-need-blockchain-technology-6db1b4ff84ba
4. What is the state of journalism today? https://www.quora.com/What-is-the-state-ofjournalism-today
5. Devetsikiotis M, Christidis K (2016) Blockchain and smart contracts for the internet of things. IEEE Access, vol 4, pp 2292–2302
6. Ethereum. https://github.com/ethereum/wiki/wiki/White-Paper
7. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system
8. Why does journalism need blockchain technology? https://blog.joincivil.com/whydoes-journalism-need-blockchain-technology-6db1b4ff84ba
9. Zheng Z, Xie S, Dai H, Chen X (2017) An overview of blockchain technology: architecture, consensus, and future trends. In: 6th IEEE international conference on big data. IEEE Press, New York, pp 557–559
10. Blockchain Consulting Service. http://www.zoaks.com/blockchain
11. Account Management. http://ethdocs.org/en/latest/account-management.htmlcreating-anaccount
12. What is ethereum? https://www.coursehero.com/file/27003507/What-is-Ethereum-%CE%93%C3%87%C3%B6-Ethereum-Homestead-0pdf/
13. A complete guide to building ethereum dApps with metamask. https://medium.com/crowdbotics/building-ethereum-dapps-with-meta-mask-9bd0685dfd57
14. Intro to Web3.js ethereum blockchain developer crash course. http://www.dappuniversity.com/articles/web3-js-intro

NEWSTRADCOIN: A Blockchain Based Privacy Preserving Secure NEWS Trading Network

Anik Islam, Md. Fazlul Kader, Md. Mofijul Islam, and Soo Young Shin

Abstract To stay up to date with world issues and cutting-edge technologies, the newspaper plays a crucial role. However, collecting news is not a very easy task. Currently, news publishers are collecting news from their correspondents through social networks, email, phone call, fax, etc. and sometimes they buy news from the agencies. However, the existing news sharing networks may not provide security for data integrity and any third party may obstruct the regular flow of news sharing. Moreover, the existing news schemes are very vulnerable in case of disclosing the identity. Therefore, a universal platform is needed in the era of globalization where anyone can share and trade news from anywhere in the world securely, without the interference of third-party, and without disclosing the identity of an individual. Recently, blockchain has gained popularity because of its security mechanism over data, identity, etc. Blockchain enables a distributed way of managing transactions where each participant of the network holds the same copy of the transactions. Therefore, with the help of pseudonymity, fault-tolerance, immutability, and the distributed structure of blockchain, a scheme (termed as NEWSTRADCOIN) is presented in this paper in which not only news can be shared securely but also anyone can earn money by selling news. The proposed NEWSTRADCOIN can provide a universal platform where publishers can directly collect news from news-gatherers in a secure way by

A. Islam · S. Y. Shin (✉)
Department of IT Convergence Engineering, Kumoh National Institute of Technology,
Gumi 39177, South Korea
e-mail: wdragon@kumoh.ac.kr

A. Islam
e-mail: anik.islam@kumoh.ac.kr

Md. F. Kader
Department of Electrical and Electronic Engineering, University of Chittagong,
Chattogram, Bangladesh
e-mail: f.kader@cu.ac.bd

Md. M. Islam
Department of CSE, University of Dhaka, Dhaka, Bangladesh
e-mail: akash.cse.du@gmail.com

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_3

maintaining data integrity, without experiencing the interference of a third-party, and without disclosing the identity of the news gatherer and publishers.

Keywords Blockchain · News trading · Privacy · Smart contract

1 Introduction

Blockchain is a distributed database that is replicated and shared among the peers of a network. The blockchain concept was first introduced by Satoshi Nakamoto, a person or a group who used this name as a pseudonym [1]. Any individual from the network can check and verify the ledgers [2]. Valid transactions are stored in the block and each block holds the hash of the previous block. Thus, the process of creating a chain of blocks is called blockchain. To append data in blockchain, data experiences distributed consensus mechanism and after getting acceptance, data is added in blockchain [3]. Blockchain can be classified into two ways based on the accessibility of users, such as (1) public blockchain and (2) private blockchain [4]. In public blockchain, anyone can add data and anyone can participate in the consensus mechanism. On the contrary, in private blockchain, only a predefined list of users can access the network and only owner can participate in the consensus process. A mechanism is introduced in blockchain named "Smart Contract" in which if the user knows each other, they can interchange information or money without the requirement of any third-party [5]. Blockchain has solved trust issues in the distributed system because no party can tamper with the data in the network [6]. In blockchain, a pair of private/public keys is used by the participants [2]. Participants uses their public key as their identity and exercise their private key to sign transactions [7]. Therefore, asymmetric cryptography is used in blockchain to maintain its security in the network.

Newspapers play a very significant role in keeping individuals up-to-date with daily events. The newspaper is an indispensable part of life for some people who are unable to start their morning without reading one. News-gathering requires a lot of effort. Every newspaper has its own correspondents who work in different areas of focus and try to collect news regarding their area [8]. Big newspaper publishers have offices in different cities and some have offices in different countries as well. Correspondents, who work in these offices, share news reports very quickly via phone or email [9]. However, those publishers, who do not have a large network of news correspondents, have to buy news from news agencies. Moreover, there is a strong possibility that the collected news may experience modification by the third party or any third party may interfere with the independence of the news gatherer. Furthermore, any third party may hinder the process of sharing the collected news. News collection from those countries; that are engaged in the civil war, massacres or surrounded by terrorists, is very risky and sometimes, it is near impossible for foreign countries to obtain news on the actual situation. Sometimes, news gatherers have to collect news by risking their lives in these countries [10] and sometimes in their own countries [11–13]. A global scheme is required where news gatherers can share their news while maintaining safety and data integrity. In addition, news publishers can collect news directly from the correspondent instead of going to the agencies without facing any obstacles.

A cloud-based news sharing scheme is introduced in [14] and another user-generated content based news-gathering system is proposed in [15]. Both of them have proposed news sharing schemes only for emergencies. Their proposed scheme cannot solve the aforementioned issues like data integrity, secure data transfer, and pseudonymity. Moreover, the proposed scheme in [14] is centralized, which may pose a threat of third party interference. None of these studies considered blockchain in their proposed scheme. The pseudonymity and immutability features of blockchain can help news-gatherers share news fearlessly. Moreover, because of the distributed structure of blockchain, anyone can collect and share news from anywhere without paying a third party. Therefore, by considering the above features of blockchain; in this paper, a blockchain-based news trading scheme (termed as NEWSTRADCOIN) is proposed in which user can share news securely and also earn money; a topic which has not been explored yet to the best of our knowledge. The major contributions of this paper are compiled as follows.

- The proposed scheme ensures data integrity, no third party interference, and the pseudonymity of news sellers so that they can gather and trade news dauntlessly.
- A new file-sharing scheme is proposed where sellers can share news files directly with buyers without disclosing their identity.
- An optimized cost function algorithm is applied so that sellers can share news file most reasonably.

The remaining sections of this paper are organized as follows: Sect. 2 illustrates the system model of NEWSTRADCOIN. The different components of NEWSTRADCOIN are also discussed in this section. In Sect. 3, different characteristics of NEWSTRADCOIN are discussed in details. Performance comparison among NEWSTRADCOIN and other existing models is demonstrated in Sect. 4. Finally, Sect. 5 draws a conclusion from this paper with future research directions.

2 Proposed NEWSTRADCOIN

A news trading scheme is devised which uses the blockchain concept to make it distributed and secure while maintaining the pseudonymity of the news seller. The proposed NEWSTRADCOIN is a new way of trading news. It has the following features:

- News gatherers can sell news dauntlessly by maintaining their pseudonymity.
- News can be shared using a distributed system so that business can avoid system failure. A user does not have to be bound to a centralized authority, and the system can be cheaper.

- News can be shared securely so that no one can cheat each other and news can be tamperproof.
- Create a marketplace where anyone can make money by selling news or working as a file miner.

The proposed system model is provided in Fig. 1. In the proposed scheme, sellers add their news in blockchain with a price. A secure channel between the user and NEWSTRADCOIN is assumed. Before buying news, buyers perform queries in the blockchain and check the news. If the buyers prefer any news, they can buy the news through NEWSTRADCOIN. When sellers receive money from buyers through the proper channel, sellers send news documents to buyers using NEWSTRADCOIN. The five major components of NEWSTRADCOIN are news seller (NWS), news buyer (NB), news miners (NM), file miners (FM), and block cop (BC). The description of these entities are provided below:

- **News Seller (NWS)**—In this entity, users are the sellers of news. They collect and sell news via NEWSTRADCOIN.
- **News Buyer (NB)**—In this entity, users are the buyers. They can search for news to buy based on their preferred categories. Buyers can also act as a seller and seller can act as a buyer and vice versa.
- **News Miners (NM)**—In this entity, miners mine news which is posted from NWS. After validating block by its hash, they assign news in the blocks in the blockchain network. In blockchain, there is a ledger which contains metadata of

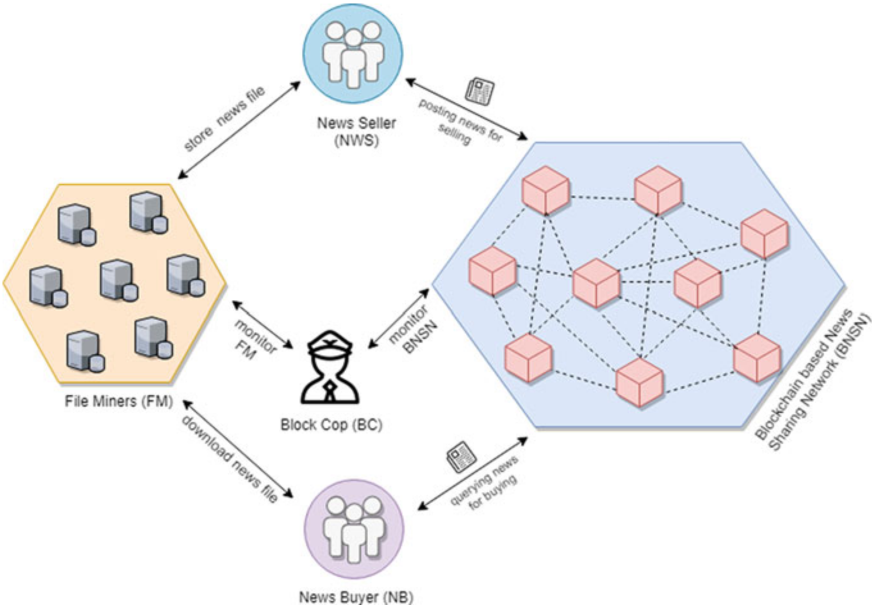


Fig. 1 System model of the proposed NEWSTRADCOIN

the news (i.e., headline and some description) and it is distributed among the participant nodes. NWS and NB can also lend their resources for mining.

- **File Miners (FM)**—In this entity, miners mine files which are shared from NWS. When users from NB buy news from NWS, NWS obtains assistance from FM to share that news file with the corresponding NB by maintaining file security and the sellers' pseudonymity.
- **Block Cop (BC)**—This entity protects cheating. Suppose, a user sells news and takes money from the buyer. If the seller does not send news files to the buyer then BC investigates the case using transaction history which is stored in the blockchain ledgers. Finally, BC imposes high penalties to the wrongdoers.

3 Features of NEWSTRADCOIN

3.1 Distributed and Secure

NEWSTRADCOIN is a distributed marketplace where there is no central authority or any third party to maintain transactions. In NEWSTRADCOIN, when a user from NWS posts news to sale, it experiences multiple steps before adding it to the blockchain. First, NM validates the transaction based on the hash generated from the block. It also checks for duplicate entries. Multiple miners engage in this process. The miner who validates the block first, obtains money as a reward, as shown in Fig. 2. Those who do not obtain a reward, store the news for future queries. Moreover, the privacy of the news is very important. It is the sellers' decision how much or to what extent they want to expose to buyers. Moreover, blockchain maintains the immutability of the data using its technology. Hence, news tampering is not possible in NEWSTRADCOIN. NM not only mines news blocks but also gathers data for the queries which are executed by buyers. When buyers make queries in the system to find expected news, NM engages in the data gathering process. The miner who provides the result first obtains the reward. If a buyer does not obtain any useful data, the buyer again makes queries using the same text in the system within a certain time range t_R, as shown in Algorithm 1. NEWSTRADCOIN maintains the security of data with the help of blockchain. Once the news is posted on the network, no one can alter the content. If anyone tries to change the news then that change has to be validated by the other members of the network. After purchasing the news, the original content is shared in a secure channel with the assistance of encryption that not only ensures the integrity of the data but also precludes the interference of a third party over the content.

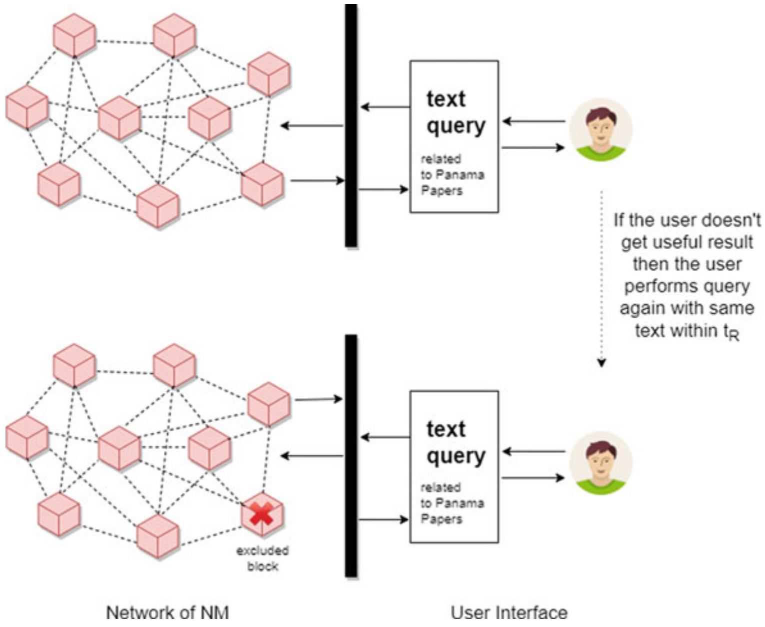


Fig. 2 Selection of miner for performing query

3.2 Secure File Transfer

The basic structure of blockchain does not support storing files because it can increase the size of a block. Blockchain generally contains the transaction history or text-based data. Hence, NEWSTRADCOIN does not store files in the NM. Instead, it uses another network of miners who rent their resources to hold news files. If a user wants to share a file with a buyer, that the buyer has already purchased, then that user can easily transfer the corresponding news file using the proposed file sharing scheme, as shown in Fig. 3. Suppose, Toma purchases news from Belal. Then, Belal sends the file to Toma, as described in Fig. 3. Firstly, Belal uploads the file to the system with a budget to store the file in FM. Miner collector (MC) receives that request from Belal. The main tasks of MC are as follows: (1) holding the miners' information with their latest space limit and costs per unit of space, (2) storing files in miners' workspace and retrieving files from miners', (3) handling user requests of storing and retrieving files, (4) splitting files with the help of the file chopper (FC) which is a module for splitting files, (5) joining files with the help of Kintsugi box (KB) which is a module for joining a file that has been split, and (6) posting transactions history to blockchain. Secondly, MC splits files into chunks so that MC can store those chunks in different miners. As MC contains the information of the miners, including their storage space limit and file storing charge, MC calculates how many chunks are needed for storing. When a user sends a request to store files, he/she sends a file along with his/her budget. Before splitting the file, MC first calculates

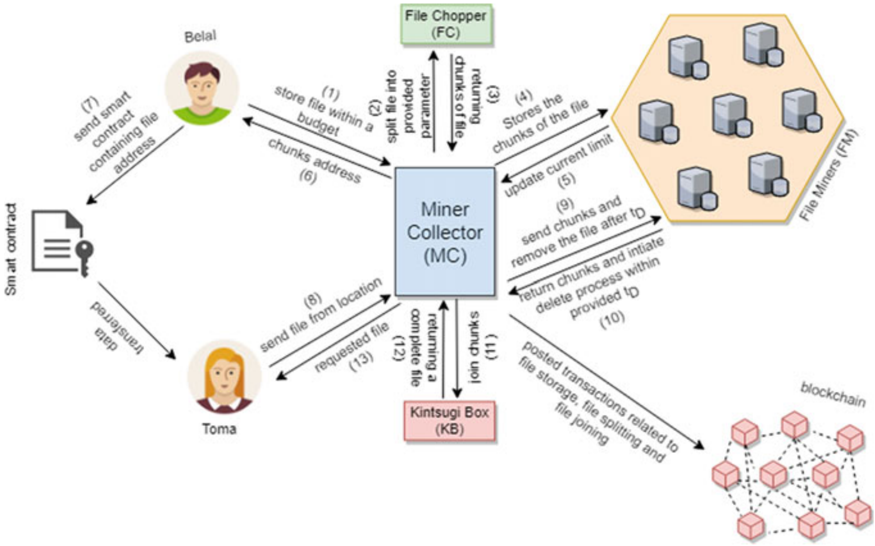

Belal
File Chopper (FC)
(7) send smart contract containing file address
(1) store file within a budget
(2) split file into provided parameter
(3) returning chunks of file
(4) Stores the chunks of the file
update current limit (5)
chunks address (6)
Miner Collector (MC)
File Miners (FM)
Sm art contract
transferred data
(9) send chunks and remove the file after tD
return chunks and intiate delete process within provided tD (10)
(8) send file from location
requested file (13)
(12) returning a com plete file
(11) join chunks
Kintsugi Box (KB)
Toma
posted transactions related to file storage, file splitting and file joining
blockchain

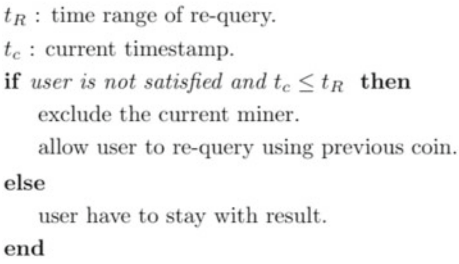
Fig. 3 Proposed secure file sharing model

the unit budget price. Let b is the unit budget price. If S is the file size and B is the budget then

$$b = \frac{B}{S} \tag{1}$$

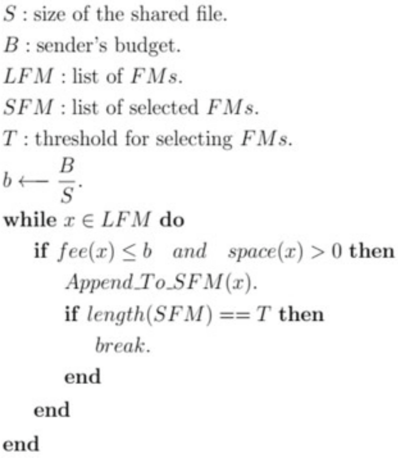
After that, MC selects FMs based on the cost per unit space calculated by $fee(x)$ and available space, as shown in Algorithm 2. According to Algorithm 2, MC first calculate b using Eq. (1) which is the budget per byte. Here, the system is considering the byte as a file unit and miners' charge is also per byte. Subsequently, MC goes through all of the FMs and checks the following: (i) whether the fee provided by anyone is equal or less than b and (ii) whether they have free space or not. If MC finds any FM, then MC adds that FM to the list of selected candidates SFM for storing files. There is a threshold T to show how many miners can participate in the file storing process.

Algorithm 1. Query execution



```
$t_R$ : time range of re-query.
$t_c$ : current timestamp.
if $user\ is\ not\ satisfied\ and\ t_c \leq t_R$ then
    exclude the current miner.
    allow user to re-query using previous coin.
else
    user have to stay with result.
end
```

Algorithm 2. Selection of FMs



```
$S$ : size of the shared file.
$B$ : sender's budget.
$LFM$ : list of $FMs$.
$SFM$ : list of selected $FMs$.
$T$ : threshold for selecting $FMs$.
$b \longleftarrow \frac{B}{S}$.
while $x \in LFM$ do
    if $fee(x) \leq b \quad and \quad space(x) > 0$ then
        $Append\_To\_SFM(x)$.
        if $length(SFM) == T$ then
            $break$.
        end
    end
end
```

If SFM reaches T then MC stops looking for miners and starts selecting how many miners can participate in the final process. In the final selection process, MC first sorts the selected FMs into ascending order based on the fee. Let SFM be the list of selected FMs. Therefore, the list of sorted FMs in ascending order is denoted by $SSFM_{FEE}$, where the subscript FEE represents the cost per unit space. After that, it continuously selects a FM and checks for space, as shown in Algorithm 3. When MC generates the final list of FMs, it sends that list to FC. As FC receives a list of FMs, it starts its chopping process, as shown in Algorithm 4. As the list is already sorted based on the fee, the cheaper miner gets priority for renting space. FC stores FM with an allotted size. There is a chance that the last chunk is smaller than the space of FM. Therefore, for the last chunk, space deduction is not needed. When the process is finished, FC returns the chunks with FMs and their allotted size. As MC gets a response from FC, MC sends chunks to the selected FMs. Finally, MC responds to Belal with the addresses of chunks. Before uploading a file, Belal encrypts the file by applying an asymmetric key algorithm. Hence, every chunk of the file is also encrypted. Let F be the file that Belal wants to share, K_{pub}^{rec} is the receivers public key. If F_E is the encrypted file then

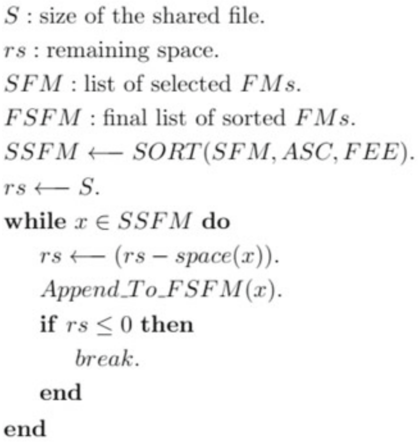
$$F_E = E_{K_{pub}^{rec}}(F, K_{pub}^{rec})$$

To access the file, Toma has to decrypt using her private key. As Belal obtains addresses of chunks from MC, he creates a smart contract containing the tuple is created:

$$\left\langle K_{pub}^{rec}, \left\{FC_{a_0}, FC_{a_1}, FC_{a_2}, \ldots, FC_{a_n}\right\}, R_a\right\rangle$$

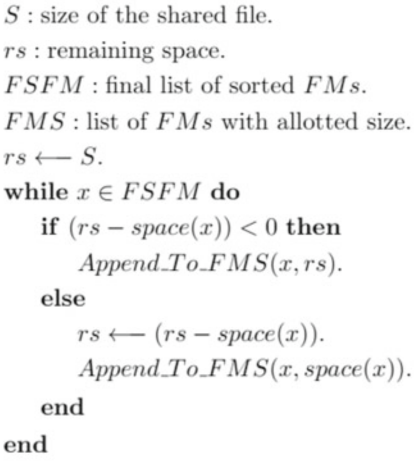
Here, R_a is the receiver address and FC_{a_n} is the address of n^{th} $(n = 0, 1, 2, \ldots)$ chunk of the file where n is the total number of chunks address.

Algorithm 3. Selection of final FMs



```
$S$ : size of the shared file.
$rs$ : remaining space.
$SFM$ : list of selected $FMs$.
$FSFM$ : final list of sorted $FMs$.
$SSFM \longleftarrow SORT(SFM, ASC, FEE)$.
$rs \longleftarrow S$.
while $x \in SSFM$ do
    $rs \longleftarrow (rs - space(x))$.
    $Append\_To\_FSFM(x)$.
    if $rs \leq 0$ then
        $break$.
    end
end
```

Algorithm 4. Selection of space for chunks



```
$S$ : size of the shared file.
$rs$ : remaining space.
$FSFM$ : final list of sorted $FMs$.
$FMS$ : list of $FMs$ with allotted size.
$rs \longleftarrow S$.
while $x \in FSFM$ do
    if $(rs - space(x)) < 0$ then
        $Append\_To\_FMS(x, rs)$.
    else
        $rs \longleftarrow (rs - space(x))$.
        $Append\_To\_FMS(x, space(x))$.
    end
end
```

When Belal finishes creating the smart contract, it is sent to Toma. As Toma obtains the smart contract from Belal, the addresses of chunks are sent to MC to collect the chunks and combine the chunks into a file. When MC obtains the address, MC collects chunks from the addresses and initiates a timer in the FM to delete chunks after a certain amount of time t_D. After passing t_D, FM deletes chunks from their storage. MC sends the chunks to Kintsugi Box (KB). The main task of KB is to join the chunks into a file. When KB finishes joining, KB returns the file to MC. After that, MC returns the file to Toma. Let K_{pri}^{rec} be the private key of the receiver. Finally, Toma decrypts the file. If F_D is the decrypted file then:

$$F_D = D_{K_{pri}^{rec}}(F_E, K_{pri}^{rec})$$

3.3 Pseudonymous

One of the major benefits of this proposed scheme is users' pseudonymity. Instead of giving personal information, the proposed scheme uses blockchain's pseudonymity techniques. In blockchain, asymmetric encryption is adopted to maintain its security. Here, public keys are used as a signature to identify a user. Instead of linking real-life information to transactions, the user signature is linked with the transaction. Every user in this system has a unique signature which they use as an identity.

NEWSTRADCOIN generates a signature using a timestamp t, nonce n, a random text T from a user, and a salt hash h. Let $S = Sig(t,\ n,\ T,\ h)$ is the signature of a user. The system generates S for a new buyer/seller to maintain transactions without revealing their identity.

3.4 Trustable

NEWSTRADCOIN is a trustable marketplace. NEWSTRADCOIN contains the fraud detection mechanism on the NM and FM. If a buyer reports a case against a seller that the seller did not send the file after getting money, then BC investigates the case. When buyer transfer money to the seller, there is a transaction history in the blockchain. When a seller sends a smart contract to a buyer and that buyer joins and decrypts that file in NEWSTRADCOIN, there is two transaction history in the blockchain. BC goes through the transaction history and checks the abnormality of transactions. If BC finds anything fishy, BC notifies the seller to solve the issue within a deadline. If the seller does not solve the issue with the buyer then BC imposes a high penalty on the seller and return the money back to the buyer. Moreover, if any FM fails to provide service after getting the fee from hoster, BC also imposes a penalty on FM. This creates trust in the system and prevents peer to involve any illegal activities.

3.5 Profitable

NEWSTRADCOIN provides a profitable marketplace. There is a lot of opportunity for earning money. First of all, anyone can collect and sell with very few charges. As sellers are pseudonymous, they can sell news dauntlessly. Moreover, NEWSTRADCOIN contains earning options for non-sellers. In NM, anyone can earn money by mining and generating data based on a query from a buyer. In FM, anyone can earn money by renting their storage space. Sellers can use their space to transfer files to buyers which helps them to remain pseudonymous. Moreover, NEWSTRADCOIN selects cheaper space providers for sellers within a seller's budget. Let T_p be the total price, if n is the total number of selected FMs and fm is the file miner then:

$$T_p = \sum_{i=0}^{n} (as_i \times fee(fm_i))$$

where $T_p \leq B$. Here, B is the budget of a seller and as_i is the allotted space for the i^{th} fm. The proposed system is able to optimize the price so that seller can store their file for a cheap price.

Table 1 Performance comparison between NEWSTRADCOIN and existing models

Features	Proposed Schemes		
	Kumar et al. [14]	Zhang et al. [15]	NEWSTRADCOIN
Mess sharing	√	√	√
Pseudonymity	×	×	√
Immutability	×	×	√
Secure transfer	×	×	√
Distributed	×	√	√

4 Performance Comparison

The performance comparison of NEWSTRADCOIN with Kumar et al. [14] and Zhang et al. [15] is demonstrated in Table 1. Here, "Mess sharing" means any user can share news which is supported by Kumar et al. [14], Zhang et al. [15], and NEWSTRADCOIN. Pseudonymity means anyone can share news without disclosing their identity which is only supported by NEWSTRADCOIN. Immutability means no one can alter the news after sharing which is only supported by NEWSTRADCOIN. Secure transfer means the transfer of content in a secure channel which is only supported by NEWSTRADCOIN. Distributed means there is no central authority in the system which is supported by both Zhang et al. [15] and NEWSTRADCOIN.

5 Conclusions and Future Work

In this paper, we proposed a secure and distributed news trading scheme exploiting the pseudonymity, immutability, and distributed mechanism of blockchain. In the proposed scheme, news-gatherers sell news to buyers while maintaining data integrity and without exposing their identity. Moreover, after the deal, sellers can maintain this pseudonymity while sharing necessary documents with buyers in the cheapest way. Therefore, our proposed scheme can easily be adopted by any news collection chain as a private network for collecting news, with little or no modification. Information regarding implementation along with detailed numerical results is kept for the future extension of this paper. Furthermore, the reputation system of buyers and sellers for managing the authenticity of content and deals, and a detailed plan for managing the wallet for easy payment can be incorporated, which can be a subject for future studies.

Acknowledgements This work was supported by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2018R1A6A1A03024003).

References

1. Islam A, Uddin MB, Kader MF, Shin SY (2018) Blockchain based secure data handover scheme in non-orthogonal multiple access. In: 2018 4th International Conference on Wireless and Telematics (ICWT), Nusa Dua, pp 1–5
2. Islam A, Shin SY (2019) BUAV: A blockchain based secure UAV-assisted data acquisition scheme in Internet of Things. J Commun Networks 21(5):491–502. https://doi.org/10.1109/JCN.2019.000050
3. Islam A, Shin SY (2019) BUS: A blockchain-enabled data acquisition scheme with the assistance of uav swarm in internet of things. IEEE Access 7:103231–103249. https://doi.org/10.1109/ACCESS.2019.2930774
4. Savelyev A (2017) Copyright in the blockchain era: Promises and challenges. Comput Law & Security Rev
5. Li X, Jiang P, Chen T, Luo X, Wen Q (2017) A survey on the security of blockchain systems. Future Generation Comput Syst
6. Islam A, Shin SY (2019) BHMUS: Blockchain based secure outdoor health monitoring scheme using uav in smart city. In: 2019 7th International Conference on Information and Communication Technology (ICoICT), Kuala Lumpur, Malaysia, pp 1–6. https://doi.org/10.1109/icoict.2019.8835373
7. Islam A, Kader MF, Shin SY (2019) BSSSQS: A blockchain-based smart and secured scheme for question sharing in the smart education system. J inf commun convergence eng 17(3):174–184. https://doi.org/10.6109/JICCE.2019.17.3.174
8. Wikipedia (2018) Newspaper production process. https://en.wikipedia.org/wiki/Newspaper_production_process
9. E. Online Newspapers| how they are published and edited| contents of a newspaper|. http://www.englishonline.at/media/newspapers/newspapers.htm
10. Montgomery SJ (2018) White house correspondent april ryan says reporters are getting death threats
11. SAUSA TRS (2018) Times reporter still getting death threats
12. Khan MI (2017) Where reporters face beatings, threats and death
13. Ibrahim A (2018) Mfwa condemns threats on life of journalist
14. Kumar M, Rayyan M, Kumar P, Rawat S (2016) Design and development of a cloud based news sharing mobile application. In: 2016 Second international innovative applications of computational intelligence on power, energy and controls with their impact on humanity (CIPECH), pp 217–221
15. Zhang Q, Wang J, Zeng Z (2016) Design and implementation of ugc-oriented news gathering system server-side for emergencies. In: 2016 First IEEE international conference on computer communication and the internet (ICCCI), pp 228–232

Food Traceability System Using Blockchain and QR Code

Nilesh Mishra, Sagar Mistry, Santosh Choudhary, Sumukh Kudu, and Rupesh Mishra

Abstract Recently, many food scandals broke out one after another in India and people are appalled. After these intimidated incidents, people are now more concerned about food safety. These issues not only harm consumer's health but also it debilitates their trust in food markets. Since the current food logistics pattern is not meeting the need and demand for the food market, building a secure and reliable food traceability system has become a necessity. Tracing food supply chain is the process of tracking the movement of a particular food item in the entire process. This paper covers the blockchain- and QR code-based food traceability system, merits and demerits of decentralized systems, and finally, the building process of the proposed system. The proposed system will provide traceability, transparency, efficiency, reliability, and security through all the stages of a food supply chain. Distributed ledgers and decentralized systems play a key role in building this application as its key features are immutability, transparency, consensus, disintermediation and distributed ledgers, and smart contracts.

Keywords Blockchain · Distributed ledgers · Smart contract · Traceability system

N. Mishra (✉) · S. Mistry · S. Choudhary · S. Kudu · R. Mishra
Department of Computer Engineering, St. Francis Institute of Technology, Mumbai, India
e-mail: mishranilesh012@gmail.com

S. Mistry
e-mail: msagar879@gmail.com

S. Choudhary
e-mail: santosh.sc53@gmail.com

S. Kudu
e-mail: sumukh.kudu@gmail.com

R. Mishra
e-mail: rrupesh.mishra@gmail.com

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_4

1 Introduction

India is known for its agricultural sector. Nowadays, food safety has become a most important issue for all of us. Are we aware of the food we consume? Is it safe or not? Can we trace back the food product to its originating farm? In the food supply chain, a lot of food products are found and among these food products in the supply chain, many food frauds take place like food tampering, adulteration, etc. Every one in ten people is affected by consuming contaminated food worldwide, of which the highest risk is with children, especially under the age of five according to WHO [1]. Hence, tracking the journey of food has become essential. Traceability means the ability to trace the food product instantly in the entire lifecycle from its origin through every stage in its journey to the consumer. The consumer can acquire a detailed journey of the food product by just scanning the QR code, which increases transparency and trust for food markets.

This paper covers the use of blockchain to implement a decentralized food tracing and tracking system. The paper is more focused on the following points:

1. Proposal of blockchain-based food traceability system that uses "permissioned blockchain."
2. Review of existing blockchain framework suited for constructing blockchain-based food traceability system [2].

2 Related Work

Along with living standards improved, food spices and additives result in varied manufactured food in the market. Some ignorant manufacturers that want to lower the cost and increase yield may lead to more uncertainty about food safety. Since food safety scandals continued to happen, food traceability has been highlighted as an important measure to get rid of the impact on the industry. In terms of current existing supply chain technologies, fundamental technologies for tracking physical goods have been around for years, such as bar codes, radio frequency identification (RFID) tags, and other data collecting sensors. Several other researchers consider the application of advanced technology, especially RFID technology, in supply chain. Sari builded a simulation model for supply chain firm to find out under what kinds of conditions the investing in RFID technology is more beneficial for the firm. The study results depicted that using RFID technology in supply chain will provide more benefits when the collaboration among supply chain participants is more intensive [3].

Wang et al. proposed a rule-based decision support system to fulfill the real-time monitoring of agri-food products during their distribution process. Based on the information transmitted by sensor-RFID equipment from the refrigerated containers, this system calculated the remaining value and shelf-life time of agri-food products in transmission [4]. Ustundaga and Tanyasb presented a simulation model to obtain the expected profits of using an RFID-based system in supply chain by calculating the performance increase in efficiency, security, accuracy, and visibility [5].

Proposed by Vatalik Buterin in 2014, Ethereum is a permissionless blockchain network optimized for smart contracts that uses its own crypto currency called Ether. Ethereum supports various functions, including smart contracts, decentralized transactions. It is believed that Ethereum is the first application of decentralized autonomous organizations. Ethereum provides an infrastructure to blockchain with a fully developed Turing complete programming language that provides an environment to create smart contracts by writing the logic with less code. This allows users to create systems they want, as well as other applications that are not related to crypto currency [6].

3 Introduction to Blockchain Technology

3.1 Basic Concepts of Blockchain

Blockchain consist of record as block associated with each other using cryptography. Each block in the blockchain contains data, hash, and hash of the previous block [7]. It provides a feature where data is saved in a distributed ledger in such a way that it cannot be changed or erased. This feature of blockchain makes it immutable. Tampering is difficult in blockchain as each block is connected to the previous block. Blockchain can be classified as public blockchain and private blockchain. Public blockchain provides access to anybody without any restriction, on the other hand, private blockchain has a restriction where only specific people can read and interact with it. Blockchain also provides the feature of building a decentralized application where you can build or add blockchain to your personal application.

3.2 Distributed Ledger

Distributed ledger is like a database that is replicated, synchronized, and distributed within the members or nodes of the network [8]. The distributed ledger consists of records of exchange of data in the network. Distributed ledger is independent of third-party administration functionality like traditional databases [9]. The advantage of the distributed ledger is the absence of central authority. Distributed ledger technology could significantly change the supply chain structure, making it more reliable. Distributed ledger technology application can change or replace the manual and inefficient work in the industry of supply chain.

3.3 Smart Contract

Smart contract is similar to a real-world contracts. The only significant difference is that they are completely digital. Smart contracts are immutable modular programs that are integrated into the blockchain system.

3.3.1 Smart Contract in Food Traceability System

Smart contract can enhance transparency in supply chain by recording these sources of goods by storing information such as date, location, and quality on a blockchain. The origin of the product can be easily verified, which will provide assurance to manufacturing that their raw materials are coming from reliable sources and the consumer has more confidence that they are purchasing a legitimate product by allowing the digital form of verification to be created by the blockchain. The smart contract can also help with transparent authorization across the network parties which will help to easily verify that other parties have the requisite certification to carry out their duties for which reliability can be recorded and managed on a blockchain as well, which will allow supply chain managers to make a more informed decision when selecting suppliers. And spur suppliers to work hard to maintain a good track record. Smart contract can provide traceability within the supply chain by tracking the inventory at every stage along the way from its raw material source to end-user delivery.

3.4 Ethereum

Ethereum is public blockchain such as distributed computing system containing smart contract features. The centralized approach has single entity control that can lead to single point failure, making the system more vulnerable to attacks. Single point failure is not possible in Ethereum due to its architecture. It is run by a lot of people all over the world so it cannot go offline. Ethereum uses peer-to-peer architecture. Ethereum network is protected by decentralized network and cryptography [10].

3.5 Consensus

Consensus mechanism is the rule or algorithm where all the members should agree on certain terms and conditions that would be standard for all nodes on the network. Some of the consensus algorithms are as follows:

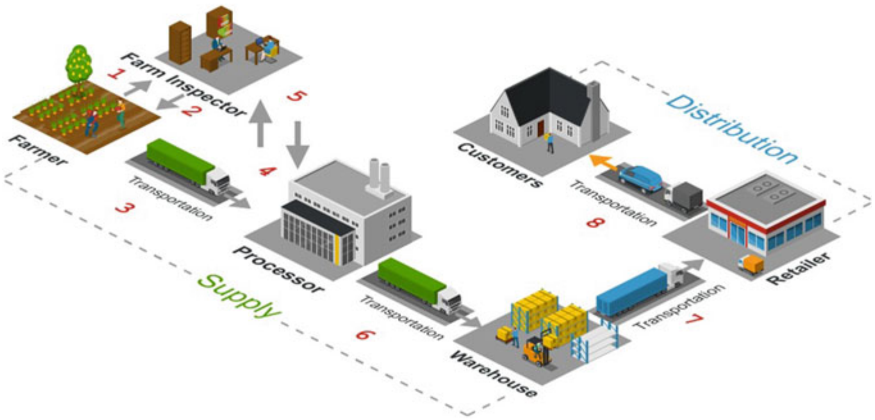


Fig. 1 Flow diagram of food traceability system

3.5.1 Proof-of-Stake

In proof-of-stake (PoS), the block producers are called validators instead of miners. Validators must provide a deposit or stake in order to participate in the process of block creation. In proof-of-stake, validators are chosen based on some selection algorithm that takes their stake into account. Once the validator is selected, they have the right to create a block. The other validators are not wasting energy doing any computation work since they are not selected.

3.5.2 Proof-of-Authority

The proof-of-authority (PoA) has the independent pre-selected authorized validators. Validators can validate the PoA-based network, block, and transaction. The validators can have authorities to insert the transaction in the blocks and do not require to monitor their computer system. Validators are service authorities who secure the network and seal the blocks (Fig. 1; Table 1).

4 Proposed System Methodology

4.1 Various Roles Involved in Food Traceability System

4.1.1 Farmer

Farmers will cultivate the crops and request farm inspector for inspections.

Table 1 Significant difference between PoS and PoA

Parameters	Proof-of-stake (PoS)	Proof-of-authority (PoA)
Network type	Public Private (Permissioned)	Private (Permissioned)
Transaction scalability	Medium	High
Transaction finality	Economic	Immediate
Token needed	Yes	No

4.1.2 Farm Inspector

Farm inspectors are responsible for inspecting farms and updating the information like crop family, type of seeds, and fertilizers used for growing the crops.

Farm inspector will visit the farm and do the necessary inspections. After inspecting the farm, the farm inspector will generate the detailed report of the inspection and keep it as a record for future purpose.

4.1.3 Processor

Processors are the organizations who process raw food materials by maintaining them at particular temperature and humidity and make it ready for packaging and to sell into markets. Processor adds the information like quantity, temperature, packaging date and time, processor name, and processor address.

Processors will send quotations to the farmers. The farmers will accept the quotation if it is feasible for the farmers to sell it to the processor.

4.1.4 Distributor

Distributors accept orders from retailers and deliver food product to them. If product is not sufficient in the warehouse, then distributor places order to processors to fill their warehouse.

4.1.5 Retailer

Retailers will sell the products to the actual end-user, i.e., to the consumers.

4.2 Overall Working of the Food Traceability System

Figure 2 shows the overall working of the proposed system.

All details shared between participants on the network will be verified using smart contracts. First, the farmer will register himself into the system. After cultivating the crops, farmer will request batch creation to admin. Admin will create the batch with new batch id and will allocate farm inspector based on nearby location for the inspection to the requesting farmer and will update the blocks. Farm inspector will inspect the crops and update and store the details such as crop id, fertilizers used, types of seeds, estimated date of expiry, and farm address on the blockchain which will be hashed data. After inspection stage, farmer will request for selling his crops to the processor with crop details like quantity, crop types, estimated selling price, etc. Respective processors who want to buy the crops will send quotation to the farmer. Farmer will select the best price suitable for him/her and even proposed system will recommend the best price to the farmer. Crops will be transported to the processor with crop details such as crop id, quantity, and farm inspector details. The processor will verify the details provided by the farmer with a farm inspector with smart contract and if details are verified, the processor will release the payment to the farmer. The

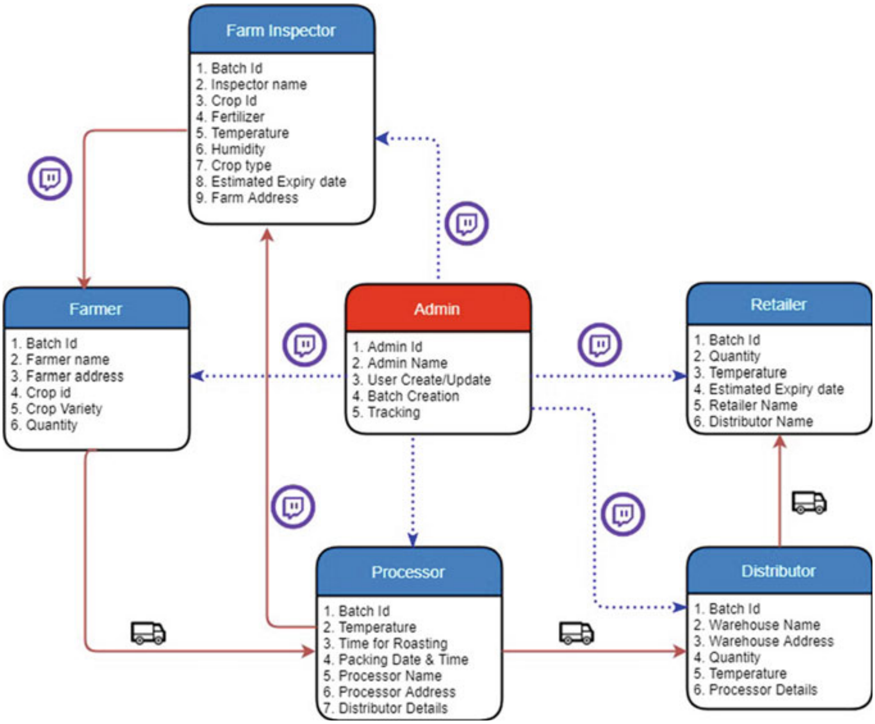


Fig. 2 System workflow diagram

processor will maintain the temperature, humidity, will roast the coffee, and update the details on the blockchain with packaging date and time, estimated expiry date, and the batch id. Batch will be transported to the warehouse by the distributor and all quality parameters will be verified with a smart contract with temperature, humidity, quantity, etc. Finally, the product will be transported to the retailer where he/she can trace all the required details about the product.

All data on the system will be hash data which is immutable and provides openness which increases the trust of the consumer.

5 Advantages and Disadvantages of Food Traceability System Using Blockchain and QR Code

5.1 Advantages

5.1.1 Tracking and Traceability Management

Food supply chain traceability builds a kind of information chain which would provide the information on food safety, processing of food, sales of the food, customer information, etc. Blockchain system makes the supply chain transparent and open; so, tracking becomes easy. Defective product could be easily caught within the supply chain with the support of tracking and transparency in the system [11].

5.1.2 Fake Products Can Be Caught

Applying the QR code allows the traceability of the product. The manual operation is not needed so the mistakes caused by human factors are avoided. The members in the blockchain cannot modify the data, which will increase the safety and quality of the product [11].

5.2 Disadvantages

5.2.1 High Cost

The system requirements for running the blockchain are high. Establishing of a traceability system needs huge investment in types of equipment and updating old equipment [11].

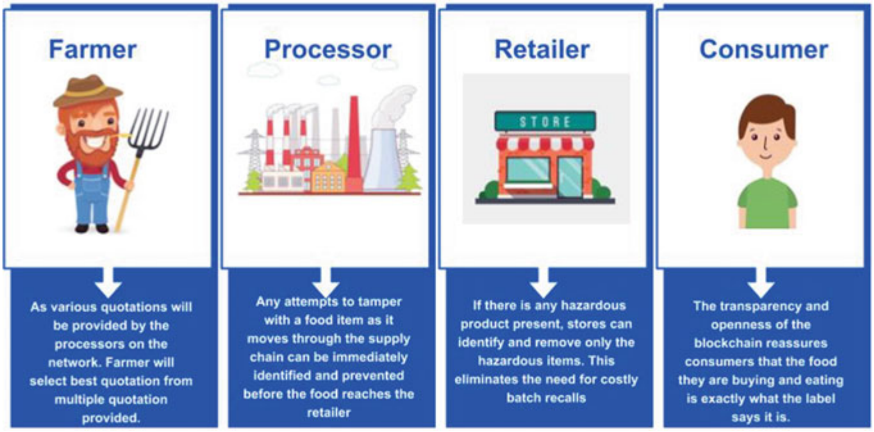

Farmer
Processor
Retailer
Consumer
STORE
As various quotations will be provided by the processors on the network. Farmer will select best quotation from multiple quotation provided.
Any attempts to tamper with a food item as it moves through the supply chain can be immediately identified and prevented before the food reaches the retailer
If there is any hazardous product present, stores can identify and remove only the hazardous items. This eliminates the need for costly batch recalls
The transparency and openness of the blockchain reassures consumers that the food they are buying and eating is exactly what the label says it is.

Fig. 3 Benefits for members in food traceability system

5.2.2 Changes in Blockchain Technology

Blockchain technology is still in the developing phase, and there are some obstacles to expanding it. The transaction capacity of blockchain is seven transactions per second due to the restricted block size. Another problem is increasing the size of the blockchain for storage and synchronization [11].

5.3 *Benefits for Participants*

Figure 3 shows the benefits of participants in the proposed system which are listed below

5.3.1 Farmer

Farmer would get various choices for quotations from processor and can select best profitable quotation.

5.3.2 Processor

Processor would be able to determine any kind of food tampering done by farmer as the processor will verify the quality of the food with farm inspector using smart contract.

5.3.3 Retailer

If there is any hazardous product present, stores can identify and remove only the hazardous items. This eliminates the need for costly batch recalls.

5.3.4 Consumer

The consumer could traceback the food product which provides transparency which increases the trust toward supply chain.

6 Conclusion and Future Scope

The study transforms the traditional supply chain management system into a decentralized food traceability system. The proposed system will track the item moving through every stage, which will be transparent throughout the chain and status of the chain will be displayed, i.e., what details are updated by every participant on the chain will be visible in timeline flow. In the future, information about sales can be provided to the farmer, which can motivate them to grow more efficient crops in large number. Blockchain and IoT together can drastically change supply chain. IoT can provide real-time monitoring support shared with distributed ledgers, which can make supply chain more accurate and scalable.

References

1. Blockchain food traceability can revolutionize the industry. https://www.openlink.com/en/insights/articles/blockchain-food-traceability-can-revolutionize-the-industry/ (Accessed 12th January 2019)
2. Hjalmarsson FP, Hreioarsson GK, Hamdaqa M, Hjalmtysson G (2018) Blockchain-based e-voting system. In: 2018 IEEE 11th international conference on cloud computing (CLOUD), San Francisco, CA, USA, pp 983–986
3. Sari K (2010) Exploring the impacts of radio frequency identification (RFID) technology on supply chain performance. Eur J Oper Res 207:174–183
4. Wang L, Kwok SK, Ip WH (2010) A radio frequency identification and sensor-based system for the transportation of food. J Food Eng 101:120–129
5. Ustundaga A, Tanyasb M (2009) The impacts of Radio Frequency Identification (RFID) technology on supply chain costs. Transport Res Part E: Logis and Transport Rev 45:29–38
6. White Paper—Ethereum/WiKi. https://github.com/ethereum/wiki/wiki/White-Paper (Accessed 13th January 2019)
7. Blockchain. https://en.wikipedia.org/wiki/Blockchain#cite_note-te20151031-1 (Accessed 13th January 2019)
8. Distributed ledger technology -fundamentals you must know. https://codeburst.io/distributed-ledger-technology-fundamentals-you-must-know-2d0f82628258 (Accessed 13th January 2019)

9. Distributed ledger technology (DLT). https://searchcio.techtarget.com/definition/distributed-ledger (Accessed 14th January 2019)
10. What is ethereum. https://cointelegraph.com/ethereum-for-beginners/what-is-ethereum (Accessed 13th January 2019)
11. Tian F (2016) An agri-food supply chain traceability system for China based on RFID and blockchain technology. In: 13th international conference on service systems and service management (ICSSSM), pp 1–6

Healthcare Privacy Approach Using Blockchain Technology

Vrushali Jalgaonkar, Mahesh Shirole, and Sunil Bhirud

Abstract Nowadays, with the advancement in information technology, great progress is seen in the healthcare domain. However, such advancement has also made healthcare data not only much bigger but also much more difficult to handle. Also, the data generated is in different formats and to access such largely scattered data is merely impossible. Health care today suffers from fragmented data, delayed communications, and disparate workflow tools caused by the lack of interoperability. Blockchain technology possesses key properties that can potentially address pressing issues in health care such as incomplete records at the time of care and difficult access to patients' health information in a secure manner. The proposed solution emphasizes on solving a current problem of storing the largely scattered healthcare information in a decentralized fashion and allowing the access of data by the authorized persons only. This application manages authentication, confidentiality, accountability, and data sharing while handling sensitive health information.

Keywords Blockchain · Healthcare data · Medical research centers · Data exchange

V. Jalgaonkar (✉) · M. Shirole · S. Bhirud
Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Mumbai, India
e-mail: vrjalgaonkar_m17@ce.vjti.ac.in

M. Shirole
e-mail: mrshirole@it.vjti.ac.in

S. Bhirud
e-mail: sgbhirud@ce.vjti.ac.in

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_5

1 Introduction

Health Information Technology (HIT) has evolved greatly, but even now, we do not have access to the entire patient's health history in a unified way. We still have different health records with diversified healthcare providers (i.e., healthcare professionals and healthcare organizations) that we interacted in our lifelong period [1, 2]. In a real scenario, a patient visits different doctors and different hospitals for treatment. At every medical appointment, the patient must have to tell his/her whole health history again, which may not be appropriate or accurate with losing time. Electronic Health Record (EHR) is a standardized information model, enabling integration among multiple healthcare providers, ranging from supporting medical prescriptions, improving disease management, and contributing in the reduction of severe medication errors while Personal Health Record (PHR) can receive data entered by patient like the patient's weight, blood pressure readings, etc. [3]. Some healthcare providers have been successful in communicating with patients using mobile technology (mPHR), which allows patient self-monitoring and managing his/her health status [4]. Thus, with such an advancement in healthcare technology, the healthcare data is becoming larger and even outdated.

Many health organizations use databases in a proprietary format. These databases are hosted in a data center inside the health organizations, with restricted access to internal health professionals. In some cases, for example, laboratory results, patients, and healthcare providers can have external access to health records in a restricted manner, only to be viewed or printed [5]. In many cases, the patient's data is not being shared by the healthcare providers and thus, they do not have up-to-date data when their patients are assisted by other healthcare providers. Moreover, these data are stored in different formats in different organizations. Being voluminous, healthcare records are either stored using cloud infrastructure to enable easy access and sharing of information among the different stakeholders or on local databases by the healthcare institutes. In addition, the security and privacy measure offered by the cloud increase the resiliency of data [6]. This brings difficulties for the exchange of healthcare data among the organizations. However, the use of cloud storage does not allow interoperability between different care providers. Also, the integrity and authenticity of the data cannot be guaranteed.

Other problems arise from the existence of duplicate and outdated data in the organizations and also the patient does not have a unified viewpoint of his data. A promising solution to these problems involves the application of blockchain technology, which provides "trustless" transactions via decentralization with pseudo-anonymity [7]. A blockchain system can be considered as a virtually incorruptible cryptographic database where critical medical information could be recorded. The system is maintained by a network of computers, which is accessible to anyone running into the network. Blockchain implements a ledger that validates the existence of digital assets (e.g., coins, as in the case of Bitcoin) and tracks where the control exists for each of them in the network. Such control is distributed, and the one (or ones) who has the control can change the state of the asset (e.g., its controllership) without permissions from any central authority [8]. The access control of heterogeneous patient's health care, records across multiple health institutions and devices needed to be carefully designed. Blockchain itself is not designed for the large-scale data storage system.

In the context of health care, a decentralized storage solution would greatly complement the weakness of blockchain in the perspective. The blockchain network as a decentralized system is more resilient as in that, there is no single-point attack or failure compared to centralized systems. Hence, by storing historical healthcare records of the patient, there will be no loss of data. Also accessing the patient data must be controlled, which can be achieved by blockchain technology.

The above observations of patient's data access and issues in the healthcare data interoperability motivate us to provide blockchain-based unified system. The paper discusses the healthcare records issues which face the problems of

- A single view of both PHR and EHR to be distributed, up-to-date and interoperable to patients, healthcare providers, medical research centers, and insurance companies,
- The historical healthcare data which is being stored in different formats and how to handle them in a unified manner.
- Validation of each peer on every access request.

The remainder of this paper is organized as follows: In Sect. 2, the related work and the existing system are discussed. Section 3 discusses the proposed system. Section 4 gives an overview of the implementation and results. Section 5 concludes the paper.

2 Literature Survey

Blockchain is a platform that alleviates the explicable on a single, centralized authority, yet still supports secure and "trustless" transactions directly between interacting entities [8]. It offers decentralization, immutability, and consensus via cryptography. This technology provides the foundations for several application domains, including cryptocurrency and Decentralized Apps (DApps) [9]. Smart contracts, as implemented in the Ethereum Blockchain [10], that provide code to directly control the exchanges of digital assets between two or more entities according to certain rules or agreements previously established between involved participants. Storing data objects and defining operations on that data can be done using smart contracts, enabling the development of DApps to interact with blockchains and provide seamless services to the application users.

In the healthcare domain, smart contracts are being applied to create secure and effective technical infrastructures to enhance care coordination and quality and thus improve the well-being of individuals and communities [7, 11]. A major problem in the production of healthcare systems today is the lack of secure links that can connect all independent health systems to communicate and establish an end-to-end reachable network while protecting healthcare information with some level of anonymity [6, 12, 13]. Although data standards like HL7 and FHIR [14] provide basic interoperability for data exchange between trusted systems, this level of interoperability is limited to the implemented standards and requires mapping of data between systems in most cases. Also, to achieve maintainability for these systems is difficult because an interface change on one system requires other parties in the trusted network to adopt the change as well. Startups like *Ujo* or *Peertraks* offer a new approach, such as how music rights can be administered and enforced using blockchain. At *Stampery*, for example, contracts, emails, and documents can be signed digitally and smart contracts can be settled [7].

The confidentiality and protection of individually identifiable health information that is transferred, received, handled, or shared by healthcare professionals and organizations must require Health Insurance Portability and Accountability Act (HIPAA) rules [15]. EHR is "information related or relevant to the wellness, health, and healthcare of an individual, in computer processable form and represented according to a standardized information model." PHR refers to a "representation of information regarding, or relevant to, the health, including wellness, development, and welfare of that individual" [3]. All systems and apps created to share health information must be HIPAA compliant. MedRec: a novel decentralized record management system [16] handles EMRs (electronic medical records), gives a comprehensive, immutable log, and easy access to patient's medical information across healthcare providers and treatment sites.

Permissionless and permissioned blockchain [17] can be helpful in managing and sharing electronic health and medical records to allow patients, hospitals, clinics, and other medical stakeholders to share data amongst themselves, and increase interoperability. In [18], a set of evaluation metrics is defined which can be used to assess DApps designed to address healthcare interoperability issues such as (1) Entire workflow is HIPAA compliant, (2) Framework employed needs to support turing complete operations, (3) Support for user identification and authentication, (4) Support for structural interoperability at minimum, (5) Scalability across large populations of healthcare participants, (6) Cost-effectiveness, and (7) Support of patient-centered care model.

By seeing all the related work, the disadvantages of the existing systems do not allow the patient's health data to be exchanged between different healthcare organizations. An efficient and effective solution is provided to store all the healthcare history of the patient in a decentralized fashion which can be accessed by different health institutes, medical researchers, and insurance companies by providing privacy on the data.

3 Proposed System

In the proposed work, we focus on the storage and controlling the access of both PHR and EHR data without any Trusted Third Party (TTP) in a distributed environment. The model's purpose is to carry out data integrity, confidentiality as well as to eliminate the inconsistency for the end-user while allowing a unified view of health records that are distributed in several health organizations. The model proposes a way in which healthcare data are organized hierarchically, encrypted, and distributed in chained data blocks on the network. These blocks store data that are located in different healthcare organizations and even in a patient-managed repository.

Figure 1, the entities involved in the system can access the patient's healthcare data which is stored in the IPFS cloud through blockchain [19]. The health data like laboratory analysis report which can be in text or image or a document format can be stored on IPFS because the data size is larger and the IPFS hash which is only of few tens of bytes is stored on the blockchain.

The major objectives of the proposed work are as follows:

- To design an approach for the healthcare domain where the system stores all historical data into the blockchain manner.
- To create a distributed environment hierarchy for parallel data processing for end-user applications.
- Validation of each peer on every access request.

The users involved in the system are

1. **Patient**—The patient's medical history is being stored which can be accessible by the authorized entities of the network. The ownership of the data is being managed by the patient.

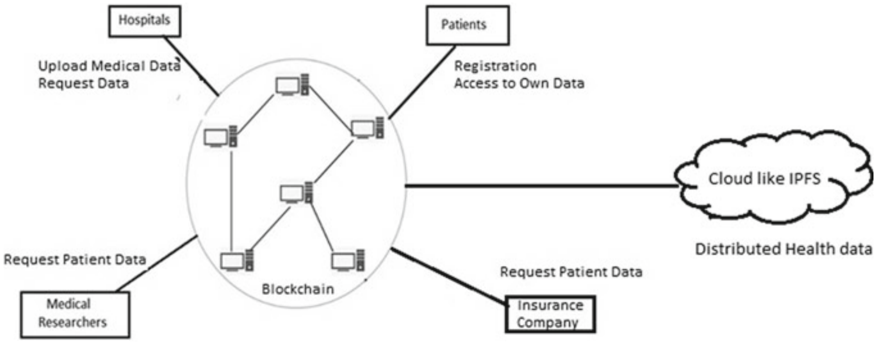


Fig. 1 System architecture diagram for the patient's storage and retrieval of information

Table 1 List of system actors with smart contracts

Actor	Contracts
Patient	1. Patient. sol (a) Patient registration (b) Permit request (c) View history
Hospital	2. Hospital. sol (a) Hospital registration
Doctor	3. Doctor. sol (a) Doctor registration (b) Update patient report
Medical researchers	4. Medical research. sol (a) Med research registration (b) Request patient data
Insurance company	5. Insurance company. sol (a) Insurance company registration (b) Request patient info

2. **Hospitals**—The doctor will treat the patient. The report generated after treatment will be uploaded to the network, thus maintaining the up-to-date healthcare information of the patient. This information would be useful by the other healthcare organizations where the patient would visit for health assistance.
3. **Medical researchers**—The patient's data should not be unknowingly accessed by anyone. The healthcare institutes would be restricted to give patient data to medical researchers without the knowledge of the patient.
4. **Insurance company**—While the patient seeks any health insurance or life insurance, the procedure to claim that benefits are time consuming. Thus, by providing request-based access to patient's health information would make the hectic insurance claiming process much faster and also reliable.

During the registration time, the new user is verified by the email verification. Once the user is verified, he/she would be considered as the verified entity of the blockchain network. Based on request access to the data, verification of the entity is done by the other blocks present in the network using a delegated proof of stake consensus mechanism algorithm. Thus, only validated entities of the network get a chance to access the healthcare information. The control access of uploading the patient's treatment report is being given to the hospital's staff, in which the authorized doctor would have all access to the health information while the other hospital staff, for example, nurse, is given access to the only view the data. The medical researchers and insurance companies can only view health data after the request is granted by the patient (Table 1).

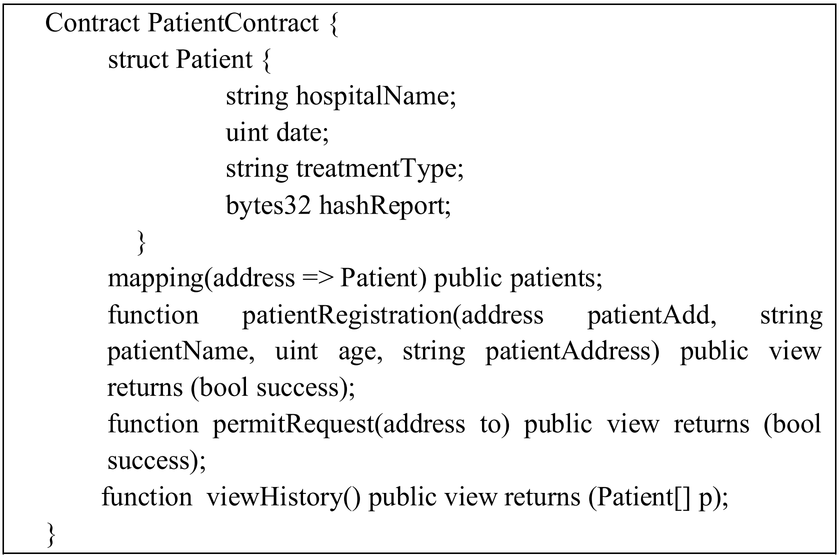
4 Implementation and Results

The contracts for the system are written in Solidity language using the truffle framework. Truffle is a development environment, testing framework, and asset pipeline for Ethereum. Metamask is an extension that allows you to run Ethereum DApps right in your browser without running a full Ethereum node.

Following is an interface contract declaring the required functions for the patient. The data structures used are structure and map. The contract's patient data structure will contain the following information.

- *hospitalName* will store the name of the hospital,
- *date* for storing the date during which the patient had reported to the hospital for the treatment,
- *treatmentType* declares the type of treatment which the patient had, it can be any medical checkup or any laboratory test like a blood test,
- *hashReport* stores the hash for the medical report which will be given during the treatment.

The patientRegistration(…) function is used to store the patient's details. The function permitRequest(…) takes the "to" address of the entity who wishes to access the patient's data. The patient's data is accessible by only those to whom the permission is granted. The function viewHistory() returns the historical patient data which the patient wants to view.



```
Contract PatientContract {
        struct Patient {
                        string hospitalName;
                        uint date;
                        string treatmentType;
                        bytes32 hashReport;
            }
        mapping(address => Patient) public patients;
        function patientRegistration(address patientAdd, string
        patientName, uint age, string patientAddress) public view
        returns (bool success);
        function permitRequest(address to) public view returns (bool
        success);
       function viewHistory() public view returns (Patient[] p);
}
```

Following is an interface contract declaring the required functions for the hospital. The data structures used are structures and mapping. The contract's hospital data structure will contain the following information.

- *hospitalName*, stating the hospital name
- *hospitalAddress*, the location address of the hospital

The hospitalRegistration(…) function registers the hospital.

```
contract Hospital Contract {
    struct Hospital {
            string hospitalName;
            string hospitalAddress;
      }
    mapping(address => Hospital) public hospitals;
    function hospitalRegistration(address hospitalAdd, string
    hospitalName, string hospitalAddress) public view returns (bool
    success);
}
```

Following is an interface contract declaring the required functions for the doctor. The data structures used are structures and mapping. The contract's doctor data structure will contain the following information:

- *doctorName*, takes the doctor's name who will be treating the patient.
- *Specialization*, specifies the highest qualification of the doctor.

The doctorRegisration(…) function registers the doctor into the network. The updatePatientReport(…) function uploads the hash generated of the medical report of the patient and a brief description of the medical report. This can be done as follows:

updatePatientReport (0×1111111111111111111111111111111111111111, "Blood test", QmYqSCWuzG8Cyo4MFQzqKcC14ct4ybAWyrAc9qzdJaFYTL)

"QmYqSCWuzG8Cyo4MFQzqKcC14ct4ybAWyrAc9qzdJaFYTL" is the hash of the medical report.

```
contract Doctor Contract {
        struct Doctor {
                string doctorName;
                string specialization;
          }
        mapping(address => Doctor) public doctors;
        function doctorRegisration (address doctorFrom, string
        doctorName, string specialzation) public view returns (bool
        success);
        function updatePatientReport (address patientAdd, string
        reportDesc, bytes32  hashReport ) public returns (bool success);
}
```

Following is an interface contract declaring the required functions for the Medical Research Center. The data structures used are structures and mapping. The contract's Medical Research data structure will contain the following information.

- *hashReport*, is the hash of the medical report which the medical research centers want to view.

The function medReseachRegistration(...) registers the medical researchers. The patient's data is requested using requestPatientData(...) function.

```
Contract MedicalResearchers Contract {
       struct MedicalResearch {
                bytes32  hashReport;
          }
        mapping(address      =>      MedicalResearch)      public
        medicalResearchers;
        function  medReseachRegistration  (address
        medResearcherFrom, string medResearchName) public view
        returns (bool success);
        function requestPatientData (address medResearcherFrom)
        public view returns (MedResearch[] M);
}
```

Following is an interface contract declaring the required functions for the insurance company. The data structures used are structures and mapping. The contract's insurance company data structure will contain the following information:

- *hashReport*, is the hash of the medical report which the medical research centers want to view.

The function insurance Company Registration(…) registers the insurance companies. The patient's data is requested using request Patient Info function.

```
contract InsuranceCompanyContract {
      struct InsuranceCompany {
                  bytes32 hashReport;
        }
      mapping(address => InsuranceCompany) public
      insuranceCompanies;
      function insuranceCompanyRegistration (address insComAdd,
      string insComName, string insComAddress) public view returns
      (bool success);
      function requestPatientInfo (address insComAdd) public view
      returns (InsuranceCom[] i);
}
```

The advantage of this system is that the historical data of the healthcare which is shared by the various medical institutes and by patients can be secured using blockchain technology. The data integrity and confidentiality can be maintained. The data can be shared between the entities in the network securely. Thus, providing an effective way to preserve the privacy of the patient's healthcare data.

We have implemented the system, while it is under test and yet not tested in the real environment.

5 Conclusion

In this paper, the distributed and scattered data of patients and hospitals are channelized using the proposed system. The system facilitates hospitals to use and interoperate healthcare data of the patients stored by different stakeholders seamlessly. Patients are capable of accessing the complete health record and authorize requesting parties to use and view their health records. The healthcare data appear to be centralized from the logical viewpoint of patient and healthcare provider, but in fact, they are physically decentralized. Usage of IPFS allowed to store a large volume of patient's data which are in different formats stored on the cloud. This data is accessible only through the blockchain; thus, reducing the misuse of the data. Thus, the model aims to support patients to take advantage of having their health history single access, as well as for healthcare providers to have their patients' health data up-to-date.

References

1. Gorp PV, Comuzzi M (2012) Lifelong personal health data and application software via virtual machines in the cloud. IEEE J Bio Health Inf 0(0)
2. Bourgeois FC, Nigrin DJ, Harper MB (2015) Preserving patient privacy and confidentiality in the era of personal health records. PEDIATRICS, 13(55)
3. ISO, Health informatics—Capacity-based ehealth architecture roadmap—Part 2: Architectural components and maturity model, Technical Report (ISO/TR TR14639-2). https://www.iso.org/obp/ui/#iso:std:iso:tr:14639:-2:ed-1:v1:en
4. Reeder B, David A (2016) Health at hand: A systematic review of smart watch uses for health and wellness. J Bio Inf
5. Kraan C.W., Piggott J.J.H., Vegt F.V.D, Wisse L.: 'Personal Health Records: Solving barriers to enhance adoption ', July 2015
6. Divya T, Shanmugapriya L (2016) Two-level security in shared m-healthcare system using owncloud. IEEE
7. Matthias M, HSG MA (2016) Blockchain Technology in Healthcare. In: The international conference on e-health networking, applications and services (Healthcom), IEEE
8. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system
9. Johnston D, Yilmaz SO, Kandah J, Bentenitis N, Hashemi F, Gross R, Wilkinson S, Mason S (2014) The general theory of decentralized applications, dapps. Github—.https://github.com/DavidJohnstonCEO/DecentralizedApplications
10. Buterin V (2013) A Next-generation smart contract and decentralized application platform. Ethereum white paper, GitHub repository
11. Leslie M.,: 'A blockchain revolution sweeps into health care, offering the possibility for a much-needed data solution', IEEE Pulse, June, 2018
12. Kim H, Song H, Lee S, Kim H, Song I (2016) A simple approach to share users. Own Healthcare Data with a Mobile Phone ICUFN, IEEE
13. Zhang Y, Qiu M, Chun-Wei T, Mohammad MH, Alamri A (2015) Health-CPS: Healthcare cyber-physical system assisted by cloud and big data. IEEE SYSTEMS JOURNAL, IEEE
14. Bender D, Sartipi K (2013) HL7 FHIR: an agile and restful approach to healthcare information exchange', IEEE
15. H. Office for Civil Rights: standards for privacy of individually identifiable health information. final rule. Federal Register, vol 67, no 157, p 53181, 2002. https://www.nap.edu/read/12458/chapter/7
16. Azaria A, Ekblaw A, Vieira T, Lippman A (2016) MedRec: using blockchain for medical data access and permission management. In: 2nd international conference on open and big data
17. Alhadhrami Z, Alghfeli S, Alghfeli M, Abedlla JA, Shuaib K (2017) Introducing blockchains for healthcare', international conference on electrical and computing technologies and applications (ICECTA)
18. Zhang P, Walker MA, White J, Schmidt DC, Lenz G (2017) metrics for assessing blockchain-based healthcare decentralized apps. In: IEEE 19th international conference on e-health networking, applications and services (Healthcom)
19. IPFS information. https://en.wikipedia.org/wiki/InterPlanetary_File_System

Supply Chain Management in E-Commerce Using Blockchain

Vaibhav D. Dhore and Neha Mishra

Abstract With the increase in the Internet and online shopping era, online shopping and E-commerce have become the most convenient way for customer and seller to buy and sell the product online using Internet. With this increasing E-commerce services, it has become mandatory to provide the trusted proof of delivery system to the customer and seller. Existing system is centralized system which completely relies on the third party for the proof of delivery which can lead to a single point of failure. Existing proof of delivery systems leads back in credibility, transparency, and traceability. Blockchain is the decentralized distributed ledger which provides the system to be transparable, traceable, and creditable. Supply chain management using Ethereum blockchain is one of the solutions that can be given. This system is the decentralized proof of delivery system that uses Ethereum smart contract to prove the delivery of a shipped item between seller and buyer. Each customer, seller, and courier services registers on the Web application. Later, seller uploads catalog of the commodities to be sold which customers choose to buy. Courier services are assigned this parcel based on the sentiment analysis, and later, at the time of delivery, payment is done using ethers.

Keywords Blockchain · Ethereum · Smart contracts · Supply chain

1 Introduction

With the widespread of technology and Internet, it has become a new trend for buying and selling the commodities online using Internet via E-commerce. This online shopping service provides customers to compare, review, and check the current trends in the market. Hence, they get a wide range of options over the different brands. E-commerce shopping is getting more favored with time especially with the increase

V. D. Dhore (✉) · N. Mishra
VJTI, Mumbai University, Mumbai, India
e-mail: vddhore@ce.vjti.ac.in

N. Mishra
e-mail: Nbmishra_m17@ce.vjti.ac.in

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_6

in the use of cell phones and Internet. Hence, it has given rise to increase in the demand of the different commodities. Hence, delivery is provided by the third party to increase the supply of the mandatory demand. Therefore, proof of delivery of physical item and product has become mandatory and need of the current time. Hence, it is needed to facilitate the shipment in a way that is trusted, transparent, and traceable mostly for the seller, buyer, and transporters that are located around the globe.

Proof of delivery helps to provide the assurance of shipped items has reached from source to destination without being tampered. This process provides each entity involved with trust. Existing proof of delivery falls back in providing the transparency, traceability, and credibility. Most of the current system is centralized and depends on the paper document which is carried by the courier services. With the advancement of the cell phones, handheld devices also being used to share the report along with the commodities. Hence, this system relies on the support of the trust on the courier services. These third-party service providers are mostly unreliable and untrusted and cost much more for delivery.

Blockchain is a decentralized distributed ledger which is immutable. A blockchain uses log which is ordered and each event, i.e., transaction is being recorded which helps to trace and audit. Hence, using Ethereum blockchain would help to create smart contract and allow the execution of the code. Blockchain transaction is being irreversible, i.e., one cannot change once the transaction is being done and created. These transactions are being stored in the form of blocks. Each block consists of the data, timestamp, and the hash value of the previous block. These blocks are being organized in the form of the Merkle tree.

An optimized proof of delivery solution should provide the requirement which are mandatory for the trade are accountability which means the entities involved cannot deny any action taken i.e. it is irreversible, penalty and incentivization which means it provide trust between entities by giving them incentive hence they are being forced to act honestly else penalty is being paid, auditability which provides proper to trace and track the system, integrity where each transaction, logs and events are being tampered proof, authentication and authorization these provide the access and authority to legitimate users, time bound which ensures the shipped item is placed within the time frame [4].

In this paper, we provide the solution for the proof of delivery system using blockchain technology. In the given framework solution, the defined system involves single or multiple courier services. This solution provided mainly focuses on the eradicating the problems faced with the current centralized system which is based on the third party for the delivery. Here, the given proof of delivery system is being implemented between seller and buyer irrespective of the number of the courier services. Payments are being done in the form of the ethers which force entities to act honestly. Interplanetary file system do provide the integrity between entities and the contract with the help of the defined hash being stored.

2 Related Work

In this section, we review and summarize work related to proof of delivery algorithms and techniques that make use of blockchain.

In [1], blockchain is a distributed database of records or public ledger of all transactions or digital events that have been executed and shared among participating parties. Each transaction in the ledger is verified by consensus of a majority of the participants in the system. The entered records cannot be deleted. The popular example that uses blockchain technology is Bitcoin; it is a peer-to-peer decentralized digital currency. The digital currency Bitcoin itself is highly controversial but the underlying blockchain technology has worked flawlessly and found a wide range of applications in both financial and non-financial world. Various applications on healthcare, identity management, cloud, Iota, security, banking, and supply chain are being en-cooperated with blockchain.

In [2], it is being explained that various aspects of blockchain can be used in different fields. Various new consensus protocol can be build such as proof of intelligence which can be later used in AI. Here, how the blockchain works logically and smart contracts can be used to gain the verified blocks.

In [3], a blockchain is a public ledger distributed over a network that records transactions executed among network participants. Each transaction is verified by network nodes according to a majority consensus mechanism before being added to the blockchain. Recorded information cannot be changed or erased and the history of each transaction can be recreated at any time. Blockchain can be used in various fields which can give better way for recoding the records such as identity management such as records of voter Id, Aadhar card data, and driving licence. Blockchain can provide various advantages like trust between the participating entities, transparency, decentralization, and automation. The basic transaction of records is shown as below.

In [4], blockchain is a distributed ledger which means all the parties taking part in the transaction or the parties present on the blockchain has the copy of the ledger and there is no centralized database. In case of any failure of the centralized database, it may lead to data loss but with blockchain this problem is solved. Another important advantage provided is the transparency of the transactions. Blockchain has found application in many new technologies with its promising features. Digital identity management can be done by blockchain where we can control our identity without depending on any central authority. It enables us to share our identity according to the need and can protect user's consent. One Name is one of the companies providing such digital authentication. In 2015, Bitcoin Foundation started a new project on the blockchain-based voting system to ensure transparency in the voting system with every vote being recorded on the blockchain. Immutability, cryptographic hash, and transparency are the advanced features provided for the blockchain based voting system. In supply chain management, blockchain enabled us to keep the track of the origin of the products. We can ensure the quality of the end products by maintaining the immutable record on the blockchain. Further, blockchain is also used with IoT, medical services, baking, real estate, etc.

In [5], various protocols are used in blockchain for adding the new block in the chain by the form of verification. This verification is done by the consensus protocol. Consensus protocol is of two major types: proof of work and proof of stake. Depending on the type of blockchain, these protocols changed. Blockchain can be of permissioned and permissionless. Access privacy of both protocols depends on the type of the blockchain.

In [6], blockchain is a new technology for data sharing between untrusted peers. However, it does not work well with massive transactions. Besides, there are high barriers between heterogeneous blockchain systems. In this paper, it is proposed that an innovative component-based framework is for exchanging information across arbitrary blockchain system called interactive multiple blockchain architecture. In given architecture, a dynamic network of multi-chain is created for inter-blockchain communication. Here, it is proposed that the inter-blockchain connection model is for routing management and messages transferring. Additionally, their proposed protocols provide transactions with atomicity and consistency in crossing-chain scene.

In [7], proof of delivery plays a major role in the delivery of any physical assets. In this paper, to maintain the trust of each entity, double collateral has been taken in the form of ethers. An automated system has been implemented which on successful delivery of the product will return ethers to the entities depending upon the given share. An automated dispute solver has also been implemented in order to gain the minimum dispute among the entities.

Each entity is managed by the smart contract which helps to develop the proper solution for the courier services. Here, more than one courier service might be required for the delivery based on the address of the delivery point. Hence, each product has the verification key which is being shared to each other for the purpose of the authentication. When the seller gives the courier service for delivery, it first checks whether the address is near me or not. If it is near it then the delivery is done by the same courier service where the customer verifies the keys given to it and then the delivery is being confirmed. If the address does not fit in the location of the courier service, then it transfers to other courier services, and key verification is done and later the delivery takes place. Here, more than one courier services might be engaged for the delivery based on the address. An arbitrator manages any dispute among the courier services and seller for the purpose of the ether transfer. If dispute happens then the whole transaction being made is failed and then the whole process is repeated again. Easy return policy has also been implemented for the purpose of better services than the traditional method of relying on the third party.

In [8] paper solution related to maintenance of whole agri-food details using RFID tags from raw materials to the final products like vegetables and meats, and fruits like for e.g. meat of pig contains details of pig, its parents, any disease related to pig, their slighter details is stored in the block chain in various units like from production to distribution to end users. Each RFID sends signal data information using sensor and network to blockchain database. These data would help to get track of the origin of the product and final delivery destination. If any major issue such as poisoning due to rotten food happens, then the further delivery can be stopped as we would know the source and destination where food has been supplied.

In [9], current anti-counterfeiting supply chains rely on a centralized authority to combat counterfeit products. Single point storage, processing and failure are the problems in this architecture. Solutions to these problems have provided blockchain technology. This paper proposes a supply chain system for detection of counterfeiting attack which is a decentralized supply chain with the help of blockchain and near-field communication (NFC) technologies. Block-supply chain replaces the centralized supply chain design and utilizes a new proposed consensus protocol that is, unlike existing protocols, fully decentralized and balances between efficiency and security. Simulation in the paper shows that the proposed protocol has good performances as well as gives nice security compared with the state-of-the-art consensus protocol Tendermint.

In [10], RFID is being used in order to perform the log of the ownership of the product; hence, due to the log creation of the system, a customer can refuse the product if the manufacturing company fails to give the ownership of the product hence helped to reduce the anti-counterfeits. Hence, here, post-supply chain ownership management is important. After the defined system of the supply chain, at the later point, it can be disrupted at the retailer where the product can be mixed with the duplicate product or the forged product. Hence, every entity can be stored on the blockchain and it can help to gain the proper verification of the genuineness of the product.

In [11], smart city has all data over the network. A smart city uses information technology to integrate and manage physical, social, and business infrastructures in order to provide better services to its dwellers while ensuring efficient and optimal utilization of available resources. With so many technologies as cloud computing, Internet of things (IoT), and interconnected networks, more innovative solution, direct interaction, and collaboration between local government and citizens can be delivered by smart cities. Even though there are many potential benefits, disruption causes many problems related to information security and privacy. This paper presents a system with a security framework that is a combination of blockchain technology and smart devices for a secure communication platform in smart cities.

Here, we are using the Ethereum smart contract service to create the smart contracts for the data management. Here, the shipment is sent and the front of the mobile app tracks the data using Bluetooth LE sensor and keeps track of the temperature, and when the shipment reaches, these temperature records are then sent to the smart contract, i.e., Ethereum node, for keeping log and taking the method for verification of the proper environment.

Current existing system includes third party for the delivery and there is no transparency for the owner of the product. Each transaction made during the process of buying and selling of a particular item can be tampered without any proof which may cause primary dissatisfaction to customer. Data log of the current system is reversible. Anyone can change the details of the item and the transaction log.

3 Proposed Solution

In the proposed blockchain solution, the main focus is given on the proof of the delivery of the commodities between seller and buyer. We are using Ethereum blockchain smart contract to maintain the proof of delivery between courier services and customer. The main entities of the system are:

- Customers: Customers are the main entities of the defined system. Each customer is provided to register at the Web-based application. These details are being stored on the blockchain in the form of the identity management. For being more concern, we can even put know your customer format of the bank by using the system such as Aadhar number.
- Seller: Seller is the major role player who sells their product on these sites and helps to maintain the E-commerce. The seller has the item to be packaged for transfer to the interested buyer. The seller creates the first contract in the chain. Therefore, the seller is the owner of the first contract created.
- Courier services: Courier services are the entities which help to deliver the product over the defined time period. Multiple couriers are available to deliver the item from the seller to the buyer if needed based on the geolocation of the seller and buyer. The transporter creates the next contract in the chain (Fig. 1).

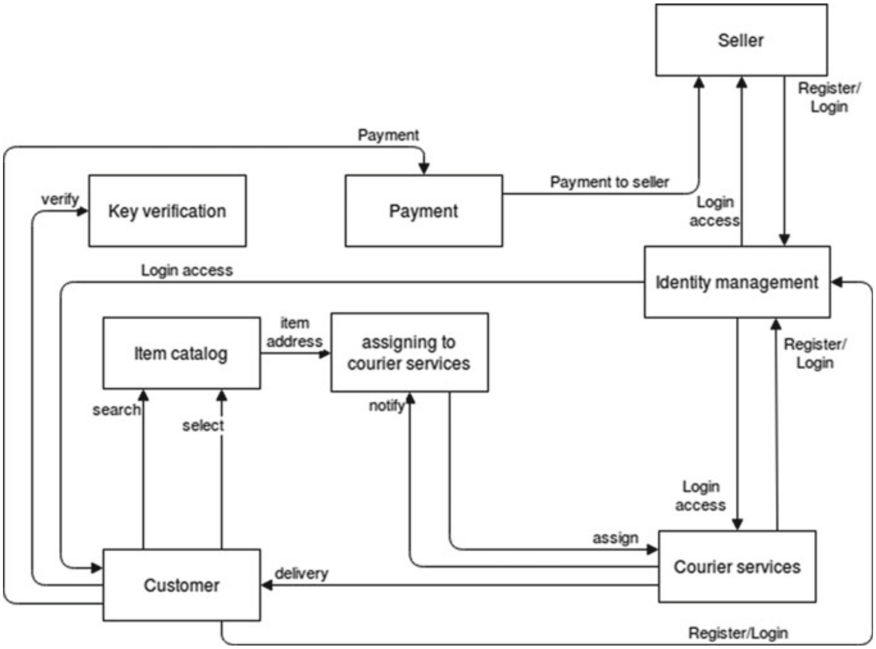


Fig. 1 Supply chain management in E-commerce using blockchain proposed system

In our system, customer registers to the blockchain using the Web application where their data is stored in the form of blocks. Each customer gets access to the item catalog from where they can choose the item to buy. Seller registers to the blockchain similar to the customer, and even courier services get registered.

Once any register customer requests for the product, that product is set for delivery by seller. Using sentiment analysis on the user review for the courier services, delivery is assigned to that service.

During the process of assignment, the key is generated and provided to the courier services which would be required at the time of verification. Customer gets the details and key as well. During the part of delivery, customer verifies the key and then the delivery is confirmed. Payment is done in the form of ethers, later which is transferred to the seller. Each transaction is recorded and maintained for future references.

In order to achieve the needed functionality with transparency and tracing the item as it moves through the chain of contracts, the smart contracts contain the following:

- Methods: Methods are used in smart contracts to create function calls. Each function is responsible for executing and implementing a desired action. Hence, in this work, some of the important functions we have created include methods to deposit the collateral, perform the key verification between any two parties as well as settle the payment and handle the dispute. All public variables have automatic getter functions created for them. However, setters have to be created as required. Hence, to change the state of a contract, a setter function was created to allow only its parent or its own child to alter its state.
- Modifiers: Modifiers are used in the smart contracts to create a requirement before the execution of a function. For instance, the collateral should be certain agreed upon amount. This is checked using a modifier. Other modifiers were also used to restrict the execution of a function based on the Ethereum address of the function caller. Therefore, certain functions can only be executed by the seller, others by the transporters and buyer, respectively.
- Events: Events act as notifications and are used as logs which can help in tracing back in case of dispute. Therefore, any function that is executed creates an event that updates all entities about the status of the item and contract until now.
- Variables: Variables are used to store information that might change as the transaction progresses or that are needed for certain checks and functionalities. Therefore, the main variables in the contracts are used to store Ethereum addresses of the participating entities, the key hash that is used in the key verification comparison, the item price, the contract state, IPFS hash, and the address of the child contract for each parent contract in the chain.

Following is an interface contract declaring the required functions for the customer. The data structures used are structure and map. The contract's customer's data structure will contain the following information:

- *custmerName* will store the name of the customer,
- *customerAddress* will store the address of the customer which will be required for the delivery purpose.
- *custmerHashReport* stores the hash for the report which will be given during the delivery process.
- *custmerAdd,* the blockchain account address.

The customer Registration(…) function is used to store the customer's details. To register the customer, the function will be executed as follows,

customerRegistration(0x1111111111111111111111111111111111111111,
Customer1, Mumbai, customer1@gmail.com)

The function viewHistory() returns the historical customer data which the customer wants to view.

```
Contract CustomerContract {
        struct Customer {
address customerAdd;
string customerName;
uint age;
string email;
bytes32 custmerHashReport;
string addr;
}
        mapping(address => Customer) public customer;
function customerRegistration(address customerAdd, string customerName,
string customerAddress, string customeremail) public view returns (bool
success);
        function viewHistory() public view returns (Customer[] c);
}
```

Following is an interface contract declaring the required functions for the seller. The data structures used are structures and mapping. The contract's seller data structure will contain the following information:

- *sellerName*, stating the seller name.
- *sellerAddress*, the location address of the seller.
- *sellerAdd,* the blockchain account address.
- *sellerEmail* mail id for contact.

The sellerRegistration(…) function registers the seller. The function is as follows,

sellerRegistration(0x3333333333333333333333333333333333333333, Seller1, Mumbai, seller1@gmail.com)

The updateItem(…) function uploads the hash generated of the details of the item and a brief description of the item. This can be done as follows,

updateItem(0x1111111111111111111111111111111111111111, "Mi LED TV 4X PRO 138.8CM", QmYqSCWuzG8Cyo4MFQzqKcC14ct4ybAWyrAc9qzdJaFYTL)

"QmYqSCWuzG8Cyo4MFQzqKcC14ct4ybAWyrAc9qzdJaFYTL" is the hash of the item details.

```
    contract SellerContract {
          struct Seller {
    address sellerAdd;
    string sellerName;
    string sellerAddress;
    string sellerEmail;
    }
           mapping(address => Seller) public sellers;
    function sellerRegistration(address sellerAdd, string sellerName, string
sellerAddress, string sellerEmail) public view returns (bool success);
    function updateItem (address sellerAdd, string itemDesc, bytes32
itemHashReport ) public returns (bool success);
}
```

Following is an interface contract declaring the required functions for the courier services. The data structures used are structures and mapping. The contract's courier services data structure will contain the following information:

- *courierServicesName* takes the courier services' name who will deliver the product to customer.
- *courierServicesEmail* takes the mail id of the courier services.
- *courierServicesAddress,* it stores the address of the courier service's address.
- *courierServicesAdd,* the blockchain account address.

The courierServicesRegisration(…) function registers the courier services into the network.

courierServicesRegisration(0x4444444444444444444444444444444444444444, CourierServices1, courierservices1@gmail.com, Mumbai)

```
   contract CourierServicesContract {
          struct CourierServices {
   address courierServicesAdd;
   string courierServicesName;
   string courierServicesEmail;
   string courierServicesAddress;
   }
          mapping(address => CourierServices) public courierServicess;
   function CourierServicesRegisration (address courierServicesAdd, string
courierServicesName, string courierServicesEmail, string courierServicesAddress)
public view returns (bool success);

}
```

Following is an interface contract declaring the required functions for the item datastore. The data structures used are structures and mapping. The contract's item datastore data structure will contain the following information:

- *itemHashReport* is the hash of the details of the item

The function itemRegistration(…) registers the item details. The function is,

itemRegistration(1, Item1) will return the status of the registered item.

The item detail is requested using requestItemData(…) function. It can be done as follows,

requestItemData(1), the item data is returned.

```
   Contract ItemContract {
         struct Item{
   uint itemId;
   bytes32  itemHashReport;
   }
   mapping(uint => Item) public items;
   function  itemRegistration  (uint itemId, string itemName) public view returns
(bool success);
   function requestItemtData (uint itemId) public view returns (Item[] I);
}
```

4 Conclusion

In this paper, we have provided a solution for the centralized E-commerce site which can be tampered. This solution helps to give the system to be decentralized and provide it with the proper proof of the delivery of the commodities in the single as well as the multiple courier services. This solution eliminates the third-party reliability of the system for the delivery. It creates benefit for the customer. Each and every transaction is being recorded, and hence, it cannot be tampered as blockchain has the property of being irreversible. This system provides transparency to the customer and has a record of the ownership of the item.

References

1. Crosby M, Pattanayak P, Verma S, Kalyanaraman V (2016) Blockchain technology: beyond bitcoin. Appl Innov 2:6–10
2. Gatteschi V, Lamberti F, Demartini C, Pranteda C, Santamaría V (2018) To blockchain or not to blockchain: that is the question. IT Professional 20(2):62–74
3. Henry R, Herzberg A, Kate A (2018) Blockchain platforms: a compendium. Blockchain Access Privacy, IEEE, pp 38–45
4. Kan L, Wei Y, Muhammad AH, Siyuan W, Linchao G, Kai H. A multiple blockchains architecture on inter-blockchain communication. In: 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C) 2018 Jul 16. IEEE, pp 139–145
5. Tian F (2016) An Agri-food supply chain traceability system for china based on RFID & Blockchain Technology. IEEE
6. Toyoda K, Mathiopoulos PT, Sasase I, Ohtsuki T (2017) A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain. IEEE Access 5:17465–17477
7. Alzahrani N, Bulusu N. Block-supply chain: a new anti-counterfeiting supply chain using NFC and blockchain. In: Proceedings of the 1st workshop on cryptocurrencies and blockchains for distributed systems 2018 Jun 15. ACM, pp 30–35
8. 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 2016 Dec 12. IEEE, pp 1392–1393
9. Hasan HR, Salah K (2018) Blockchain-based proof of delivery of physical assets with single and multiple transporters. IEEE Access 6:46781–46793
10. Consumers are now doing most of their shopping online. Accessed 13 Jun 2018
11. UPS study: purchases from marketplaces nearly universal retail now global as e-commerce shoppers cross borders. Accessed 13 Jun 2018. Available: http://fortune.com/2016/06/08/onlineshopping-increases/

Electronic Polling Agent Using Blockchain: A New Approach

Aishwarya Babu and Vaibhav D. Dhore

Abstract The blockchain supports all kinds of potential for improving digital systems. Frequently, we hear about all the potential of blockchain in reference to economic services and finance systems. But the digital system that could possibly be most enhanced by blockchain is e-voting. Over the progression of the former election cycle, fraud, fake votes, and "system rigging" were widespread. This makes it clear that both conventional paper-based ballot system and the centralized electronic-based voting system have definite apertures creating vulnerabilities and vote manipulation. The blockchain is an immutable ledger, and integrating blockchain technology with a voting system can give the solution to numerous problems faced by modern systems. The motive for e-voting system using blockchain is to facilitate transparency, fairness, and audibility. While the number of voters increases, the time taken to cast vote, i.e., time taken to generate block also increases due to which the throughput of the system decreases. A new system is proposed in this paper, which aims to provide a fast, secure, and transparent voting approach using blockchain.

Keywords Blockchain · Polling agent · E-voting

1 Introduction

With the sudden popularity of bitcoin, the technology behind it which is blockchain had its come back. Before it existed in cryptocurrency, it had modest origins as an idea in computer science, especially in the domains of cryptography and data structures. It consolidates the openness of the Internet with the security of cryptography to provide everyone with a quicker, secured way to verify important information and build trust. Due to its potential feature, blockchain acts as a perfect technology to combine with polling agent.

A. Babu (✉) · V. D. Dhore
VJTI, Mumbai University, Mumbai, India
e-mail: aishwaryababu95@gmail.com

V. D. Dhore
e-mail: vddhore@ce.vjti.ac.in

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_7

1.1 Blockchain

Blockchain can be denied as a prototype of the distributed ledger for controlling a durable and tamper-proof account of transactional information. Originally, blockchain is an increasing list of records, described blocks, which are associated using cryptography. Per block comprises a cryptographic hash concerning the preceding block, a timestamp of the transaction, and the actual transaction data. By intention, a blockchain is a repellent to alteration of the data. It is "an unrestricted, distributed ledger which logs events among two parties effectively as well as efficiently in a variable and intermittent way" (Fig. 1).

To make application as a distributed ledger, a blockchain is achieved by a peer-to-peer interface jointly adhering to a contract for inter-node transmission and validating new blocks. Once recorded, the data in each assigned block cannot be remodeled retroactively without revision of all subsequent blocks, which demands an agreement of the network majority. Although blockchain recordings are not inevitable, blockchains may be thought to be guarded by design and illustrate to be a dispersed computing system with great fault immunity. Decentralized consensus has therefore been claimed with a blockchain.

Currently, there are three varieties of blockchain networks

1. Public blockchain.
2. Private blockchain.
3. Consortium blockchain (Fig. 2).

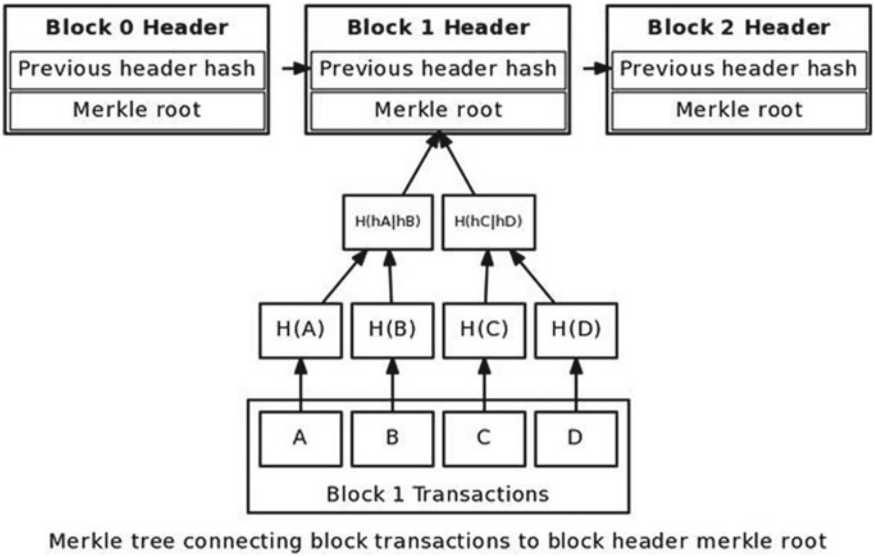


Fig. 1 Blockchain

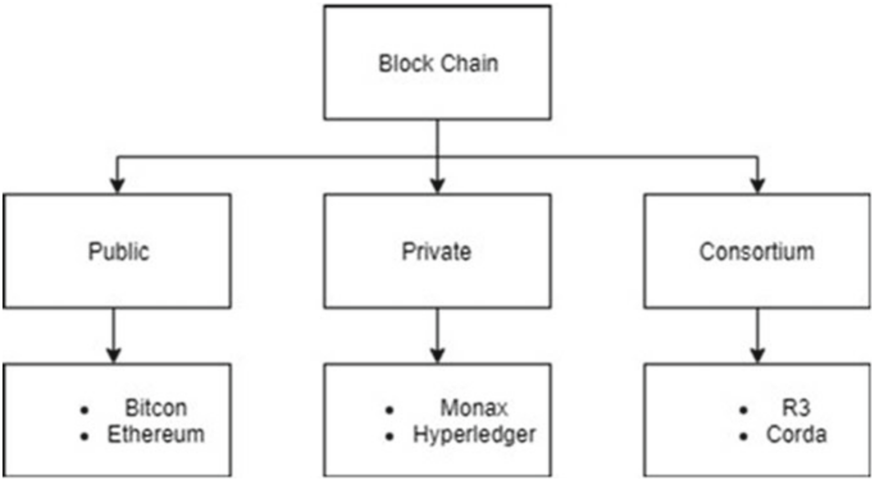


Fig. 2 Types of blockchain

1.2 Electronic Voting

Electronic voting (e-voting), which utilizes electronic systems to support casting and counting votes in an election, has remained a research topic of importance for the past several decades. In association with the conventional paper-based voting, remote e-voting is environmentally friendly, real-time counting and processing, and less error prone. Until the time and efforts to vote to reduce the overall voter attendance may progress. But such centralized e-voting system can be vulnerable to hacking, suspicious to fraud, and various manipulations. Such centralized systems are depended on the third-party to conduct the voting process and to tally the votes. Monitoring the actions performed by such systems is difficult. There is no reliability that votes counted by such systems are audited without any manipulation. Moreover, the intruder can get into the central server and alter the votes without any trace of action.

The current centralized-based e-voting system has number of disadvantages:

1. One central organization has full control over the system.
2. If intruder manipulates the data, its traces can be easily eliminated.
3. Lack of transparency.
4. It is difficult to follow the vote to check if it has been counted properly, without revealing the privacy.
5. Security is also important in fair elections as each vote needs to be secured and valued which is often not the case.

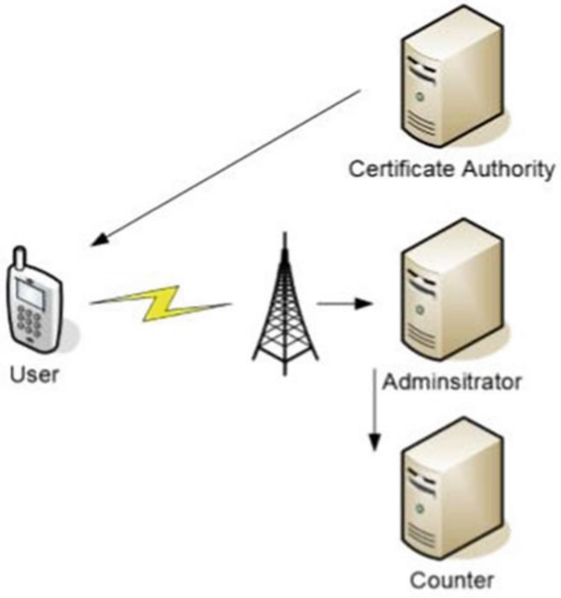
2 Literature Survey

An online voting system has been a hot subject in literature for long. Several voting models have been introduced over the years to protect the privacy of voting information. Various cryptographic tools have been used in e-voting protocol. In some cases, an assigned third-party is involved to make e-voting systems more easily to be implemented and controlled. However, a powerful third-party may also become the vulnerable spot of the whole system. Electronic voting protocol is merged with the blockchain model to design voting protocols making third-party redundant, which provides anonymity and verifiability as well.

A mobile-based voting system is proposed in [1]; here, the user"s votes are safeguarded by using elliptic curve cryptography (ECC) algorithm. ECC has a homomorphic encryption property which helps to keep the user anonymous. This property has made the ECC be more suitable to use in constraint devices. ECC is immediately used for encrypting data in the mobile device without using symmetric cryptography. As a result, it will conserve disk storage as there is no symmetric key encryption cost and encryption processing time will also be reduced while the security strength is still sustained (Fig. 3).

In [2], e-voting system is proposed through which people can vote using their smartphones or website. It is a central server-based system. In this, security is achieved using OTP approach which is usually used everywhere nowadays. Firstly, the user is required to register and authenticate himself using Voter-ID and if valid then by Aadhaar ID also. If the Aadhaar ID is also valid, then he will become a valid

Fig. 3 Mobile voting scheme [1]



voter to cast vote and will have login ID and password (cannot be changed). Voter registration and voting activity are strictly governed by system administrator. After election, voter is allowed to check whether his vote is counted or not on the result page. The benefit of this system is, no other person can cast vote for other person and no multiple voting is allowed. Also, unique identification of user through Aadhar and Voter-ID provides security and flexibility. Being a central server third-party-based system, it is prone to various vulnerabilities.

In [3], author has proposed a blockchain-based system in which the voter can change the vote in case it changes its mind, during the election-time window. Here, a centralized system is responsible for assuring that only qualified people can vote and get into the system, and every eligible user gets a token which takes the form of digital signature. It used Blockchain to store the votes, which was been encrypted using the token.

A Privacy-Preserving Voting Protocol on Blockchain is suggested by Zhang and Huang [4]; it consists of two components: client and smart contracts. The message flow among peers smart contract and ledger client holds the voting procedures that will be implemented by each particular voter. Client holds the voting operations that will be performed by each individual voter. Smart contract maintains the voting logic that requires collaboration and consensus between all voting participants. Here, the voters initially encrypt the vote, after which it is validated. These votes are been decrypted and verified, and if they are found invalid, then devoting is conducted. This process continues until all the votes are valid. At the end, votes are aggregated to tally results (Fig. 4).

In [5], the author has used blockchain to develop an e-voting model. Blind signature is used as encryption algorithm; it is used to hide the voter's choice. Due to the transparency property from blockchain, ballots are visible when they are cast to the blockchain network. This exposes the progress of the election during the voting phase, and may greatly influence the outcome of the election.

In [6], the author has proposed a Ethereum-based e-voting system; it is implemented for a small-scale system like department-level or university-level election.

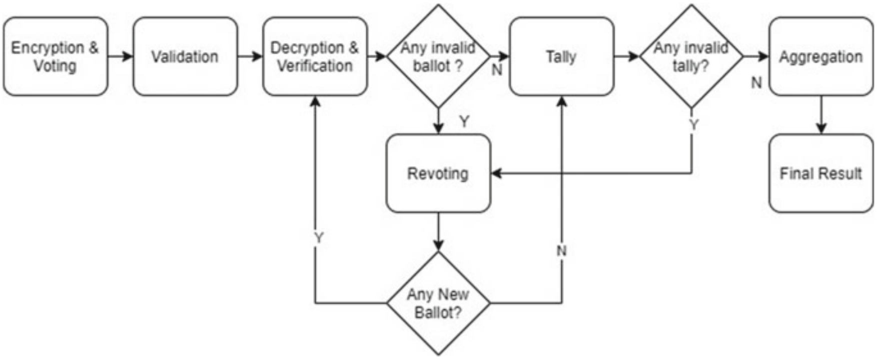


Fig. 4 Process flow [4]

Different code blocks are given which constitutes of smart contracts used for validation. Another smart contract is built to count votes at the end of the election process. The main drawback of the system is that as the number of voters increases, the time taken to cast the vote also increases.

In [7], after verification process, every node produces private and public key pair. The generated public key is distributed to all nodes listed in the system; as a result, every node maintains a public key list of every node. "Get turn method" is used to generate blocks. As a result, collision that can occur in a data transmission network is minimized. But, due to this method, the voter has to wait until its turn to cast vote which may be very tedious in real time situation, as each voter has to wait for its turn.

3 Proposed Methodology

The proposed methodology integrates blockchain paradigm into electronic polling and came up with a feasible and general e-polling idea with a high degree of decentralization. A three-tier authentication protocol is been used to ensure that vote is casted by eligible and verified candidates only. After each authentication round, the voters are certified by digital signature, which is been verified during tally phase. Votes are only been counted if they have digital signature certificate from both authorities (Fig. 5).

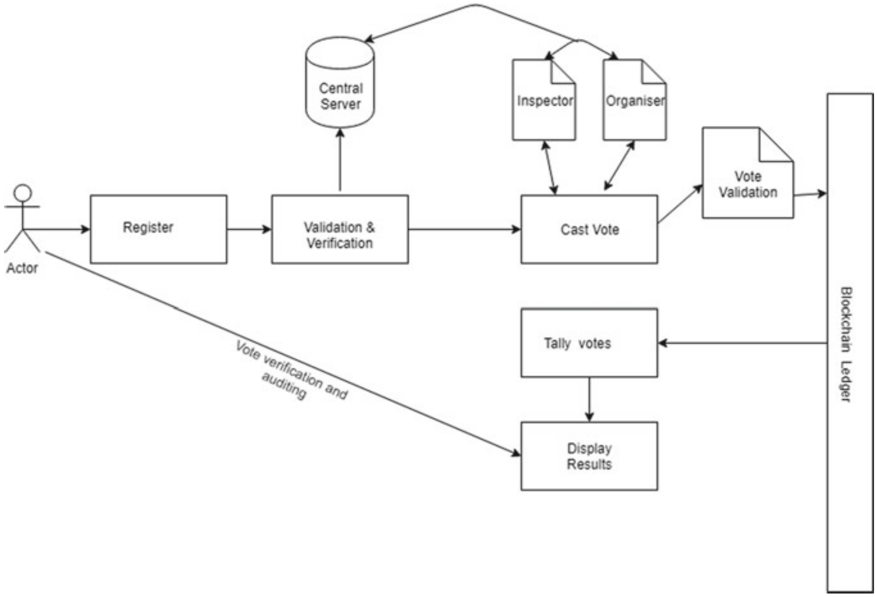


Fig. 5 Architecture diagram of proposed electronic polling agent

The system can be divided into three major modules according to its functionality:

- Identity management.
- Cryptographic privacy.
- Aggregation and auditing.

3.1 Identity Management

Identity management is the major field in blockchain, and this module determines who all can enter the system. Initially, the user needs to register into the system by providing all the required documents and credentials. This information will be stored in a separate private blockchain, which will be broadcasted to the election authority during authentication phase. The requirement of documents and information will vary according to the polling environment. After which the system will check user's identity and eligibility of voting by cross checking it with the data given by the polling organization. For experimental purpose, phone number or email id, OTP methodology will be used for verification. After users are found eligible, then they will be given access to blockchain network and required cryptographic keys will be given to them. Now, they can proceed for polling process.

3.2 Cryptographic Privacy

In any polling system, securing voters identity is the primary motive. No one should know, accept the user that to whom it cast the vote. There should be no traceable link between the vote and the voter. To ensure this kind of anonymity in the system, various cryptographic methodology and algorithm are applied. Blind signature will be the methodology used to ensure vote anonymity in this system.

Three main entities in the blind signature process are

- **Voters**: a set containing all eligible voters.
- **Organizers**: the set of the election organizer, where $|\text{organizer}| \geq 1$
  The organizer's duties are to hold the election, verify and record eligible voters' information, and associate with voters throughout the election.
- **Inspectors**: the set of all inspectors, where $|\text{Inspector}| \leq |\text{Organizer}|$
  Inspectors are introduced in order to restrict the organizer's power and inspect the organizer's behaviors. Inspectors also interact with voters throughout the election (Fig. 6).

Organizer and Inspector:

- It owns a signing function S'() which is private.
- Its Inverse S() is public.

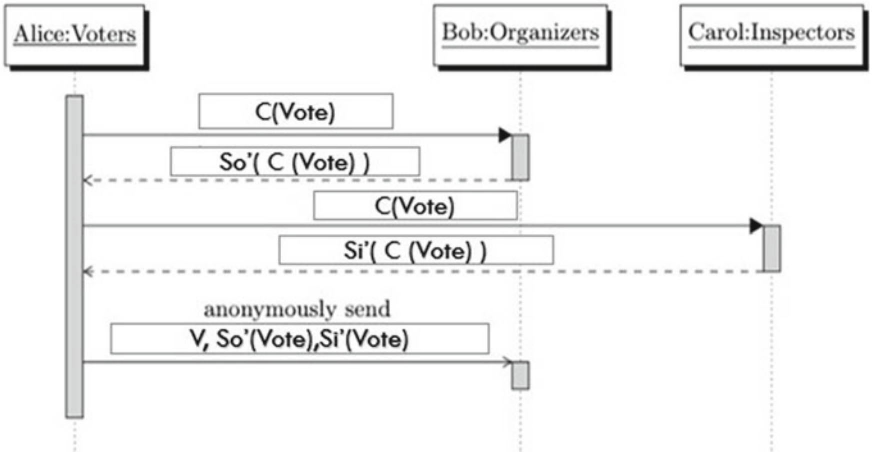


Fig. 6 Sequence diagram of blind signature

- And it has a property S(S'(X)) = X Voter.
- It owns two set of signing function C() and C'(), both are private.
- And it has a property C'(S'(C(Vote))) = S'(Vote).

3.3 Aggregation and Auditing

After the polling window timeouts, all the eligible votes are filtered and counted. With smart contract, it is ensured if the ballot has a proper digital signature of both organizer and inspector. After which the votes are tallied to display the results. After results are been displayed, inverse function of the signatures is given to the voters, with which they can decrypt the votes and audit it to verify if the results are displayed correctly.

4 Conclusion

In this paper, a privacy-preserving system using blockchain is introduced, which prunes out the need of a trusted third-party to tally votes. Blockchain enforces a trust factor into the system, as every transaction in the system is traceable and no one can wipe it out without any trace. And with the use of blind signature mythology, the voters are being verified without their votes getting relieved. The traditional E-voting system lacks this level of transparency. The voters need to trust the system blindly, on the fact that the vote which they casted is recorded and tallied intently. The proposed system has solved this issue by incorporating blockchain technology.

References

1. Ahmad T, Hu J, Han S (2009) An efficient mobile voting system security scheme based on elliptic curve cryptography. Third international conference on network and system security. © 2009 IEEE
2. Sontakke C, Payghan S, Raut S et al (2017) Online voting system via mobile. Int J Eng Sci Comput
3. Hardwick FS, Gioulis A et al. E-voting with blockchain: an evoting protocol with decentralisation and voter privacy. arXiv:1805.10258v2
4. Zhang W, Huang S (2018) A privacy-preserving voting protocol on blockchain. 11th international conference on cloud computing. © 2018 IEEE
5. Liu Y, Wang. An e-voting protocol based on blockchain. Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
6. Koc AK et al. Towards secure e-voting using Ethereum Blockchain. Comput Control Netw 978-1-5386-3449-3/18. ©2018 IEEE
7. Hanifatunnisa R, Rahardjo B, Blockchain based e-voting recording system design. 978-1-5386-3546-9/17. © 2017 IEEE

Issuing and Verifying University Certificates on Blockchain

Dhiren Patel, Balakarthikeyan Rajan, Yogesh Mangnaik, Jatin Jain, Vasu Mistry, and Pearl Patel

Abstract In this paper, we propose a decentralized approach toward issuance of institute degree and their verification system using blockchain technology. Keeping the encrypted record on a public blockchain, using Oracles, we provide a framework for quick verification of the degrees. The system introduces the concept of generating a single root hash for all the degree-encrypted hashes to avoid excessive on-chain data storage. The system also explains the different smart contract workflow required to implement the idea along with the use of ERC223 token to monetize the Oracle service.

Keywords Blockchain · Merkle Tree · Smart contract · Verification

1 Introduction

Traditional security has been ensured through access and control with an institution in the center which may exploit and misuse trust of clients and users. Blockchain is a technology which decentralizes the access and control mechanism thus avoiding central points of trust. Blockchain is able to provide a tamper evident system to ensure data security.

In this paper, we present a system of issuing and verifying degree certificates by university/institute using blockchain and show by using Oracles, a way to decrypt the encrypted certificates on the blockchain thus allowing a method to monetize the entire system. Use of decentralized blockchain and smart contract will improve the

D. Patel · B. Rajan (✉) · Y. Mangnaik · J. Jain
Veermata Jijabai Technological Institute, Mumbai 400019, India
e-mail: bkybala9@gmail.com

V. Mistry
Independent Researcher, Mumbai, India
e-mail: vasu5235@gmail.com

P. Patel
Vidyalankar Institute of Technology, Mumbai 400307, India

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_8

entire process of issuance and verification of degree certificates both on performance (time) and security (tamper proof).

Rest of this paper is organized as follows: Sect. 2 discusses basic improvements existing in certificate issuance system. In Sect. 3, we present a workflow for such system on blockchain. Section 4 discusses verification system for issued certificate. Section 5 discusses implementation architecture, with conclusions and references at the end.

2 Motivation and Background

2.1 Existing Degree Certificate Issuing and Verification Procedure

The existing system for certification issuance for the graduates and its validation is a long and cumbersome process. This paper was motivated by analyzing the current scenario in typical university/institution and is an attempt to apply blockchain technology for improving the current workflow. Most universities/institutes offer hard copies of degree certificates to their graduates. This makes verification during background checks cumbersome and typically requires the institute to check the copies and send back a signed copy attesting that these certificates are valid. A typical workflow is depicted in Fig. 1. Institutes also generally charge some money to process such verification requests and on an average, this process might take one to three weeks. To alleviate this problem, we propose a decentralized solution using blockchain technology to allow verification of such certificate data. This also means that now student certificates are uploaded onto the blockchain network. This gives the added benefit that students can now supply their certificates or educational record for verification without the use of any hard copy. Such verification can be done seamlessly via smart contracts which will also charge a small fee for verification. The smart contract provides a secure way to verify whether the given submitted certificate is genuine or not and collects some fees in the form of tokens as payment for this service.

2.2 Blockchain

Blockchains are distributed digital ledgers of cryptographically signed transactions that are grouped into blocks. Each block is linked to the previous one after validation and consensus of all participating nodes. As new blocks are added, older blocks become more difficult to modify. New blocks are replicated across all copies of the ledger in the network, and any conflicts are resolved automatically using established rules [1].

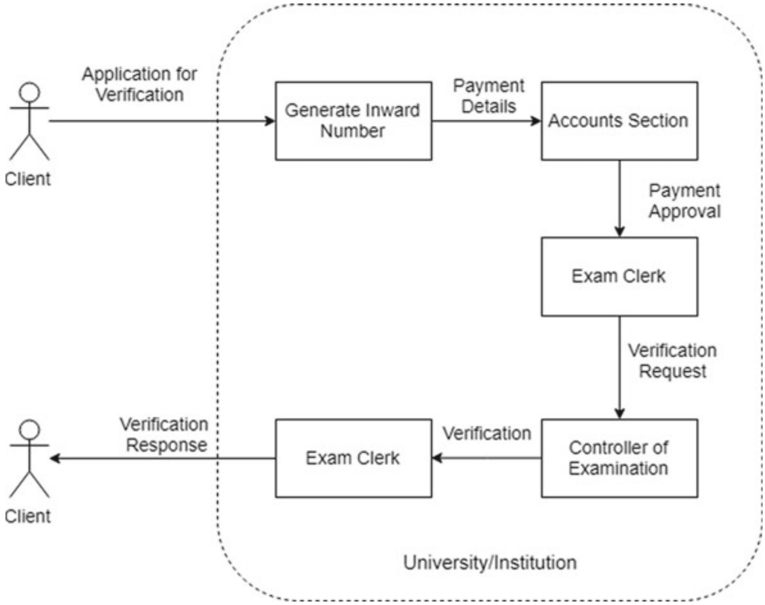


Fig. 1 Typical workflow for degree certificate verification in university/institute

At their most basic level, blockchain enables a community of users to record transactions in a ledger that is public to that community, such that no transaction can be changed once published. A block is an individual unit of a blockchain, composed of a collection of transactions and a block header. A block header keeps a collection of metadata about the block that contains a hash-value of its parent in the blockchain, and a hash of the aforementioned metadata and the data of the block itself [2].

A public blockchain network is completely open and anyone can join and participate in the network. The network typically has an incentivizing mechanism to encourage more participants to join the network. Ethereum is a turing complete programmable public blockchain platform [3].

2.3 Ethereum Public Blockchain

Ethereum is an open blockchain platform that lets anyone build and use decentralized applications that run on blockchain technology. Ethereum is a programmable blockchain. Rather than giving users a set of predefined operations (e.g., bitcoin transactions), Ethereum allows users to create their own operations of any complexity they wish. In this way, It serves the purpose for many different types of decentralized blockchain applications, including but not limited to cryptocurrencies. Ethereum Virtual Machine ("EVM") can execute code of arbitrary algorithmic complexity. In computer science terms, Ethereum is “Turing complete.” Developers can create applications that run on the EVM using friendly programming languages modelled on existing languages like JavaScript and Python [4].

A contract is a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain. Contract accounts are able to pass messages between themselves as well as doing practically turing complete computation. Contracts live on the blockchain in an Ethereum-specific binary format called Ethereum Virtual Machine (EVM) bytecode, and contracts are typically written in some high-level language such as Solidity and then compiled into bytecode to be uploaded on the blockchain.

2.4 Merkle Trees and Merkle Proofs

A tree constructed by hashing paired data (the leaves), then pairing and hashing the results until a single hash remains, the Merkle Root. Merkle Tree is a tree in which every leaf node is labelled with the hash of a data block and every non-leaf node is labelled with the cryptographic hash of the labels of its child nodes. Hash trees allow efficient and secure verification of the contents of large data structures. Hash trees are a generalization of hash lists and hash chains. Demonstrating that a leaf node is a part of a given binary hash tree requires computing a number of hashes proportional to the logarithm of the number of leaf nodes of the tree (Fig. 2).

Merkle Proofs are established by hashing a hash’s corresponding hash together and climbing up the tree until you obtain the root hash which is or can be publicly

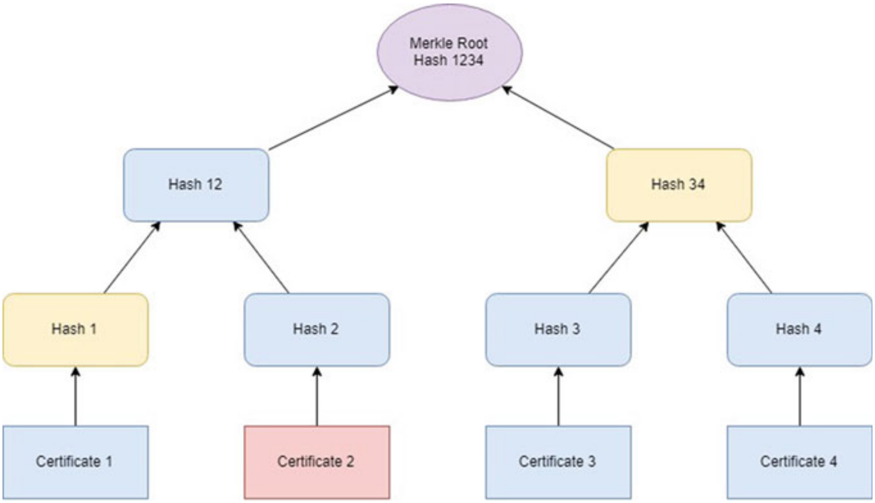


Fig. 2 Merkle Tree

known. Given that one-way hashes are intended to be collision free and deterministic, no two plaintext hashes are the same [5].

Assuming such a hash-function which we shall call as H, then

$$\begin{aligned} \text{Hash 1} &= \text{H (Certificate 1)} \\ \text{Hash 2} &= \text{H (Certificate 2)} \\ \text{Then Hash 12} &= \text{H (Hash 1 | Hash 2)} \\ \Rightarrow \text{Hash12} &= \text{H (H (Certificate1) | H (Certificate 2))} \end{aligned}$$

Merkle Proofs are used to decide upon the following factors:

- If the data belongs to the Merkle Tree.
- To concisely prove the validity of data being part of a data set without storing the whole data set.
- To ensure the validity of a certain data set being inclusive in a larger data set without revealing either the complete data set or its subset.

2.5 Oracle

An oracle is an agent that finds and verifies real-world occurrences and submits this information to the blockchain to be used by smart contracts.

Oraclize service allows smart contracts to connect and obtain results from different services like IPFS, WolframAlpha.

- oraclize_query(“IPFS”, “<file-hash>”) will retrieve the contents stored on the IPFS network
- oraclize_query(“WolframAlpha”, “flip a coin”) will return either “heads” or “tails” to the smart contract [6].

Oracle allows smart contracts to interact with the outside world using APIs.

Oracalize.it [6] is a service and library for Solidity which provides Oracle services. Oracles attest for the proof of incoming information from outside a blockchain. The oracle can be thought of as a gateway which provides secure communication between a blockchain and the rest of the Internet.

Today, most universities/institutes issue paper certificates to students who have passed their offered courses. Issuing paper certificates involves a properly designed process to ensure certificates are not temperable and are not illegally duplicated. Also one has to ensure an easy way to identify and attest to the validity of a certificate. That is, we need to authenticate and validate certificates. Many digital courses allow online validation of issued certificates to check their genuineness. The cost for this in terms of infrastructure and facilitation is passed onto students. Companies also spend time and money themselves or third-party agencies to verify authenticity of students joining them and the validity of their degrees. The system proposed in this paper aims to address the problems by making these certificates available on a public blockchain and writing appropriate smart contract codes to publish and verify these certificates. The verification process can be built in such a way that a small fee would be needed to be paid by the requester to assess the genuineness of the certificate. The certificate now can be just given as a small string to the holder (a hashed certificate) rather than a pdf document. This would make the verification process transparent, trustable, and faster.

One such system has already been implemented by the Massachusetts Institute of Technology [8]. The existing system allows verification code to be run independently by any party and provides a sample code and system to be based on. In this paper, we explore the creation of such a system where verification process and certificate issuance are delegated to a smart contract on the Ethereum network and an Oracle service. This allows institutes greater control on the process and can design their own verification algorithms, and also enables institutes to charge fees as in the former system for verification requests. We also try to achieve this with minimal input from the side of the requester and the student. Since the system using encrypted hashes to construct the Merkle Tree, an Oracle service is used to get the encrypted hash. Thus, this system now ensures that there is a fee charged to the requester for the verification by the use of encrypted hashes and decrypting them using oracles

3 Workflow for Degree Certificate Issuance System on Blockchain

3.1 Digitizing the Certificates

Certificates need to be stored in a convenient digital format. In this case, we shall create a simple JSON schema to represent the certificate.

The following is a representative schema for the certificate

```
.certificate {
full_name: String,
institute: String,
cgpa: Double,
student_id: String,
batch: Int
hash: SHA256
proof: List of SHA256
}
```

The hash field here is populated as a SHA256 of full_name, institute, cgpa, student_id, and batch only. Let us call a subset of the above fields as a minimal certificate. The following minimal certificate is then hashed to get hash H. The H is re-hashed with the private key of the institute to get an encrypted hash E. This procedure shall now be repeated for every certificate making up a batch. For example, all certificates for the Batch of 2017 shall be generated and for each certificate, the hash E shall be computed as described above. After this, a Merkle Tree is generated for the entire batch and its Merkle Root M is found out in the next steps. The proof field includes the Merkle Path of these encrypted hashes which will be used as a check to see whether the certificate has been included by the Merkle Root or not.

3.2 Generating Certificate Blocks

Once the Encrypted Hashes E for every certificate is known, each of them can be directly put on to the blockchain. But this will involve a lot of excess computation and costs, each transaction costs gas fees and thus, one/the institute would end up spending a lot of money due to the fact that there are many certificates to be published. To overcome this, we use all of these encrypted hashes to create the Merkle Tree. The root of this tree called M, is known as Merkle Root and it is representative of the elements which made up this root. Specifically, Merkle Paths allow us to create a Merkle Proof where in knowing the Merkle Root, the corresponding Merkle Path to a leaf and the data of the leaf, we can verify whether the leaf was part of this root or not. Thus, we can securely say that the hash (i.e., whether the certifcate) was part of the root or not. Thus, now our transaction is reduced to only one, which is a single transaction containing the Merkle Root can be published. We shall now add this Merkle Path to the proof field of our digital certificate. Once every certificate has its proof field added, we can email this to each candidate. This ends the process from the side of the institute.

3.3 Publishing the Certificates on the Blockchain

The Merkle Root generated will now be added to the blockchain. To enable this we call a function on our smart contract with this Merkle Root as the data. The contract will accept this transaction and fire an event; tagging this transaction if and only if the caller is the contract deployer which in this case will be the university/institute. This published transaction will be mined by one of the miners on the ethereum network and included in some block that will be added to the blockchain. As the time goes on, more blocks will be mined and included in the blockchain. As more blocks are appended to the block containing the published transaction, the harder it is to change it. There is a certain number of blocks after which the transaction is considered to be immutable. Every block appended is considered to be a verification for the previous blocks.

4 Verification of Issued Certificates

We shall provide a simple smart contract function which shall return a Boolean value true or false depending on the status of the verification. The verification process follows the following steps. Before submitting the hash for verification to the smart contract, we run a small client-side check to ensure that the hash field in the certificate matches the hash of the certificate. The contract will receive only the payment tokens, certificate hash field, and the Merkle Proof field.

1. The smart contract first finds the transaction associated with this batch, since every batch had a different Merkle Root.
2. The smart contract verifies the Merkle Root transaction's issuer's identity. It checks if the transaction containing the Merkle Root was signed by the private key of the institute/university.
3. The contract requests the Oracle service for the encrypted hash for the certificate hash given as input.
4. Next, the contract builds the Merkle Root using the path info and checks whether this generated Merkle Root is same as the one in the transaction. This finally ensures that the certificate was part of the Merkle Root (Figs. 3, 4, 5 and 6).

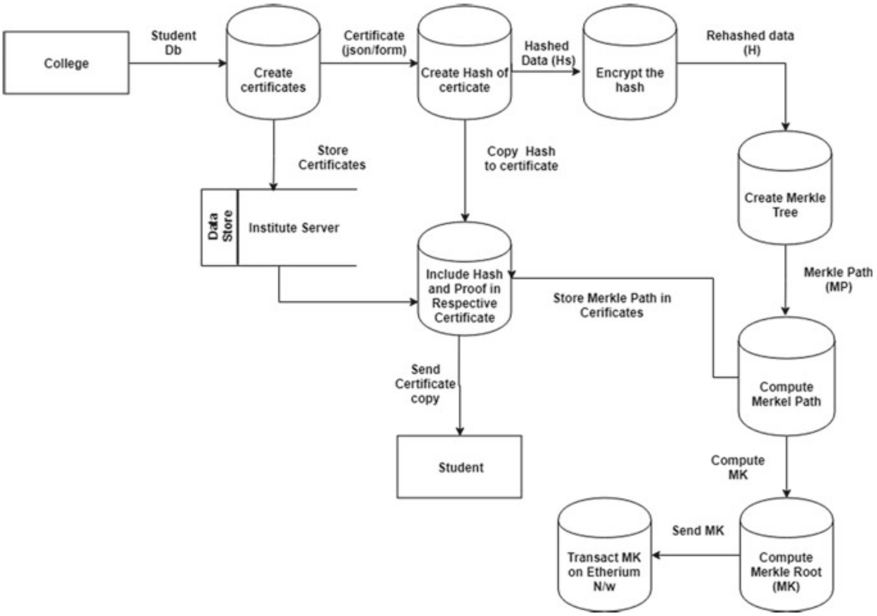


Fig. 3 Data flow diagram showing issuance of the certificate by the institute/university

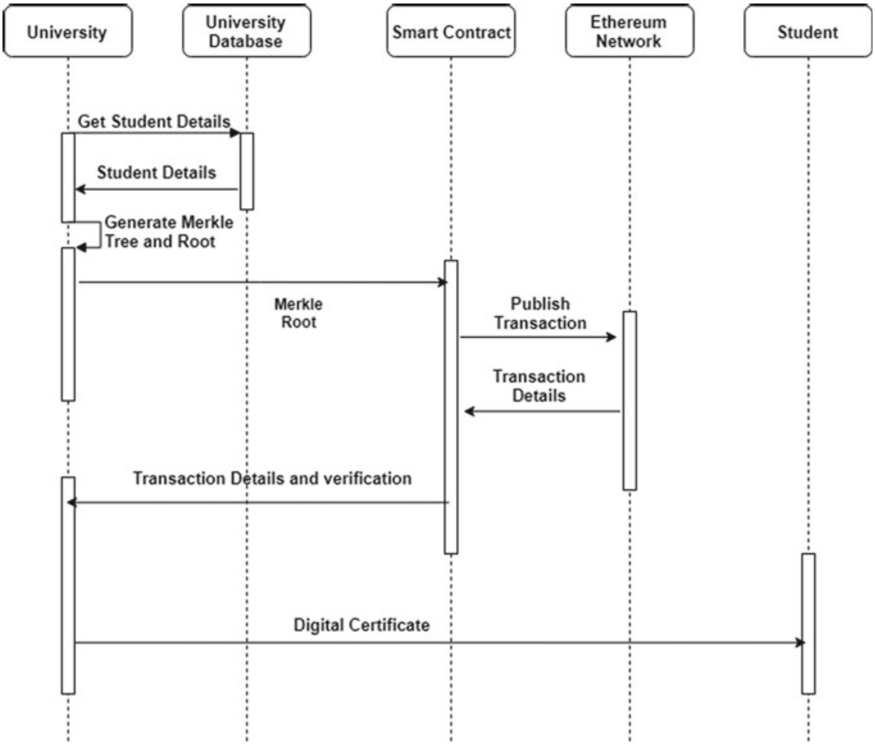


Fig. 4 Sequence diagram showing issuance of the certificate by the institute/university

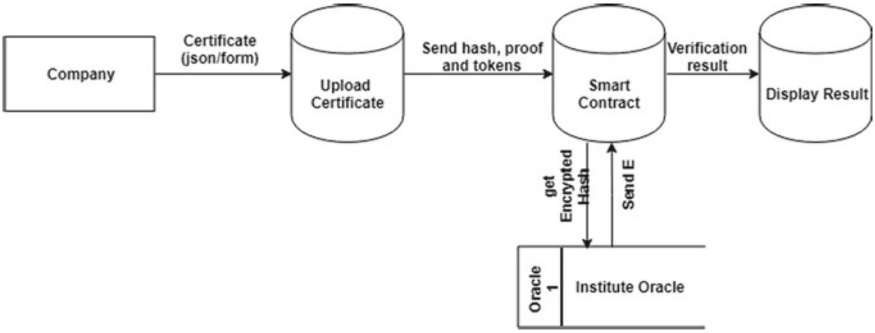


Fig. 5 Data flow diagram showing verification of the certificate by company

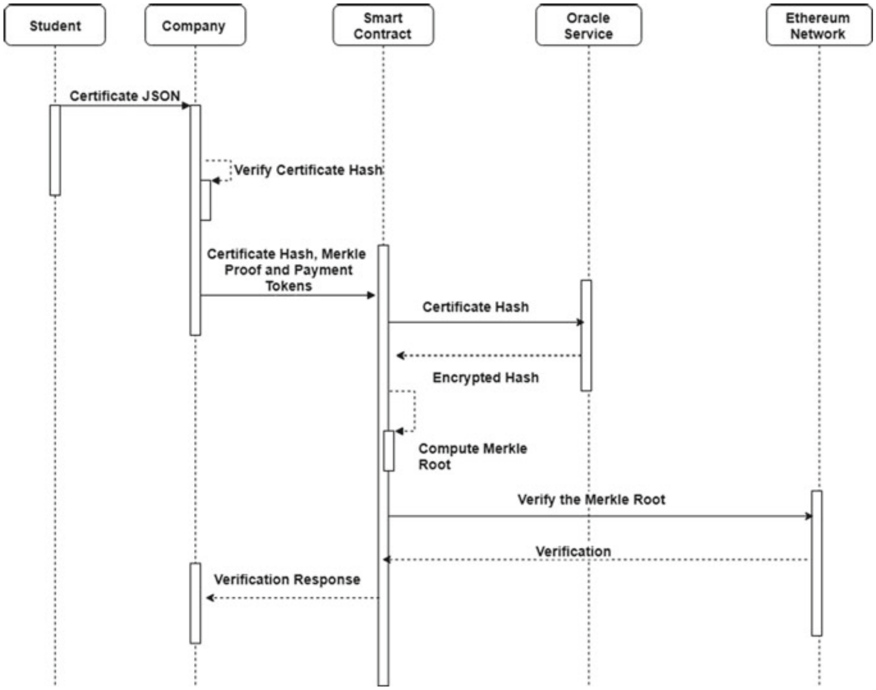


Fig. 6 Sequence diagram showing verification of the certificate by company

5 Implementation Architecture

5.1 Smart Contracts

The deployment will include smart contracts to facilitate publication of the certificate to the blockchain and also the verification of published certificates. The ***publish*** function would take the signed transaction with the Merkle Root and record it publically. This function can be only called by the contract deployer that is the institute/university can only publish the certificates. A mapping will be recorded between each batch and the transaction containing its Merkle Root allowing easy access to that year's certificates.

The second function ***verify***, shall be accessible to any verifier and accepts as fees a set of predefined tokens to carry out the verification. It shall return only a Boolean Yes/No and an optional message detailing failure if needed. This function shall accept the given certificate which shall contain the hash and the Merkle Proof to validate the authenticity of the certificate. It performs tasks as outlined in the verification steps. The tokens for the same can be pulled out of the client (verification request submitter) Metamask [9] wallet.

5.2 Server-Side Deployment

This is the only code which shall be maintained on the server by the institute. This code contains the service which is responsible for loading the form which the end-user will utilize for the verification process. It can also contain additional code to collect statistics and other data.

5.3 Payment Gateway

To monetize the process of verification of certificates, tokens are accepted by the smart contract before disclosing the results of such validation as payment. To generate these tokens, we shall deploy using the institute address ERC223 tokens [8]. A payment gateway shall accept money (real currency in INR) and instruct our deployment of the ERC223 tokens smart contract to issue equivalent tokens to the recipient address. This shall be a transaction called by the institute to the recipient. The tokens shall be reflected in the recipients Metamask wallet [7].

5.4 Oracle Service

We would need to implement an Oracle service which listens for an event and returns the corresponding encrypted hash for the given Hash. This service can be implemented using Oracalize.it [6]. The Oracle will prevent the requester from directly knowing the encrypted hash (Fig. 7).

The architecture is implemented in a way so as to allow the university/institute to maintain control over the process of verification and monetize it in a way it deems to be fit. It is also made in a way to store least amount of information at the institute's end. It also minimizes processing at the servers maintained by the institute. Thus, the architecture ensures that minimal trust is put on the issuing university/institute and allow for a decentralized verification of certificates.

6 Conclusions

With the rising increase of educational certificate fraud and misuse, it becomes imperative to design an easy to use trustless, decentralized validation system to verify authenticity of certificates and to make student digital certificates tamper proof in nature. We have looked at an interesting use case and implemented a solution on the Ethereum public blockchain to ensure tamper-proof certificate issuance and to verify their authenticity. This also reduces time and efforts spent by the institute in

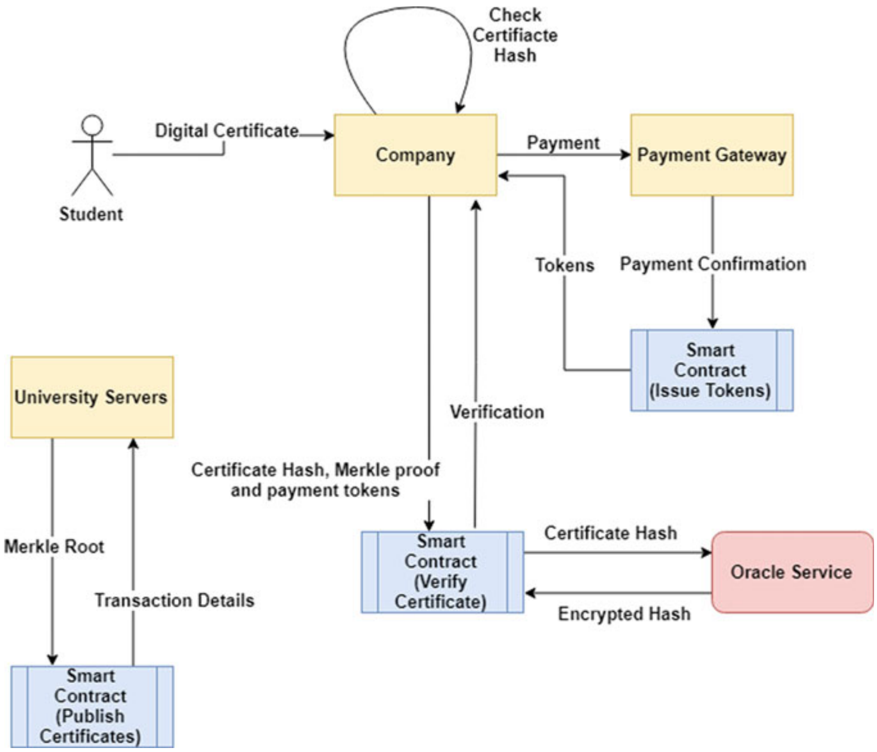


Fig. 7 Implementation architecture diagram for certificate issuance and verification system

verification of their certificates while still allowing them to monetize the system. The system also ensures that minimal trust is needed on the institute/university for verification of the issued certificates.

References

1. Yaga D, Mell P, Roby N, Scarfone K (2018) Blockchain technology overview. Draft NISTIR 8202, NIST, US
2. Wurster S et al (2017) Specification on Blockchain Technology. ISO/TC 307, Tokyo
3. Ethereum Project. https://www.ethereum.org/
4. Becker G (2008) Merkle signature schemes, Merkle trees and their cryptanalysis. Ruhr-University Bochum, Technical Report
5. Oraclize-blockchain oracle service, enabling data-rich smart contracts. http://www.oraclize.it/
6. Digital Certificates Project. http://certificates.media.mit.edu/
7. MetaMask Ethereum Browser Extension. https://metamask.io/

8. Ethereum: ERC223 token standard—Issue#223—ethereum/EIPs. https://github.com/ethereum/EIPs/issues/223
9. Cheng JC, Lee NY, Chi C, Chen YH (2018) Blockchain and smart contract for digital certificate. 2018 IEEE International Conference on Applied System Invention (ICASI). https://doi.org/10.1109/icasi.2018.8394455

Integrating Blockchain with Local Public Service System

Vinayak Ashok Bharadi, Purva Pramod Ghag, Sukanya Rupesh Chavan, Shivani Shivram Gawas, and Atiya Kazi

Abstract A smart city is having various types of offices, which are having different types of data. These departments are related to each other. In some cases, if the user wants to access a particular type of data of any local governance department then that data is available for the user, during the process sometimes the data may be hacked and modified by third party entity. Therefore, for removing such drawback, we are using blockchain technology here. In this paper, a blockchain-based system is proposed for bringing trust and integration between different subunits of local public service systems. The existing systems are working in silos; the proposed system uses Azure Blockchain Workbench for integrating such units and bringing sync and trust between them. This results into faster and secure operation at both the administrative as well as consumer (public) end.

Keywords Blockchain · Smart contract · Peer-to-peer · Ledger

1 Introduction

As we know nowadays, the sensitive and confidential information is hacked by any unauthorized user. The unauthorized user will manipulate the data and that data will be a breach in some cases. To overcome this hacking we are using trusted

V. A. Bharadi (✉) · P. P. Ghag · S. R. Chavan · S. S. Gawas · A. Kazi
Department of Information Technology, Mumbai University, Finolex Academy of Management and Technology, Ratnagiri, Maharashtra 415639, India
e-mail: vinayak.bharadi@famt.ac.in

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_9

technology called blockchain. Blockchain is the general-purpose technology, which is most important for security purpose. Creating blockchain for local governance with Microsoft Azure is a secure way to exchange information between servers and client, which is required for authentication and trust [1]. In smart cities, the government provides different types of services to citizens. Government-related blockchain applications are including digital identity, the storing of important and fair decisions, marital status, business licenses, passports, criminal records, and even tax records. Further research is recommended to compare the variety of initiatives and to analyze the source of benefits. In our system, a user will request for collection of data from citizens, devices, and assets. The requested data can be broadcast to peer-to-peer network consisting of computers, known as nodes. This network of nodes validates this collection of data and user's status using password verification algorithms. This verified data can involve data belonging to transportation systems, law enforcement, water supply networks, waste management, hospitals, and other community services. Once data is verified, the data is combined with other data to create a new block of data for the blockchain ledger. The new block is then added to the existing blockchain, in a way that is permanent and unalterable and then this data can be forwarded to any requesting user on a timely basis [2]. We cannot modify the data once the blockchain is created otherwise the whole chain of blocks consisting of data will be a collapse. Single database is replaced by distributed ledger, which consists shared information that information is restricted and provides high security and accessibility.

1.1 What Is Blockchain Technology?

A blockchain is a growing list of records called blocks, which are linked to each other using a cryptography system. Each block contains cryptographic hash of the previous block, a timestamp, and data. A blockchain is resistant to modification of data. An open distributed ledger can record transactions between two parties. For use of distributed ledger blockchain, manage peer-to-peer networks collectively adhering to protocol for internode communication and validating new blocks. Satoshi Nakamoto invents a blockchain in 2008 to serve the public transaction ledger of the cryptocurrency bitcoin [3]. Blockchain technology can collect multiple areas. The primary use of this technology is distributed ledger for cryptocurrencies. Public blockchain and private blockchain are the two types of the blockchain. In public blockchain, it has no access restrictions and the private blockchain is permissioned that is the private blockchain is restricted (Fig. 1).

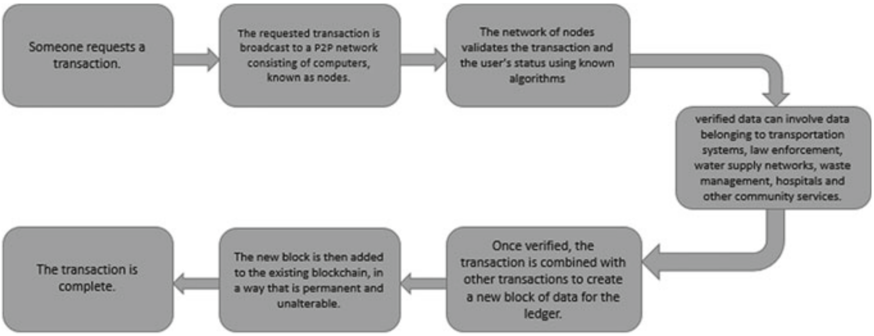

Someone requests a transaction.
The requested transaction is broadcast to a P2P network consisting of computers, known as nodes.
The network of nodes validates the transaction and the user's status using known algorithms
verified data can involve data belonging to transportation systems, law enforcement, water supply networks, waste management, hospitals and other community services.
Once verified, the transaction is combined with other transactions to create a new block of data for the ledger.
The new block is then added to the existing blockchain, in a way that is permanent and unalterable.
The transaction is complete.

Fig. 1 Functionality of blockchain

2 Existing System

Banarjee [4] proposed a blockchain-based SCM for bringing transparency and authenticity. At Infosys, they implemented the blockchain protocol with a three-layer architecture, based on Oracle and SAP at the application layer and the blockchain at the bottom layer. Huckle et al. [5] explored how the Internet of Things and blockchain technology can benefit shared economy applications. The focus of this research understood how blockchain could be exploited to create decentralized, shared economy applications that allow people to monetize, securely, their things to create more wealth. Here the authors have proposed a system based on blockchain and integrating it with the local governance. The systems under consideration are registration of birth and death, police station, marriage registrar, ration card at the district level.

In the current system for getting birth- and death-related data, which can be filled handwritten way and with the help of technical equipment like typewriters or computers. If the user wants the head of registers or other public authorities who verify these records and documents will sign, get birth and death records, then it. In case, user loses the records then the new record is generated by the registry with the original receipt and written application [6]. However, this exchange of transaction is not secured in some cases. Further, the different subunits are working in separation and many times their transactions are independent on an operational basis but connected or dependent on a legal basis. Therefore, we have to make transactions integrated, secure and prevent access from unauthorized users and avoiding data manipulation and deletion.

3 Proposed System

In the paper, a blockchain-based approach is proposed for bringing trust and integrity between different departments of the local government subunits at district level.

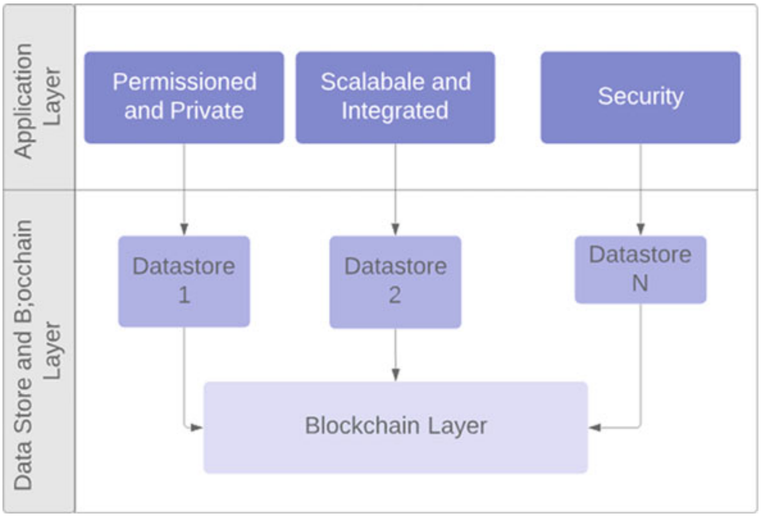


Fig. 2 Proposed system layered view

Police department, Marriage Registrar, Birth and Death registrar, Ration Card office, and Local maternity homes are the different units or the nodes of the blockchain. This will be a three-layer system. At the top, there will be the application layer, later the distributed data storage layer and the underlying blockchain layer as shown below (Fig. 2).

For the timely collection of data, user will log into a system that will act as the registration. A Portal is linked to a smart contract in the blockchain network. The user will use a unique reference number (URN) for logging in. All relevant information (hospitals like birth, death, and stillbirth) shall be updated to editable fields in the smart contract. The smart contract moves to peer-to-peer node, for the validation and review of the information and the user will digitally sign the data. The smart contract forwarded to the admin's node, where the admin validates each field in the 'smart form' with a binary response (valid or invalid). If all fields in the smart form are valid, a unique digital hash value associated with that form is generated, which is stamped to the record and updated to the blockchain. A corresponding record will be generated as a data/file that will now be updated to the citizen's similar account, printed with a unique digital hash value [2] (Figs. 3 and 4).

4 Implementation and Results

The system is implemented on Azure Blockchain Workbench. The Azure Blockchain Workbench is the fastest way to get started with blockchain on Azure. This tool allows developers to deploy a blockchain ledger along with a set of relevant Azure services

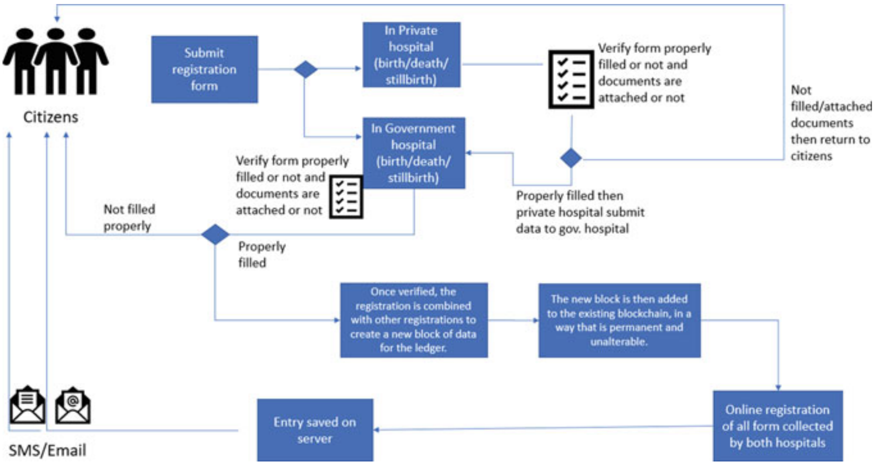

Citizens
Submit registration form
In Private hospital (birth/death/ stillbirth)
Verify form properly filled or not and documents are attached or not
Not filled/attached documents then return to citizens
In Government hospital (birth/death/ stillbirth)
Verify form properly filled or not and documents are attached or not
Properly filled then private hospital submit data to gov. hospital
Not filled properly
Properly filled
Once verified, the registration is combined with other registrations to create a new block of data for the ledger.
The new block is then added to the existing blockchain, in a way that is permanent and unalterable.
Online registration of all form collected by both hospitals
Entry saved on server
SMS/Email

Fig. 3 Example of smart contract

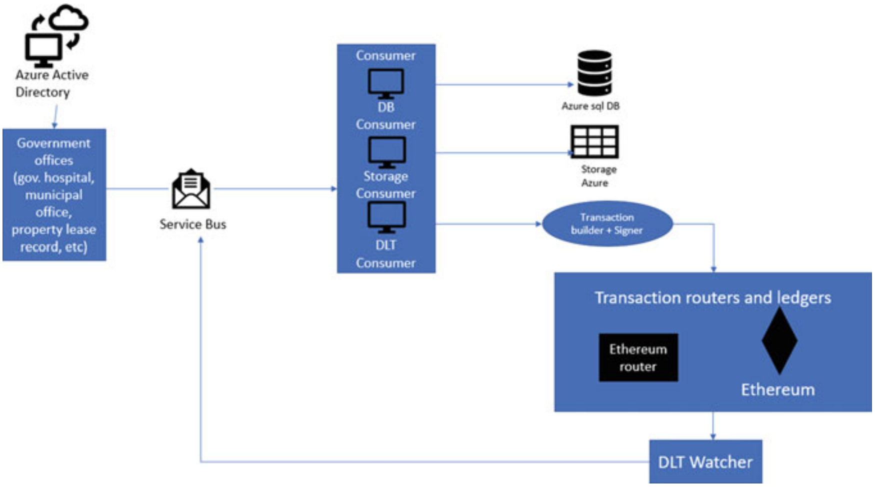

Azure Active Directory
Government offices (gov. hospital, municipal office, property lease record, etc)
Service Bus
Consumer
DB Consumer
Storage Consumer
DLT Consumer
Azure sql DB
Storage Azure
Transaction builder + Signer
Transaction routers and ledgers
Ethereum router
Ethereum
DLT Watcher

Fig. 4 Proposed architecture of blockchain on Microsoft Azure workbench

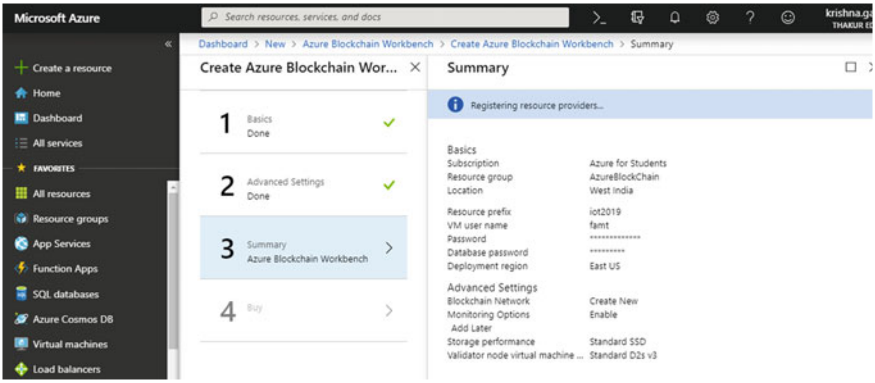
most often used to build a blockchain-based application. The blockchain has the following Azure services being provisioned

- 1 App Service Plan (Standard)
- 1 Application Insights
- 1 Event Grid Topic
- 2 Azure Key Vaults
- 1 Service Bus Namespace
- 2 SQL Databases (Standard S0)
- 2 Azure Storage accounts (Standard LRS)
- 2 Virtual Machine scale sets (ledger nodes and workbench microservices)
- 2 Virtual Network resource groups (each with Load Balancer, Network Security Group, Public IP Address, Virtual Network)
- Optional: Azure Monitor.

The central node on the Azure Blockchain Workbench has the following configuration:

Validator node virtual machine size—Standard D2s v3
Number of virtual CPUs—4
Load balancing
RAM—16 GB
Storage performance—Standard SSD (Fig. 5).

The blockchain was configured with the JASON configuration file.



1 Basics Done ✓
2 Advanced Settings Done ✓
3 Summary Azure Blockchain Workbench
4 Buy

Registering resource providers...

Basics
Subscription Azure for Students
Resource group AzureBlockChain
Location West India
Resource prefix iot2019
VM user name famt
Password *************
Database password *********
Deployment region East US

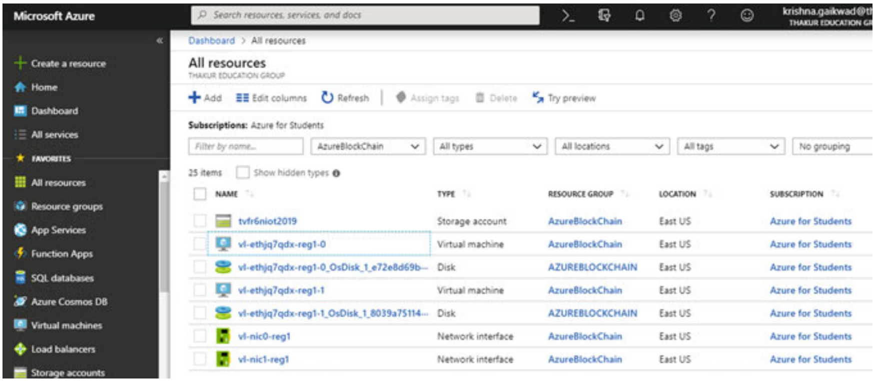
Advanced Settings
Blockchain Network Create New
Monitoring Options Enable
Add Later
Storage performance Standard SSD
Validator node virtual machine ... Standard D2s v3

Fig. 5 Deployment of blockchain on Microsoft Azure Blockchain workbench

```
{
  "ApplicationName": "MahaeSevaBlockChain",
  "DisplayName": " MahaeSeva",
  "Description": "Integrating Local Government Services on
Blockchain",
  "ApplicationRoles": [
    {
      "Name": "Civil Hospital Maternity",
      "Description": "Details of Birth."
    },
    {
      "Name": "MahaeSeva",
      "Description": "Birth Certificate Issuing Authority"
    }
  ],
  "Workflows": [
    {
      "Name": "Civil Hospital",
      "DisplayName": "Report of Deaths Happened",
      "Description": "Records of Deaths in the locality",
      "Initiators": [ "Requestor" ],
      "StartState": "Request",
      "Properties": [
        {
          "Name": "State",
          "DisplayName": "State",
          "Description": "Holds the state of the contract.",
          "Type": {
            "Name": "state"
          }
        },
```

The abovementioned code sniplets describes typical contract for the maternity ward and civil hospital mortuary. When one incident of birth or death is reported, it will trigger a contract on blockchain. The different subunits of Azure Blockchain are listed in the following figure; this is a snapshot of Live Blockchain on Azure Cloud (Figs. 6, 7 and Table 1).

This shows the functionality of the blockchain in action, further the smart contracts have to be deployed on the blockchain and integrated with all the functionality of every unit's node.



All resources
THAKUR EDUCATION GROUP

Subscriptions: Azure for Students

25 items

NAME	TYPE	RESOURCE GROUP	LOCATION	SUBSCRIPTION
tvfr6niot2019	Storage account	AzureBlockChain	East US	Azure for Students
vl-ethjq7qdx-reg1-0	Virtual machine	AzureBlockChain	East US	Azure for Students
vl-ethjq7qdx-reg1-0_OsDisk_1_e72e8d69b...	Disk	AZUREBLOCKCHAIN	East US	Azure for Students
vl-ethjq7qdx-reg1-1	Virtual machine	AzureBlockChain	East US	Azure for Students
vl-ethjq7qdx-reg1-1_OsDisk_1_8039a75114...	Disk	AZUREBLOCKCHAIN	East US	Azure for Students
vl-nic0-reg1	Network interface	AzureBlockChain	East US	Azure for Students
vl-nic1-reg1	Network interface	AzureBlockChain	East US	Azure for Students

Fig. 6 Live Azure Blockchain

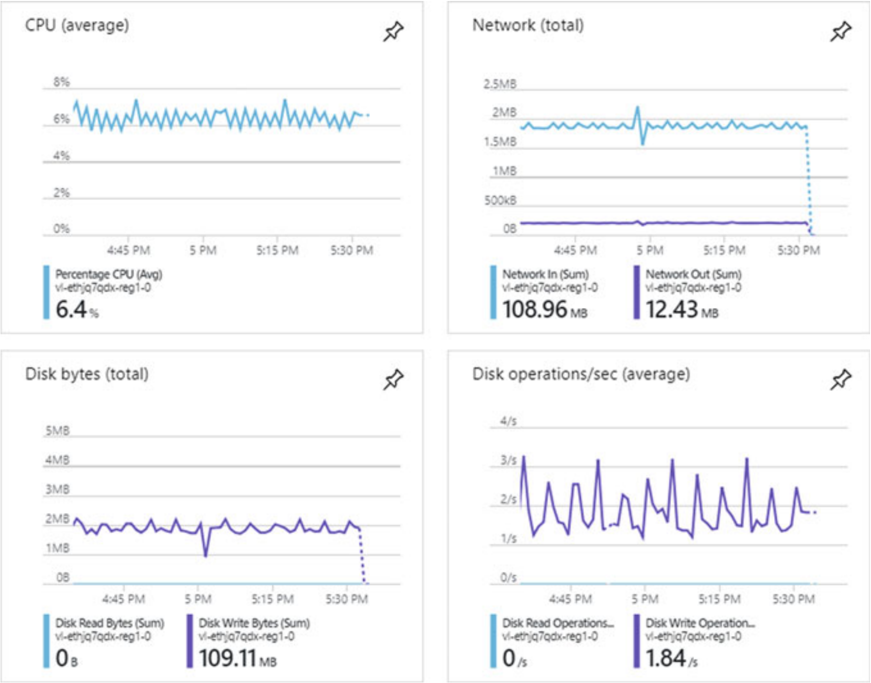


Fig. 7 Node parameters of blockchain in operation

Table 1 Azure Blockchain components

Service name	Type	Resource group
Network security group	Azure Blockchain	Azure Blockchain
ethjq7qdx-vnet-reg1	Virtual network	Azure Blockchain
ethjq7qdxstore	Storage account	Azure Blockchain
iot2019-e.g.-tvfr6n	Event grid topic	Azure Blockchain
iot2019-lb	Load balancer	Azure Blockchain
iot2019-lb-public-ip	Public IP address	Azure Blockchain
iot2019-plan	App service plan	Azure Blockchain
iot2019-sb-tvfr6n	Service bus namespace	Azure Blockchain
iot2019-subnet-workers-nsg	Network security group	Azure Blockchain
iot2019-tvfr6n	Application Insights	Azure Blockchain
iot2019-vnet	Virtual network	Azure Blockchain
poaAvailabilitySet-reg1	Availability set	Azure Blockchain
tvfr6n-iot (db-tvfr6n-iot/tvfr6n-iot)	SQL database	Azure Blockchain
tvfr6niot2019	Storage account	Azure Blockchain
vl-ethjq7qdx-reg1-0	Virtual machine	Azure Blockchain
vl-ethjq7qdx-reg1-0_OsDisk_1_e72e8d69b92548bead463d0d665f089c	Disk	Azure Blockchain
vl-ethjq7qdx-reg1-1	Virtual machine	Azure Blockchain
vl-ethjq7qdx-reg1-1_OsDisk_1_8039a75114e340eba85d9dbbe1e71364	Disk	Azure Blockchain
vl-nic0-reg1	Network interface	Azure Blockchain
vl-nic1-reg1	Network interface	Azure Blockchain
db-tvfr6n-iot	SQL server	Azure Blockchain

(continued)

Table 1 (continued)

Service name	Type	Resource group
ethjq7qdx-akv	Key vault	Azure Blockchain
ethjq7qdx-lbpip-reg1	Public IP address	Azure Blockchain
ethjq7qdx-oms	Log analytics workspace	Azure Blockchain
ethjq7qdx-vlLb-reg1	Load balancer	Azure Blockchain
ethjq7qdx-vlNsg-reg1	Network security group	Azure Blockchain
ethjq7qdx-vnet-reg1	Virtual network	Azure Blockchain

5 Conclusion

In this paper, we investigated how blockchain is innovative and used to access transactions in peer-to-peer networks. It provides transparency and avoiding fraud and corruption in smart cities. Information is accessed in an easy manner, which enhances the speed of access, which increases efficiency and transaction manners. The meaning of the word ‘smart’ which converted to ‘smart city’ which can help us to understand the requirements or needs of smart cities and which technology is used to adopt that requirements.

The blockchain technology contributes to make city smart by providing services for citizens. In this paper, we first gave an overview of blockchain technology [7]. Then we discussed what the needs of blockchain technology in local governance are and how it is implemented by using Microsoft Azure cloud service.

Following are the points, which can be modified, in the existing system:

1. Currently, we are working with Microsoft Azure for making blockchain. Instead of that, we can develop our private blockchain for our system.
2. Now we have created our blockchain using single machine and in upcoming days, we can use multiple nodes for our blockchain network.
3. Detailing of Smart contracts and deployment on the blockchain.
4. Database setup and integrating with the blockchain.

References

1. Olnes S, Janssen M, Ubacht J (2017) Blockchain in government: benefits and implications of distributed ledger technology for information sharing. Article in government information quarterly
2. Dr.Sanjaya Baru, secretory general FICCI., Neel Ratan, Regional managing partner and government leader, PwC., "Blockchain:The next innovation to make our cities smarter"
3. https://en.wikipedia.org/wiki/Blockchain
4. Banarjee A (2018) Integrating blockchain with ERP. Infosys Whitepaper
5. Huckle S, Bhattacharya R, White M, Beloff N (2016) Internet of things, blockchain and shared economy applications. The 7th international conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2016), Procedia Computer Science, vol 98, pp 461–466. https://doi.org/10.1016/j.procs.2016.09.074
6. https://en.wikipedia.org/wiki/Birth_certificate
7. Sun J, Yan J, Kem Z, Zhang K (2016) Blockchain-based sharing services: what blockchain technology can contribute to smart cities. Financ Innov 2:26
8. Nofer M, Gomber P, Hinz O, Schiereck D (2017) Blockchain
9. Kamanashis Biswas School of Information & Communication Technology Griffith University, Gold Coast, Australia, Vallipuram Muthukku marasamy Institute for Integrated and Intelligent Systems Griffith University, Gold Coast, Australia (2016) IEEE 18th international conference on high performance computing and communications; IEEE 14th international conference on smart city; IEEE 2nd international conference on data science and systems. Securing smart cities using blockchain technology
10. Zheng Z, Xie S, Dai H-N, Chen X, Sun Y-S, Wang H (2018) Blockchain challenges and opportunities: a survey. Int J Web Grid Serv 14(4)
11. https://www.smartcity.press/blockchain-implementations-in-smart-cities/
12. https://image.slidesharecdn.com/electronicdeliveryofservicesinitiative-110612071037-phpapp01/95/electronic-delivery-of-services-initiative-10–728.jpg?cb=1307862670
13. Yli-Huumo J, Ko D, Choi S, Park S, Smolander K. Where is current researchon blockchain

GeoSharding—A Machine Learning-Based Sharding Protocol

Hardik Ruparel, Shreyashree Chiplunkar, Shalin Shah, Madhav Goradia, and Mahesh Shirole

Abstract Sharding is one of the most prominent concepts which involves the division of the network into shards for concurrent processing of transactions. Different sharding protocols are being implemented in blockchains to enhance its scalability. The existing blockchain systems create shards using proof-of-work consensus protocol. This research aims at developing a machine learning-based sharding process that uses the nodes' geographical locations—latitudes and longitudes. IP addresses of the nodes are mapped to geographical coordinates, and these coordinates are then divided into shards using a suitable clustering algorithm. The nodes in the shards are geographically closer, thereby reducing the propagation delay in the network during intra-shard communication. GeoSharding has been tested to be significantly faster as compared to PoW-based sharding. This optimizes the network sharding process, thus escalating the scalability to a new level.

Keywords Blockchain · Sharding · Clustering · Proof of work · Machine learning

H. Ruparel (✉) · S. Chiplunkar · S. Shah · M. Goradia · M. Shirole
CE & IT Department, Veermata Jijabai Technological Institute, Mumbai, India
e-mail: hruparel_b15@it.vjti.ac.in

S. Chiplunkar
e-mail: suchiplunkar_b15@it.vjti.ac.in

S. Shah
e-mail: spshah_b15@it.vjti.ac.in

M. Goradia
e-mail: gmbipin_b15@it.vjti.ac.in

M. Shirole
e-mail: mrshirole@it.vjti.ac.in

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_10

1 Introduction

Blockchain is an immutable and a shared ledger that enables the process of recording values known as transactions and tracking assets in a business network. Virtually anything of value can be recorded, tracked and traded on a blockchain network, reducing risk and cutting costs for all involved [1]. Being decentralized in nature, a blockchain is highly secure as no single user can alter or remove an entry in the blockchain since it would require changing all the blocks, which is realistically impossible.

However, one of the most pivotal problems in the current blockchain architecture is scalability. Today's representative blockchain such as bitcoin [2] takes 10 min or longer to confirm transactions and achieves a throughput of 7 Tx/s. In comparison, the average throughput of the current centralized systems is around 2000 and 56,000 Tx/s during holidays. For a wide-scale adoption of blockchain, this issue needs to be addressed as soon as possible [3].

Effective throughput in the overlay network is defined as the percentage of nodes that receive the propagated blocks within an average block interval period. 10% of the nodes in the network would be unable to keep up if the transaction rate exceeds the 90% effective throughput, potentially resulting in a denial of service to users and reducing the network's effective mining power [3].

$$X\% \text{ effective throughput} = \frac{\text{block size}}{X\% \text{ propagation delay}} \tag{1}$$

The formula in Eq. 1 is crucial for understanding why reparameterization like increasing the block size and decreasing the latency provides limited benefits. The bitcoin community has put forth various proposals to modify the parameters like block size and block interval that could make the system scalable. Two guidelines should be followed to ensure that at least 90% of the nodes in the network have sufficient throughput:

- **Throughput limit**. Given today's 10 min average block interval, the block size should not exceed 4 MB. Maximum throughput of at most 27 Tx/s is obtained at 4 MB block size.
- **Latency limit**. If full utilization of the network's bandwidth is to be achieved, the block interval should not be smaller than 12 s [3].

Hence, a completely new and scalable architecture is needed to be developed to increase the transaction rate of blockchains. As per our understanding, one can achieve scalability in five planes like network, consensus, storage, view and side plane. In our approach, we aim at achieving scalability in the network plane using geographical sharding. This paper is organized as follows: Sect. 2 discusses the basic concepts used in the paper. Section 3 discusses the proposed system. The results of the experiment and comparisons with the existing systems are presented in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Basic Concepts

2.1 Sharding

Sharding is the division of the network into independent groups of nodes called shards for concurrent processing of transactions. For sharding blockchain systems, the nodes will only maintain a portion of the data and not the entire information [4]. However, each node does not load the information on the entire blockchain, thus helping in scalability. In blockchain, sharding can be done at different levels like network level, transaction level and computational level.

- **Network sharding**. Network sharding is a technique that allows the network to be segregated into smaller groups of nodes called *shards*.
- **Transaction sharding.** Whenever a transaction reaches the network, it is assigned to a specific shard. The assignment of a transaction to a particular shard is based on a few bits of the sending address of the transaction. This prevents the double-spending problem as transactions sent by a user are assigned to a particular shard only.
- **Computation sharding**. The sharding of computational resources in the blockchain network via an overlay above the consensus process is called *computational sharding*. Computations can be performed efficiently with some shards acting as mappers and the rest as reducers for a map-reduce task.

2.2 Clustering

In machine learning, clustering is the process of combining objects based on their attributes and aggregating them according to their similarities. There are many clustering techniques like *distance-based, density-based, interval or statistical-based*. In our research, we use distance as the similarity measure for creating clusters. In the case of blockchain, network sharding is done for nodes, where nodes in a cluster are closer to each other than the nodes in another cluster.

Various distance measures like *Euclidean, Squared Euclidean, Manhattan, Correlation, Haversine,*, etc., can be used for measuring the distance between data points. However, we require a formula that measures the distance between locations on the Earth's surface because Earth is a globe. So, we use Haversine distance, which is suitable for spherical shapes.

Haversine Distance. The Haversine formula helps in calculating the shortest distance between any two points on a spherical surface using their latitudes and longitudes measured along the surface. This formula is given in Eq. 2, and it is important in navigation [5].

Table 1 Comparison between k-means and DBSCAN

Parameters	**K-means**	**DBSCAN**
Number of clusters	*k*	Depends on the density of the data points
Size of clusters	Similar-sized	Differs across clusters
Shape of clusters	Prefers spherical clusters	Handles skewed or randomly shaped clusters

$$d = 2r \arcsin \sqrt{\sin^2\left(\frac{\phi 2 - \phi 1}{2}\right) + \cos(\phi 1)\cos(\phi 2)\sin^2\left(\frac{\lambda 2 - \lambda 1}{2}\right)} \tag{2}$$

where d—the distance between the two points (along a great circle of the sphere), r—the radius of the sphere, $\phi 1$, $\phi 2$—the latitude of points 1 and 2, respectively, and $\lambda 1$, $\lambda 2$—the longitude of points 1 and 2, respectively.

2.3 Comparison of Clustering Algorithms

An appropriate clustering algorithm needs to be chosen for efficient formation of clusters. Since clustering algorithm is used for network sharding, the parameters such as distance function and number of clusters are chosen accordingly. Different clustering algorithms that can be used are *k-means, mean-shift, DBSCAN, expectation–maximization (EM), agglomerative, hierarchical,* etc. The two clustering algorithms suitable to our motive of clustering GPS points are:

- **K-means**. K-means clustering [6] classifies a given dataset into *k* number of clusters, where *k* is specified in advance. The clusters are then represented as points, and all the data points are associated with the nearest clusters. These clusters are then recomputed and adjusted until a desired result is reached.
- **DBSCAN**. DBSCAN [6] is another approach that makes an assumption that a cluster is a connected region with relatively dense points. DBSCAN algorithm requires two parameters: the maximum distance (*eps*) and the minimum number of points (*minPts*) in the given region which are required to form a cluster (Table 1).

Since the benefits offered by DBSCAN clustering algorithm are not relevant to our results and also the number of clusters cannot be specified beforehand, we choose k-means as the algorithm for clustering GPS points.

2.4 Similarity Measure in Leader Election

Various measures can be used for calculating the similarity between two strings, out of which Hamming and Levenshtein distance are most suitable in this context:

- **Hamming distance**. The number of places at which the corresponding characters in two equal length strings are different
- **Levenshtein distance**. The minimum number of single-character edits (insertions, deletions or substitutions) required to change one string into the other.

Hamming distance calculates similarity much faster than Levenshtein distance, but it has some limitations. This can be explained using an example: Consider the strings "12345", "51234" and "67890". The Hamming distance between the first and the second string is 5 and that between the first and the third string is also 5. On the other hand, the Levenshtein distance between the first and the second string is 2 and that between the first and the third string is 5. It can be seen that the first two strings are similar, but the Hamming distance fails to determine this. Also, comparing the similarity values obtained from both the distance measures, we have found that Hamming gives a very narrow range of values, whereas Levenshtein gives a far wider range of values. Considering Hamming, similarity measure will no longer be a distinguishing factor in leader election. Hence, we choose Levenshtein distance.

3 Proposed System

Our proposed system aims at achieving scalability and improving efficiency in the network plane of the blockchain architecture. There are existing systems that implement sharding for achieving scalability. However, these systems are not very efficient due to various factors. Some of the factors are: (a) time taken to create a shard and thus higher downtime of the network in the event of shard recreation, (b) propagation delay in intra-shard communication and (c) higher computational resources required in the existing systems. Thus, we aim to improve the efficiency by introducing the concept of geographical sharding for the network plane.

3.1 Parameters Considered for Analysis

The following parameters are used for the analysis of our proposed system:

- **Propagation delay**. In computer networks, propagation delay is the amount of time it takes for the head of the signal to travel from the sender to the receiver. It can be computed as the ratio between the link length and the propagation speed over the specific medium [7].

- **Computational resources**. A computational resource is a resource used by some computational models in the solution of computational problems [8]. PoW is a task, which requires huge computational resources.
- **Shard creation time**. Shard creation time is defined as the time taken for dividing the network into shards. Our research aims at reducing shard creation time.
- **Delay in joining of new nodes**. New nodes cannot join the network at any time. They have to wait for the shard reformation epoch and perform PoW. If they could not compute PoW within a specific time, they have to wait for the next epoch. Our research aims at including all such nodes during the next epoch itself and not delay it further.
- **Network downtime**. Downtime refers to a period of time during which a computer system, a server or a network is shut off or unavailable for use. There are various reasons for downtime, and in blockchain, one of these reasons is the shard reformation phase.

3.2 Geographical Mapping

The first step in implementing GeoSharding is mapping the Internet protocol (IP) addresses of the nodes to locations on the Earth's surface. Various APIs are available that allow us to do IP lookup and return the details in XML format. The details of all the IP addresses are stored by regional Internet registries. In the response file, we obtain the latitudes and longitudes which will be used in the Haversine formula to calculate the distance between two nodes.

3.3 Leader Election

To increase the throughput of the blockchain system, a leader is elected from each shard at every epoch [9]. Every node within a shard during a leader election epoch calculates the target value, which is the hash of its shard name and the current block number. Every node then calculates the hash of its address which includes the IP address and its port number. The similarity between these two strings of hashes forms a factor in deciding the leader of the shard [10]. The similarity between these two strings of hashes is calculated using Levenshtein distance. The more similar the hash of the node's address and the hash of the shard name and block number, the more are the chances of that node becoming the leader.

Once the distance is calculated, it is broadcasted to every node within the shard [11]. The distances received are also verified by the receiving nodes. All nodes have an array list of ages depicting the time since every node has become a leader. The node which has become a leader recently has a lower age value. Every node then calculates the cumulative scores based on the ages and the distances received from other nodes. To give a fair chance to the nodes which are not elected as a leader for a long time, the age can be given a higher weight than the distance, like 60% weightage to the age factor and 40% weightage to the distance factor. This cumulative score is called as the leader competence score. The node which has the maximum score becomes the leader for that epoch. The address of the elected leader is then broadcasted to every other node within the shard. The age of the leader node is decreased, and the ages of other nodes within the shard are increased. After network sharding, every node has the shard information in the form of a structure. The structure of the shard is:

```
Struct Shard contains
        shardName string        //name of the shard in string format
        shardlist[ ] Address     //addresses of all the nodes within the shard
        agelist[ ] float64        //ages of all the nodes within the shard
end
```

Algorithm 1 Leader Election Algorithm

```
function ELECT(shard, currentBlockNumber, myAddress)
    Input: shard - a structure that every node receives during network sharding. currentBlockNumber- an integer stating the current block number in the blockchain. myAddress - the address of the node executing the program
    Output: Leader of the shard
$A \leftarrow$ Hash(shard.shardName, currentBlockNumber)
$B \leftarrow$ Hash(myAddress)
distance $\leftarrow$ Levenshtein distance (A, B)
Broadcast distance to the nodes in shard.shardlist
Verify the distances received from other nodes
Compute the competence score of each node to become the leader using distance and age of the node from the shard.agelist
Broadcast the leader
if leader = myAddress then
        Broadcast "I am the leader"
Decrement the age of the leader node and increment the ages of all other nodes
```

4 Results

The result analysis is completely based on the assumption that proof of work for establishing mining identities takes approximately 10 min. The proposed sharding technique is applied to the GeoLite2 database which contains over 3.3 million entries. The objective function which is used to optimize the clusters is composed of the Haversine distance function. The Ethereum and bitcoin network have around 25,000 and 7000 reachable nodes, respectively. So, to analyze the results of sharding, we consider a larger network size, and the database is sampled randomly to choose 33,645 records for clustering. The results written in the following section are obtained by the implementation of k-means [6, 12] in GoLang [13], and the clustering algorithm

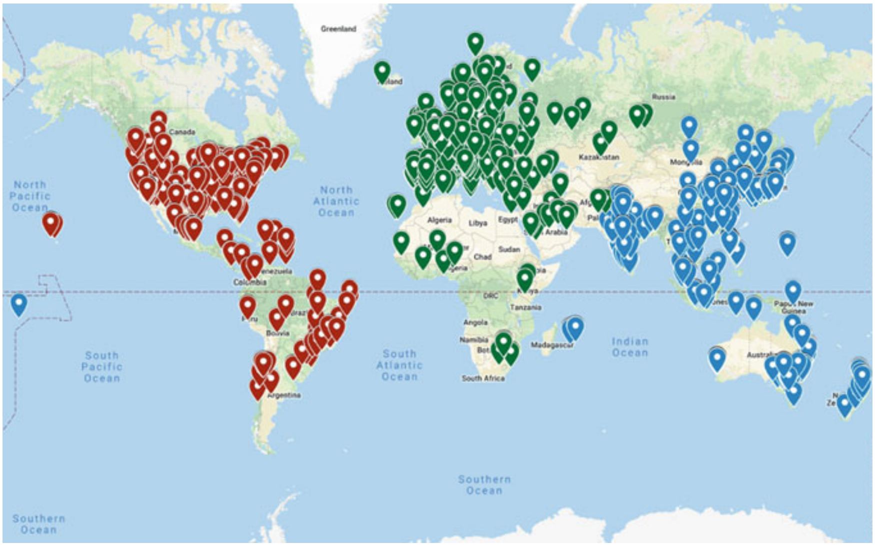


Fig. 1 Geographical sharding produced using k-means clustering algorithm for $k = 3$

has been executed 10 times to reduce the variations in the result. This entire process of training and forming seven clusters took an average of 0.84434 s.

Figure 1 shows a snapshot of the proposed method for creating shards. These shards are created using the k-means clustering algorithm, where the value of k=3. The nodes in each shard are geographically closer than the nodes in other shards.

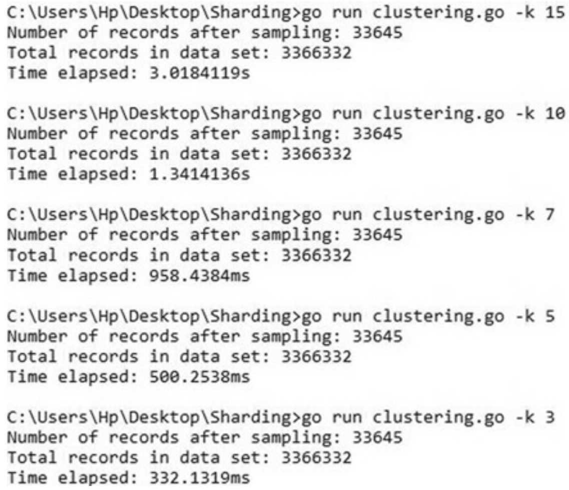
4.1 Dataset Description

GeoLite2 City database provided by the MaxMind Developers [14] was used for the implementation of the concept. This dataset contains a total of 3.366332 million records and 10 columns viz:
network (IP in string), *geoname_id* (integer), *registered_country_geoname_id* (integer), *represented_country_geoname_id* (integer), *is_anonymous_proxy* (bo-olean), *is_satellite_provider* (boolean), *postal_code* (string), *latitude* (decimal), *longitude* (decimal), *accuracy_radius* (integer). The attributes used in formation of clusters are *network*, *latitude* and *longitude*.

Preprocessing and Sampling. The database is first preprocessed to handle missing values. The dataset contains some records with missing latitude and longitude values. Since the number of such records is only 19, these records are ignored. 33,645 records were chosen randomly without replacement from the remaining records and then supplied as an input to the clustering algorithm.

Table 2 Time taken for running clustering algorithm on 33,645 records

#Clusters	Total time (s)	Average time per clustering (s)
3	2.5391	0.2539
5	4.0836	0.40836
7	8.4434	0.84434
10	13.9184	1.39184
15	26.8803	2.68803



```
C:\Users\Hp\Desktop\Sharding>go run clustering.go -k 15
Number of records after sampling: 33645
Total records in data set: 3366332
Time elapsed: 3.0184119s

C:\Users\Hp\Desktop\Sharding>go run clustering.go -k 10
Number of records after sampling: 33645
Total records in data set: 3366332
Time elapsed: 1.3414136s

C:\Users\Hp\Desktop\Sharding>go run clustering.go -k 7
Number of records after sampling: 33645
Total records in data set: 3366332
Time elapsed: 958.4384ms

C:\Users\Hp\Desktop\Sharding>go run clustering.go -k 5
Number of records after sampling: 33645
Total records in data set: 3366332
Time elapsed: 500.2538ms

C:\Users\Hp\Desktop\Sharding>go run clustering.go -k 3
Number of records after sampling: 33645
Total records in data set: 3366332
Time elapsed: 332.1319ms
```

Fig. 2 Output snippet for geographical sharding

4.2 *Speed Analysis*

The speed of the proposed system is measured by the time taken to form the clusters. Figure 2 shows the time taken for clustering when the number of clusters specified is 15, 10, 7, 5 and 3. This program was run 10 times, and the average time taken for clustering using specified number of clusters is shown below in Table 2. The table shows that the average time taken for creating 15 shards in a network of 33,645 nodes is 2.68803 s, which is considerably faster than the general PoW-based method since the method requires every node to solve PoW and listen to PoW results of other slower nodes for at least 600 s.

4.3 *Scalability of the Proposed System*

A simple and most efficient way to check the scalability is by increasing the sample size of the dataset, which is fed to the clustering algorithm. The results in Table 3 show the time taken to form 10 shards for different sizes of the dataset.

Table 3 Time taken for running clustering algorithm on different sizes of dataset

#Nodes (Sample size)	Time taken for shard creation (s)
3367	0.0746991
33,645	1.6607684
3,36,487	9.3331801
33,64,864	226.1304

Considering the current Ethereum network size, even if we increase its size 13 times to 0.336 million nodes, the time it takes for creating 10 shards is as little as 9.3331801 s, which is at most 1.5% of the time taken for PoW-based techniques [15]. For the network size of 3.3 million nodes, it takes 226.1304 s to form 10 shards, which still saves at least 70% of the time utilized for shards formation in PoW-based techniques. Thus, even if a network of such a huge size is reached, the time taken by GeoSharding to form shards will be at most 30% of the time taken for PoW-based techniques, which is a significant advantage in making the system faster and scalable.

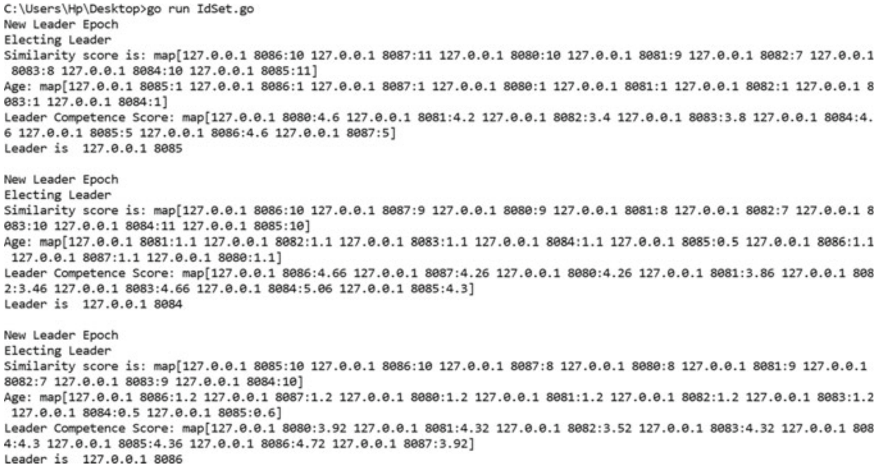
4.4 Efficiency and Security

The quality of clusters determines the efficiency of the clustering algorithm. Figure 1 shows that the clustering algorithm has created discrete clusters and divided the nodes into shards efficiently. This map is developed with the help of Google My Maps [16] which enables you to visualize the points on the world map. Each point on the map is an IP address of a node at that location. A csv file containing longitude, latitude and a cluster id of the points is given as an input.

All the nodes run the clustering algorithm, and the shard that each node belongs to is decided in consensus by all the nodes. Since it is considered that at least 2/3rd of the nodes in network are honest, a node will always be assigned to the correct shard, thereby making the system secure. This process of sharding is repeated every 20 hours; hence, this sharding approach will significantly decrease the downtime of the network. Every new node that joins the network has to wait for the shard reformation epoch.

4.5 Leader Election

The leader election algorithm needs to be run once every leader epoch. This algorithm is proposed in Sect. 3.3. Below is an output snippet of leader election run for three epochs. A leader is chosen such that every node is given a fair chance. Figure 3 shows the addresses of the nodes in the shard and the similarity distances computed



```
C:\Users\Hp\Desktop>go run IdSet.go
New Leader Epoch
Electing Leader
Similarity score is: map[127.0.0.1 8086:10 127.0.0.1 8087:11 127.0.0.1 8080:10 127.0.0.1 8081:9 127.0.0.1 8082:7 127.0.0.1
 8083:8 127.0.0.1 8084:10 127.0.0.1 8085:11]
Age: map[127.0.0.1 8085:1 127.0.0.1 8086:1 127.0.0.1 8087:1 127.0.0.1 8080:1 127.0.0.1 8081:1 127.0.0.1 8082:1 127.0.0.1 8
083:1 127.0.0.1 8084:1]
Leader Competence Score: map[127.0.0.1 8080:4.6 127.0.0.1 8081:4.2 127.0.0.1 8082:3.4 127.0.0.1 8083:3.8 127.0.0.1 8084:4.
6 127.0.0.1 8085:5 127.0.0.1 8086:4.6 127.0.0.1 8087:5]
Leader is  127.0.0.1 8085

New Leader Epoch
Electing Leader
Similarity score is: map[127.0.0.1 8086:10 127.0.0.1 8087:9 127.0.0.1 8080:9 127.0.0.1 8081:8 127.0.0.1 8082:7 127.0.0.1 8
083:10 127.0.0.1 8084:11 127.0.0.1 8085:10]
Age: map[127.0.0.1 8081:1.1 127.0.0.1 8082:1.1 127.0.0.1 8083:1.1 127.0.0.1 8084:1.1 127.0.0.1 8085:0.5 127.0.0.1 8086:1.1
 127.0.0.1 8087:1.1 127.0.0.1 8080:1.1]
Leader Competence Score: map[127.0.0.1 8086:4.66 127.0.0.1 8087:4.26 127.0.0.1 8080:4.26 127.0.0.1 8081:3.86 127.0.0.1 808
2:3.46 127.0.0.1 8083:4.66 127.0.0.1 8084:5.06 127.0.0.1 8085:4.3]
Leader is  127.0.0.1 8084

New Leader Epoch
Electing Leader
Similarity score is: map[127.0.0.1 8085:10 127.0.0.1 8086:10 127.0.0.1 8087:8 127.0.0.1 8080:8 127.0.0.1 8081:9 127.0.0.1
8082:7 127.0.0.1 8083:9 127.0.0.1 8084:10]
Age: map[127.0.0.1 8086:1.2 127.0.0.1 8087:1.2 127.0.0.1 8080:1.2 127.0.0.1 8081:1.2 127.0.0.1 8082:1.2 127.0.0.1 8083:1.2
 127.0.0.1 8084:0.5 127.0.0.1 8085:0.6]
Leader Competence Score: map[127.0.0.1 8080:3.92 127.0.0.1 8081:4.32 127.0.0.1 8082:3.52 127.0.0.1 8083:4.32 127.0.0.1 808
4:4.3 127.0.0.1 8085:4.36 127.0.0.1 8086:4.72 127.0.0.1 8087:3.92]
Leader is  127.0.0.1 8086
```

Fig. 3 Output snippet of leader election for three leader epochs

by them. This distance is the Levenshtein distance which is explained in Sect. 3.3. Also, the corresponding current ages of the nodes are shown. Leader competence score is calculated for every node using the ages and the similarity distances as shown in Eq. 3:

$$Leader\_competence\_score = 0.4 \times similarity\_distance + 0.6 \times age \qquad (3)$$

The factors 0.4 and 0.6 assure that every node is given a fair chance to become a leader. The node with the highest competence score is elected as the leader. Further, the age of the elected leader is decremented to a value of 0.5, and the ages of other nodes are incremented by 0.1 so that other nodes have a greater chance to become the leader in the next epoch.

4.6 *Comparison with the Existing Solutions*

Elastico [15] uniformly partitions the mining network (securely) into smaller committees (shards), each of which processes a disjoint set of transactions. Communication in Elastico takes place between the nodes in the shards and the directory committee. Elastico was the first one to propose the idea of network and transaction sharding. However, the problem of scalability cannot be solved alone by these sharding techniques as the intra-shard communication is still cumbersome since the nodes in one shard might be located at different locations, thus leading to propagation delay. Our proposed system solves this problem by using clustering

Table 4 Comparison of performance parameters with related systems [15, 17, 18]

Parameters	Proposed system	Elastico [15]	Zilliqa[17, 18]
Propagation delay	Low	High	High
Consumption of computational resources	Low	High	High
Shard creation time	Very low	High	High
Network downtime	Very low	Moderate	Moderate
Joining of new nodes	At shard reformation epoch	At shard reformation epoch if PoW is solved or else at next epoch	At Shard Reformation Epoch if PoW is solved or else at next epoch

algorithms to geographically cluster nodes into shards and process transactions in parallel, thus reducing the propagation delay.

Zilliqa [17, 18] uses proof of work to form shards. The nodes are sorted according to the nonce values obtained from the PoW results, shards are created and the results are communicated. This is a very time-consuming process causing a greater downtime of the network during shard formation epoch. Our system uses k-means for sharding which results in faster creation of shards.

Table 4 shows the comparison of our proposed system with the existing systems. The proposed system is analyzed to provide the following benefits as compared to the existing scalable architectures:

Propagation Delay. In the existing blockchain systems with sharding, all nodes perform proof of work and are divided into shards based on the PoW result. This results in shards that consist of randomly located nodes.

In GeoSharding, the nodes in a particular shard are closer to each other than the nodes in other shards, as shown in Fig. 1. Hence, the propagation delay during intra-shard communication is less in GeoSharding as compared to the existing sharded systems.

Computational Resources. Proof of work is a task which requires huge computational resources. However, GeoSharding has very moderate system requirements, and the clustering program can be run by any node. This would result in an increase in the mining power and hence the efficiency of the system.

Shard Creation Time. Proof of work is a time-consuming process. However, GeoSharding creates shards in a very short time since it uses a clustering algorithm like k-means, and hence, the shard setup phase is optimized. Shard reformation occurs every 20 h.

Joining of New Nodes. All nodes willing to join the network are assigned to specific shards based on the clustering algorithm, whereas in the existing systems, if a new node fails to compute PoW in a specific time, it is denied to join the network in that epoch.

Reduction in Downtime of the Network. The downtime of the proposed system is significantly less in comparison with other systems since the shard creation phase is optimized to obtain better results.

5 Conclusion

GeoSharding is a scalable and efficient algorithm for clustering nodes based on the latitude and the longitude values. Proof-of-work-based sharding technique first requires a considerable amount of time to solve PoW for establishing mining identities. It requires additional time to broadcast the network sharding result of all the nodes and reaches a consensus on these shard lists. This paper proposed sharding the network on the basis of k-means algorithm which is fast, scalable and accurate in creating shards. GeoSharding accelerates this process of sharding the network, considerably faster than PoW-based sharding. Also, the number of messages communicated with each peer node is 2/3rd of the messages communicated in PoW-based sharding. For every shard reformation epoch, the time taken to form shards in the worst case will still be better than the time taken for PoW-based network sharding, thereby improving the overall throughput of the architecture.

References

1. Gupta M.: Blockchain for Dummies. 2nd IBM Limited edn. John Wiley & Sons, Inc, Hoboken, NJ (2018)
2. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System
3. Croman, K., Decker, C., Eyal, I., Gencer, A., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E., Song, D., Wattenhofer, R.: On Scaling Decentralized Blockchains (A Position Paper)
4. Sharding in Blockchain, https://medium.com/edchain/what-is-sharding-in-blockchain-8afd9ed4cff0
5. Haversine Formula, https://en.wikipedia.org/wiki/Haversine_formula
6. Ganeshan D.: GPS Clustering and Analytics, http://web.cs.wpi.edu/~emmanuel/courses/cs528/F17/slides/papers/deepak_ganesan_GPS_clustering.pdf
7. Propagation Delay, https://en.wikipedia.org/wiki/Propagation_delay
8. Computational Resources, https://en.wikipedia.org/wiki/Computational_resource
9. Eyal, I., Gencer, A., Sirer, E., Renesse, R.: Bitcoin-NG: A Scalable Blockchain Protocol. In: Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation, Santa Clara, CA, USA (2016)
10. Obeidat, A., Gubarev, V.: Leader Election in Peer-to-Peer Systems
11. Perlin/noise-Library, https://github.com/perlin-network/noise
12. K-means implementation in Go, https://github.com/MathieuMailhos/gomeans
13. The Go Programming Language, https://tour.golang.org
14. GeoIP2 City Dataset,https://dev.maxmind.com/geoip/geoip2/geoip2-city-country-csv-databases/
15. Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., Saxena, P.: A Secure Sharding Protocol For Open Blockchains

16. Google MyMaps, https://www.google.com/mymaps
17. The Zilliqa Team: The Zilliqa Project: A Secure, Scalable Blockchain Platform. Version 1.0 (May 2018)
18. The Zilliqa Team (Aug. 2017) The ZILLIQA Technical Whitepaper. Version 1

MapReduce-Based Framework For Blockchain Scalability

Maneesh Darisi, Om Modi, Vasu Mistry, and Dhiren Patel

Abstract The proliferation of blockchain technology into the wide spectrum of industries has been stymied by its inability to scale. With the augmenting popularity of blockchains, scalability of blockchain is hindering it from attaining the meteoric transaction rate present in the existing solutions like MasterCard and VISA. Presently, the scalability solutions of blockchains use several off-chain and on-chain mechanisms. This paper proposes an on-chain solution backed by big data technologies. We aim to provide a real-time scalable transaction processing by using big data framework to overcome the roadblocks of scalability. Our framework has horizontal and vertical scaling to augment the sluggish blockchain transaction rate using *sharding* frameworks of big data.

Keywords Blockchain scalability · Big data · Sharding · MapReduce

1 Introduction

Blockchain is a sequential agglomeration of an immutable data structure called blocks that includes the set of valid transactions which is transparent to every member of the chain and has entities like miners who perform consensus and deploy blocks on the chain facilitating the security from single point of failure.

M. Darisi · O. Modi (✉) · D. Patel
Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Mumbai 400019, India
e-mail: om.vjti@gmail.com

V. Mistry
Independent Researcher, Mumbai, India

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_11

> A blockchain is a tuple (G, B) where G is a genesis state and $B = [\beta 1, \beta 2, \beta 3\ldots]$ is an ordered list of blocks. A blockchain is valid if every $\beta \in B$ is valid, and so $G + \beta 0 + \beta 1 + \cdots = \sigma f$ is a valid state. A block β is a package containing a list of transactions T, a reference to a parent block and auxiliary verification data. A block is valid in the context of a state σ if:
>
> • The block's transaction list is valid in the context of σ
>
> • Some other conditions, generally determined by the consensus algorithm (e.g. proof of work), are met. Buterin [1]

It has been a decade ever since blockchain has come into this technology world but has still failed to throw light on some burning issues like scalability, interoperability, governance, etc. Scalability in the context of blockchains is dependent on security and capacity limitation of public blockchains protocol which demands:

- Every transaction on the blockchain must be processed by every single node on the blockchain.
- Every operation like payment and deployment of a smart contract must be replicated by all the full nodes.

This makes the public blockchain autonomous and reliable and eliminates the need for dependence on any counterparty. But the above protocol depreciates the throughput of the blockchain drastically. The proof–of-work (PoW) [2] protocol puts the constraint onto the blockchain that the blockchain throughput is equal to the processing capability of the individual nodes only. We require blockchains to scale to the global user database and at the same time ensure as much decentralization as possible.

Presently, the scalability solutions of blockchain are divided into two main categories:

1. On-chain solutions: modifying the underlying blockchain infrastructure.
2. Off-chain solutions: develop additional infrastructure that connects to the blockchain.

Ethereum [3] projects relating to on-chain solutions focus on the changing of the present consensus algorithm from proof of work [2] and to proof of stake [4] (PoS) and implement sharding on to the blockchain infrastructure. Ethereum projects relating to off-chain solutions which focus on implementing state channels, lightning network called Raiden [5], a new concept layer to scale smart contracts called plasma [6], and Truebit [7] to process complex computations off the chain. Scalability of blockchain systems is heavily affected by three factors—decentralization, scalability and security.

Due to its operating nature, colossal amount of data is generated, and to deal with it, big data provides the tools which are highly scalable and powerful for executing distributed parallel computations on the data.

MapReduce [8] is a stalwart and robust framework that provides a distributed and parallel environment for scalable and fault-tolerant computations of batch jobs. In the mapper phase, the data is divided into shards and has a <key, value> format which is given to reducer for its aggregation and giving the overall output of the job. Scalability is ensured as the mappers and reducers execute across the cluster parallelly in a distributed manner. The concurrency and task allocations are performed intrinsically and securely by MapReduce.

Hadoop [9] provides inherently the benefits which are required for the blockchain to scale. Hadoop works best in a cluster by using low-cost commodity hardware and is buttressed by the parallelism provided by MapReduce. Hadoop at its core is majorly made up of HDFS and MapReduce. HDFS filesystem provides fault tolerance and security which is required while computation of transactions taking place in the Hadoop ecosystem. HDFS by itself supports several cryptomechanisms to secure data using encryptions and authentication frameworks. Thus, by utilizing low-cost commodity hardware and being able to run these scalable Hadoop ecosystems, decentralization can be achieved, while scalability of transactions is possible by using the MapReduce framework. Knox [10] and Ranger [11] intrinsically provide security when parallel computations take place in the Hadoop environment. We aim to break the trilemma by using ecosystem to counter the scalability problem of the blockchain.

We aim to integrate big data technologies on to the blockchain architecture to improve the present blockchain transaction rate. We propose the architectural and technological changes required in the network and consensus layer to scale the present blockchain architecture.

Rest of the paper is organized as follows: Sect. 2 discusses related work in the field of blockchain scalability, and Sect. 3 describes the proposed solution with conclusion and references at the end.

2 Related Work

Practically, it has been proven that proof of work is sluggish time and resource consuming. Therefore, a new wave of consensus algorithms aims to overthrow proof of work. Delegated proof of stake, Byzantine fault-tolerant [12] variants, etc., aim to obtain only the consensus of only a representative group of entities rather than the entire network. These systems abate the decentralization aspect of blockchain in order to increase scalability.

Polkadot [13] and Cosmos [14] aim to build a network infrastructure which to resolves interoperability issues among heterogeneous blockchains. Polkadot has developed a relay chain and several parachains for provisioning interoperability and scalability, but Polkadot is not envisaged to support the deployment of customized complex blockchains smart contracts. Cosmos is an Internet of blockchains scalability network that achieves scalability by using high performance, secure tendermint [15] consensus core which provides a transaction rate up to 10000 tx per second. Cosmos implements a intrinsic star topology to connect heterogeneous independent blockchains(zones) using a central hub. Polkadot and Cosmos are not envisaged to support the deployment of customized complex blockchains smart contracts.

Zilliqa [16] and Merklix tree [17] envision to implement sharding onto the existing blockchain platforms. The Merklix tree aims to increase the torpid transaction of bitcoin using sharding. Zilliqa and Merklix tree compromise on security aspect of the blockchain.

Cardano [18] is an innovative project which not only addresses scalability and interoperability issues of blockchain but also aspires to improve the governance of the existing blockchain system.

3 Proposed Solution

Our solution solves one of the most difficult conundrums hindering the proliferation of blockchain technology in the industry by using big data. We address the problem of scalability while not compromising on decentralization using the existing Hadoop scalable tools. Our solution implements transaction and network sharding to ameliorate the torpid blockchain rate.

Since we are using *MapReduce,* all the inputs to *MapReduce* framework should be in a <key, value> pair format.

According to Fig. 1, the *MapReduce* workflow consists of the:

- Mapper phase
- Reducer phase
- Final Consensus phase.

There are two input pools to our framework which are

1. Transaction pool
2. Miner pool.

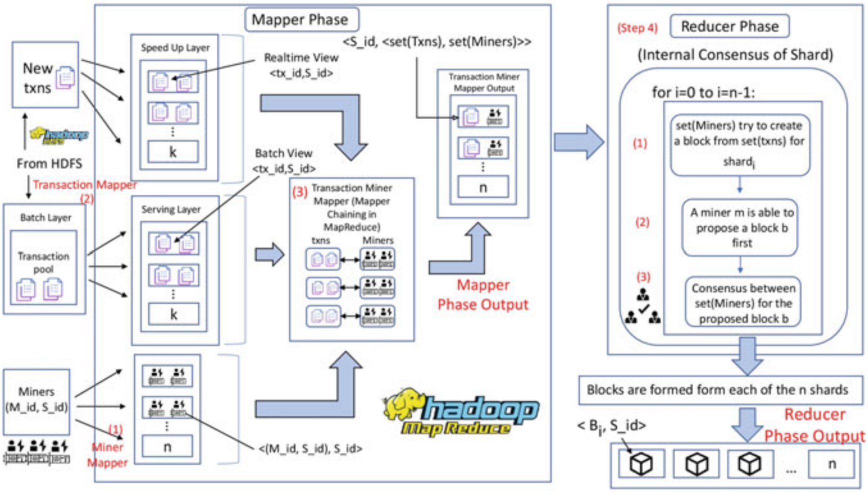

Mapper Phase
<S_id, <set(Txns), set(Miners)>>
New txns
From HDFS
Transaction Mapper (2)
Batch Layer
Transaction pool
Speed Up Layer
k
Realtime View <tx_id,S_id>
Batch View <tx_id,S_id>
Serving Layer
k
(3) Transaction Miner Mapper (Mapper Chaining in MapReduce)
txns
Miners
Transaction Miner Mapper Output
n
Mapper Phase Output
Miners (M_id, S_id)
(1) Miner Mapper
n
<(M_id, S_id), S_id>
hadoop Map Reduce
(Step 4)
Reducer Phase
(Internal Consensus of Shard)
for i=0 to i=n-1:
(1)
set(Miners) try to create a block from set(txns) for shard$_i$
(2)
A miner m is able to propose a block b first
(3)
Consensus between set(Miners) for the proposed block b
Blocks are formed form each of the n shards
< B$_i$, S_id>
Reducer Phase Output
...
n

Fig. 1 Detailed implementation workflow architecture of the mapper and the reducer phase

Step 1: To bifurcate the miner pool into ‘n’ shards (according to Eq. 1) using random sampling rather than applying geographic sharding. The major disadvantage of geographic sharding is the fact that the miners bifurcated based on their geographic locations reducing the decentralization of blockchain systems. If the shards are created based on location, then the mining pool of a certain location can have the authority over the internal consensus of that shard. It is better to randomly sample the miners based on their quantity into n shard, thus avoiding the concentration of authority and increasing the security of the system. But this leads to the issue network propagation delay among the miners in the shard. All the new miners added to the mining pool are taken to pre-computed shards after each epoch.
Step 2: The transaction pool is divided into ‘k’ shards (according to Eq. 1). Two types of sharding can be applied on the transaction pool 1. state-based sharding and 2. transaction-based sharding. Our framework is presently using transaction-based sharding and the lambda architecture to dynamically update the transaction pools [19, 20]. The batch layer creates batch views of the transactions that have entered the transaction pool and sends it to the service layer for further processing. Speedup layer produces real-time views and sends them to the transaction-miner mapper directly along with the batch views, thus reducing latency time. All the batch views are updated after each epoch.
Step 3: The transaction-miner mapper maps the k%n transaction shards to the n miner shards.
Step 4: In the reducer phase, the miners in each miner shard must come to consensus upon the k%n transaction shards and have to propose a block. At the end of the reducer phase, each reducer would have proposed a block which had received the consensus from the miners present in its shard.
Step 5: In the final consensus phase, we aim to use Byzantine fault-tolerance algorithms/proof of stake algorithms in order to obtain the consensus of all the shards on the blocks proposed by the reducer phase.

The consensus algorithm we apply in the final consensus phase and reducer has an heavy impact on the performance and throughput transaction rate which we obtain at the end All the intershard and intrashard communication are handled by Apache Kafka [21] using the publisher subscriber models.

3.1 Actors of Solution

In the rest of the solution, we will use these actors:

- Miner
  Miners validate new transactions and propose new blocks and record them on the global ledger and receive an incentive when they solve the complex mathematical problem.

- Leader
  Leader chooses ‘h’ members from each shard to form a validation group.
- Validator
  These are miners are subset of global miner list who validate blocks proposed by each shard which are added to the global ledger.

3.2 Updation of Transaction Pool

Lambda architecture can be used for efficient and real-time batch processing of transaction pool. It mainly consists of these three layers: batch layer, service layer, speedup layer.

Speedup Layer
This layer is primarily used to provide low latency updates. Speedup layer uses an incremental computation approach rather than a batch computation approach. Speedup layer relies on the fact that the transaction data stored is transient and modicum in nature. Processing data on a smaller scale provides greater design flexibility. Speedup allows complex computations to take place on these real-time views. Speedup layer aims to provide low latency updates to the batch views. Therefore, all the real-time views are directly supplied to the transaction-miner mapper.

There two major functions of the speedup layer:

- Storing the real-time views.
- Processing the input data stream so as to update the real-time transaction views.

Batch Layer
The batch layer is used to store the immutable growing transactional pool. The batch layer is responsible for creating real-time batch views for the service layer to process the blockchain transactions. The batch layer needs to be able to do two things:

- Store an immutable, constantly growing transactional pool.
- Create and continuously update batch views for the service layer.

The serial processing of transactions results in low scalability; thus, we aim to do batch processing and process these blockchain transfers using MapReduce. MapReduce is best done using batch processing paradigm.

Service Layer
Service layer is used to process the batch views that it receives from the service and speed layer. When new batch views are available, the serving layer automatically swaps in so that more up-to-date results are published on to the blockchain. The service layer is distributed among several machines to improve horizontal scalability using distributed frameworks. The major concerns of the service layer are ensuring low latencies and high throughput by using distributed frameworks.

3.3 Implementation Workflow

Referring Fig. 1.

- Mapper Phase

Step 1: Miner Mapper

Working: Miner mapper is used to bifurcating the available blockchain miners into n shards.
Input: Each miner is assigned a unique number and M_id indicates the miner id.
Output: M_id indicates the miner id and S_id for shard id.

Step 2: Transaction Mapper

Working: Bifurcate the blockchain transactions into k shards.
Input: All transactions in the transaction pool are referenced using two tuples < tx_id, tx > where tx_id denotes unique transaction ID and tx indicates transaction parameters of blockchain.
Output: The transaction mapper outputs a two-tuple output < tx, S_id > .

The accumulated transactions are divided into k splits, and the transaction mapper outputs a key as transaction and S_id the shard id to which the transaction belongs.

Step 3: Transaction-Miner Mapper

Working: 'k%n' transaction shards are mapped to each miner shard. If $(k = n)$, then one-one mapping will take place between miner shards and transaction shards.
Input: 'k' transaction shards and n miner shards.
Output: The transaction-miner mapper outputs a two tuple <S_id, <List(txns), List(Miners) $\gg$ S_id which indicates miner-transaction shard, List(txns) indicates the list of transaction shards in a miner-transaction shard and List(Miners) indicates the list of miner shards in miner-transaction shard.

Step 4: Reducer Phase

Working: used to apply consensus(aggregation) on the individual mappers. The number of reducers is decided using Eq. (2).
Input: Receives two-tuple <S_id, <List(Txns), List(Miners) $\gg$ List(txns) indicates the list of transaction shards in a miner-transaction shard list(miners) which indicates the list of miner shards in miner-transaction shard.

Output: Miner transaction shards apply consensus algorithms on the transaction shards and propose new blocks. The output of this layer is a two tuple <B_i, S_id> where 'B_i' represents the block proposed by shard *i*.

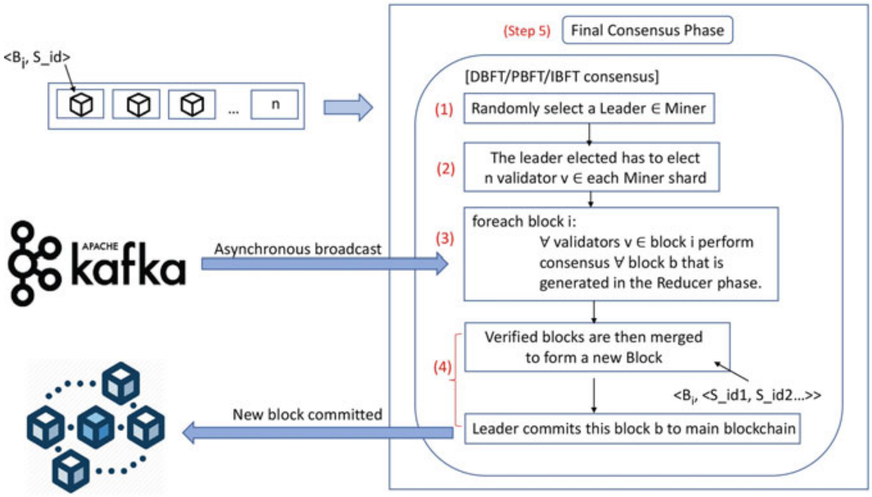
Algorithm-

```
Procedure reducer (List (<S_id, <List(Txns),
List(Miners)>>) returns list (<Block $B_i$, S_id,
List(Miners)>)
Foreach shard i <-0 to i<-n-1 with step=1 do
  Step 1: ∀ miners m ∈ shard i CreateBlock(List(Txns))
  Step 2: ∃miner m ∈ shard i ProposesBlock(List(Txns))
          returns $B_i$
  Step 3: d<- ∀ miners m ∈ shard $i$ perform
          consensusAlgorithm(Block $B_i$)
          If(d==false) then
            DiscardBlock(Block $B_i$)
End Foreach
```

Step 5: Final Consensus Phase

Working (*Referring* Fig. 2).

The consensus algorithms used in this phase are Byzantine Fault Tolerance Algorithm and its variants (based on the criteria mentioned in Table 1). Byzantine fault



<B_i, S_id>
...
n
Asynchronous broadcast
New block committed
(Step 5)
Final Consensus Phase
[DBFT/PBFT/IBFT consensus]
(1)
Randomly select a Leader ∈ Miner
(2)
The leader elected has to elect n validator v ∈ each Miner shard
(3)
foreach block i:
∀ validators v ∈ block i perform consensus ∀ block b that is generated in the Reducer phase.
(4)
Verified blocks are then merged to form a new Block
<B_i, <S_id1, S_id2...>>
Leader commits this block b to main blockchain

Fig. 2 Detailed implementation workflow architecture of the final consensus phase

Table 1 Comparison among consensus algorithms

	Proof of work	Proof of stake	Practical BFT	Delegated BFT	Federated BFT	Istanbul BFT	Delegated POS
Description	Node has to prove that it has performed amount of work	Node has to stake the ownership of the virtual currency	Node are sequentially ordered with one as the leader and the others as backup nodes	Delegates are elected, and then a speaker elected from them proposes a block to all the other delegates	Variants of the BFT by making them open-ended with respect to node participation	A proposer proposes a block and validators agree and broadcast their decision (geth implementation)	Elects the set of proposers and validators based upon stake and punish on malicious behavior
Used by	Bitcoin Blockchain, Ethereum (EthHash)	Cardano	Zilliqa, Hyperledger fabric	Neo	Ripple, Stellar	–	EOS
Token needed	Yes	Yes	No	No	No	No	Yes
Throughput	~7 txns/s	~10–15 txn s/s	~10,000 txns/s	~1000 txns/s	1500–2000 txns/s	400–1200 txns/s	50–250 txns/s
No. of commits	6 confirmations	2/3rd confirmations	f + 1 (f is no. of faulty node)	2/3rd of delegates agree	2/3rd confirmations	2f + 1 (f is no. of faulty node)	2/3rd confirmations
Possible attacks	51% attack	Nothing at stake attack Sybil attack	Nodes increase then msg count increases	>33 % dishonest delegates	Malicious validators	If honest nodes are less than dishonest	Low voter turnout attack
Scalability of peer network	High	High	Low	Low	High	Low	High
Cost of participation	Yes	Yes	No	No	No	No	Yes
Transaction finality	Probabilistic	Probabilistic	Immediate	Immediate	Immediate	Immediate	Probabilistic

tolerance algorithms provide better throughput compared to PoW and other consensus algorithms when there are limited number of participants participating in the consensus phase. The cost of participation in Byzantine fault tolerance and variants consensus is less compared to PoW and PoS, thus enabling lucid addition of nodes into the consensus phase. In BFT and its variants, only a part of the miners participate in the consensus rather than the whole, thus reaching faster consensus and ameliorating the validation of unconfirmed transactions in the transaction pool. They provide faster consensus in limited number of nodes compared to other consensus models. From the list(miners) which we receive from the reducer phase. We select a leader from the global miner list who samples h miners from each shard and form a group of h*n. This group formed needs to perform the final consensus on the blocks B0 to Bn-1. All the members of the group that has been formed can communicate with each other asynchronously using Apache Kafka's publisher subscriber model and can exchange data between each other seamlessly. Blocks which are of the form <Bi, <S_id1, S_id2, … ≫ indicate that they have received consensus from all the validators, and the block is committed to the main blockchain.

Algorithm-

```
STEP1: Randomly select a leader ∈ globalList(Miners)
STEP2: The leader elected has to elect h miners v∈ $shard_i$
using randomly to form a group of h*n members known as
validators Set V
STEP3: Foreach Block i <-0 to i<-n-1 with step =1 do
          STEP3.1: $d_i$<- ∀ validators ∈ $V_i$ perform
                  consensusAlgorithm(Block $B_i$)
        END FOREACH
STEP4: Foreach Block i <-0 to i<-n-1 with step =1 do
          If ($d_i$==true) then
            commit the blocks $B_i$ which are of the form
            <$B_i$, <$S\_id_1$, $S\_id_2$, $S\_id_3$..., $S\_id_n$> to the
          mainchain.
          Else
          DiscardBlock(Block $B_i$)
      END Foreach
```

3.4 *Estimating the Optimal Values of Mappers and Reducers*

There are several parameters in MapReduce that need to configure to get better performance than the baseline performance that MapReduce provides. Tuning of hyperparameters is necessary to get optimal performance [19].

Determine the Number of Mappers

$$M = (P - I) * K \tag{1}$$

- 'M' denotes the number of mappers.
- 'P' denotes the number of physical cores.
- 'I' denotes the number of reserved cores.
- 'K' denotes the CPU hyperthreading factor which ranges from [0.95 to 1.75].

Upper limit on the number of mappers is $M = F/B$.

where $F =$ input file size supplied to the cluster
$B =$ block size that is used by the cluster.

Determine the Number of Reducers

$$R = K * (\text{number of nodes} * \text{mapred.reduce.parallel.copies}) \tag{2}$$

- 'K' denotes the CPU hyperthreading factor which ranges from [0.95 to 1.75].
- 'Mapred.reduce.parallel.copies' is the maximum number of reducers that can execute in parallel.
- 'R' denotes the number of reducers.

The upper and lower limits on the number of reducers should be in the range [C/2, 2 * C] wherein C denotes the number of CPU cores. Apply 2/3 mapper technique which states that the number of reducers should be 2/3 that of the number of mappers estimated.

These formulas above have been used to determine the values in our proposed blockchain solution. These starting values obtained by these formulas act as a starting point to bifurcate the mining and transaction pool of our blockchain solution. These parameters help to determine the processing power required to run this architecture, and these hyper parameters can be fine-tuned over time.

3.5 Comparative Analysis Among Consensus Algorithms

This section provides a comparative analysis between consensus algorithms based on the criteria mentioned in Table 1 [22]

In our solution, the type of consensus algorithm in the intershard and intrashard consensus phase affects the performance of our system. BFTs are apt for permissioned blockchains, whereas the rest are good for public chains. There is a clear trade-off between scalability of consensus algorithms and time consumed for reaching consensus. Hence, BFTs are less scalable compared to the rest. Apart from PoW, PoS and Dpos, all the other consensus algorithms have the capability to revert the transaction from the confirmed block. Transaction finality is immediate in BFTs, whereas the others have probabilistic finality. By reducing the participation of peer nodes in consensus, we can certainly boost the transaction rate in a blockchain architecture.

Using the above parameters, we choose in the consensus algorithms for step 4 and step 5.

3.6 Performance Analysis Between the Proposed Solution and Existing Solutions

- *Block Commit Time*: required average time to commit the block to the main chain since it was created. Our solution aims to optimize this time by sharding the transaction pool compared to other solution by using Hadoop lambda architecture (Table 2).
- *Transaction Confirmation Time*: average time needed for a transaction to be confirmed into a mined block. Our solution uses faster consensus mechanism at intershard and intrashard consensus layers compared to other solutions.
- *Mempool Size*: Our framework can handle differential Mempool loads with the help of the lambda architecture compared to the other existing solutions.

Table 2 Comparative analysis between the proposed solution and existing solutions using performance metrics

Criteria	Proposed solution	Zilliqa and similar
Block commit time	Low	Moderate
Transaction confirmation time (depends on consensus)	Low to moderate	Moderate
Mempool size	High	Moderate
Network down time	Low	Low to moderate
Intershard communication	Moderate	Low
Transaction throughput (TPS)	High	High

- *Network Down Time*: Recovery of nodes in our solution is possible due high availability and replication features of Hadoop which are lacking in Zilliqa.
- *Intershard Communication*: Intershard communication in our solution is easy compared to other solutions because of Apache Kafka which provides seamless interaction among the shards which is hindrance to other solutions.

4 Conclusion

We have looked at an interesting way of how we can leverage big data concepts to resolve the scalability concern in distributed ledgers. For the past decade, the amount of reliance of user and application generated data have increased manifold. Big data technology has enabled us to scale at par with the increase in demand. Blockchain is now being viewed as a potential technology of the future, replacing the current centralized architectures.

In this paper, by focusing on the scalability aspect in blockchain, we have addressed one of the potential points of hindrance for one of the most important technologies of the future. The framework aims to integrate the well-established big data *sharding* frameworks with blockchain to ameliorate the scalability of blockchains. The paper proposed an initial overview of our research in the blockchain field and a framework with native big data *sharding* framework called *MapReduce* and aims to use an integrated *MapReduce* framework like Apache Spark for future work.

We aspire to integrate the present big data solutions with the present blockchain solutions.

References

1. Buterin V (2015) Notes on scalable blockchain protocols
2. Satoshi N, Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. Bitcoin
3. Buterin V (2013) Ethereum white paper: a next generation smart contract & decentralized application platform. Ethereum
4. Buterin V (2014) Proof of stake: how i learned to love weak subjectivity
5. The raiden network (2018) https://raiden.network/
6. Poon J, Buterin V (2017) Plasma: scalable autonomous smart contracts. White Paper
7. Teutsch J, Reitwießner C (2017) A scalable verification solution for blockchains. https://People.Cs.Uchicago…, p 50
8. Donald Miner AS (2014) Map reduce design patterns
9. Reilly O, Seu M (2012) Hadoop, the definitive guide. Online
10. Knox Gateway (2019) https://knox.apache.org/
11. Apache Ranger (2019) https://ranger.apache.org/
12. Abraham I, Malkhi D (2017) The Blockchain Consensus Layer and BFT. Bull EATCS
13. Wood G (2017) Polkadot: vision for a heterogeneous multi-chain framework. Whitepaper
14. Kwon J, Buchman E (2018) A network of distributed ledgers
15. Kwon J (2014) TenderMint : consensus without mining

16. Team TZ (2017) Zilliqa technical whitepaper. Zilliqa
17. Using Merklix tree to shard block validation | Deadalnix's den (2016) https://www.deadalnix.me/2016/11/06/using-merklix-tree-to-shard-block-validation/
18. Charles Hoskinson (2017) Why are we building Cardano
19. Tannir K (2014) Optimizing Hadoop for MapReduce. Packt Publishing
20. Kiran M, Murphy P, Monga I, Dugan J, Baveja SS (2015) Lambda architecture for cost-effective batch and speed big data processing. In: Proceedings—2015 IEEE International Conference on Big Data, IEEE Big Data 2015
21. Shapira G, Narkhede N, Palino T (2017) Kafka: the definitive guide—O'Reilly Media. O'Reilly Media
22. Baliga A (2017) Understanding blockchain consensus models. Whitepaper

Cryptocurrency Token: An Overview

Mahesh Shirole, Maneesh Darisi, and Sunil Bhirud

Abstract With the advent of blockchain, the trustless transactions between cross-border parties becomes easy. The profusion of eclectic tokens coming into the cryptocurrency world, one primary requirement is to enable robust and secure exchange between different varieties of tokens. With the rise of ICOs into the cryptocurrency market, the funding of research-based projects has become meteoric in nature. The dearth of knowledge about these different varieties of tokens is blocking amateur developers to unlock the true potential of the blockchain technology. There are myriad tokens, which are either fungible or non-fungible tokens that are used to represent different types of assets and research projects. This paper mainly focuses to provide an overview of tokens and comparative analysis for token standards available in the present cryptocurrency world.

Keywords Blockchain · Cryptocurrencies · Token · ICO

1 Introduction

Cryptocurrencies are digital currencies that are still in their embryonic stage and have been gaining a lot of attention worldwide. Cryptocurrency is neither government-issued nor government-regulated currency. Cryptocurrency can be used as a medium of exchange and to perform monetary transactions, in the same way as the printable currency/bills can be used. Although the number of blockchain use cases is augmenting as the time progresses, the lack of understanding of this new technology and tokens is restricting the use of cryptocurrencies widely. The first ever cryptocur-

M. Shirole · M. Darisi (✉) · S. Bhirud
CE & IT Department, Veermata Jijabai Technological Institute, Mumbai 400019, India
e-mail: mdarisi_b15@it.vjti.ac.in

M. Shirole
e-mail: mrshirole@it.vjti.ac.in

S. Bhirud
e-mail: sgbhirud@ce.vjti.ac.in

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_12

rency bitcoin [13] was created by Satoshi Nakamoto introduced in the year 2009. Subsequently, the development of blockchain technology has lead to the emergence of Ethereum [1]. Ever since advent of these cryptocurrencies in the market, it has attracted several potent investors. Roughly, there are about 2116 cryptocurrencies in the cryptoworld and keep augmenting regularly. These cryptocurrencies have a market capital of $119,068,338,608 [9] by the end of 2018. They have changed the way how cross-border transactions take place.

Initial coin offering (ICO) is the means of crowdfunding and thus leads to the creation of a new cryptocurrency. ICOs sell tokens/coins to the different stakeholders in exchange of legal tender or cryptocurrencies. It uses several cryptocurrency protocols to a create token on the top of the existing blockchain. ICOs have exponentially increased in the last two years with launch of ERC20 open standard token protocol. In last year 2018, total 686 new ICOs are launched, with a peak of 144 in May, raising total of $21,498,711,596 fund [2].

DApps used to define their own token standard and implementation for their private currency. With the launch of ERC20 token standard describing the rules and standards for cryptocurrency tokens, all the DApps and start-ups are using ERC20 token standard. It helped to uniformly understand tokens, its format and a way to interact with tokens. As of January 16, 2019, a total of 162,906 total token contracts are found on Etherscan [9].

In this paper, token standards are explored on different aspects such as standard, attack vectors, use cases of the attacks, improvements to ERC20 standard and other secure token proposals. This paper is organized as follows: Sect. 2 discusses the cryptocurrency tokens. Section 3 discusses token attacks and their possible solutions. A brief comparison of tokens is given in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Cryptocurrency Tokens

Cryptocurrencies can be classified based on their: (a) **usage**: fungible and non-fungible, (b) **type of implementation**: currency token (payment), utility token and security token. Fungibility is one of the essential characteristics of the currency. It can be used to represent anything that is interchangeable in the real world. *Fungible tokens* possess currency-like properties rather than unique and valuable assets. These tokens are interchangeable, uniform across platforms and are divisible into smaller units. All fungible tokens are based on ERC20 standard. *Non-fungible tokens* enable people and organizations to think beyond the cryptocurrencies, such as IDs and certificates. Non-fungible tokens are uniquely identifiable during trade, i.e., interaction and circulation. These tokens are non-interchangeable, unique in nature and non-divisible into smaller units. ERC721 standard is also used for non-fungible tokens on Ethereum blockchain.

Inherently, every currency token is associated with its own unique blockchain. Blockchain platforms serve as a medium for payment of goods and services. Currency tokens are used to perform monetary transactions using digital currencies rather than fiat currencies. Bitcoin and ether are paragons of cryptocurrencies and have their own blockchain platform.

2.1 Utility Tokens

Utility tokens are digital assets, which are built to support the structure of investor's payment mechanism. Utility tokens are handled using DApps. The proliferation of these tokens is enabling blockchain to affect various spectrums of the industry. Utility tokens are provided by businesses; the holder of the utility token gets access to different functions provided by businesses for trading tokens. Most of the tokens available on the blockchain are utility tokens.

ERC20. ERC20 is an open standard protocol that defines the software interface to implement tokens in Ethereum ecosystem. With ERC20 token standard protocol specification, all ICO developers now implement their own code with same method names and their arguments. Thus, improving the interoperability of different tokens implemented by different developers. This reduced the complexity of implementation of tokens, uniform interface and increased the rate of liquidity of the different tokens. ERC20 token standard interface includes six methods: $totalSupply$, $balanceOf$, $transfer$, $transferFrom$, $approve$ and $allowance$ and two events: $Transfer$ and $Approval$. In short, the standard facilitates to share, transfer, exchange and trade tokens seamlessly through cryptoworld.

ERC223. ERC223 [6] has suggested an improvement to ERC20 token standard. ERC20 suffers token losses due to $transfer$ function in the contract, which does not support token receiving and handling mechanism. The total loss estimated as on 27 Dec 2017 is $3,000,000 [9]. ERC223 token standard is backward compatible with ERC20 as it uses same interface. However, it requires contracts to implement the $tokenFallBack$ function.

$$function\ tokenFallback(address\ \_from, uint\ \_value, bytes\ \_data)$$

ERC223 is applicable to new contracts rather than old deployed contracts.

ERC777. ERC777 [8] token standard allows a new way to interact with contracts with the help of ERC820. ERC777 is backward compatible with earlier token standards and thus mitigates the problem of ERC223 to modify the contract. ERC777 takes support of ERC820 standard, which is a contract registry that will verify whether a contract is token compatible or not. In case if ERC777 token contract is not registered or compatible, then the transaction will raise an exception, thus preventing loss of tokens. The only problem with ERC777 is that it uses a central registry for smart contracts lookup.

ERC721. ERC721 [7] is a token standard that defines an interface to allow non-fungible tokens to be managed, owned and traded by a smart contract. It defines functions: $name$, $symbol$, $totalSupply$, $balanceOf$, $ownerOf$, $approve$, $takeOwnership$, $tokenOfOwnerByIndex$, $transfer$ and $tokenMetadata$. It defines two events: $Transfer$ and $Approval$. The $takeOwnership$ and $transfer$ functions define how the contract will handle token ownership and how ownership can be transferred. The function $tokenMetadata$ makes token non-fungible by its unique set of attributes. ERC721 does not mandate implementation of a token metadata or restrict addition of extra functions.

2.2 Security Tokens

Security tokens need to follow myriad federation rules and regulations; hence, they are complex. They are the type of assets, which assure positive ROI on their holding. Such returns are guaranteed by the platform itself or the company, which had launched the security token. A security token shares many of the characteristics of both fungible and non-fungible tokens.

ERC1400. ERC1400 [3] is a simple restricted token standard developed for corporate governance and banking considering securities laws. It is fully open source ensuring security, quality and interoperability of tokens. ERC1400 tokens are partially fungible. One ERC1400 token issued by one entity may not be exchangeable with another, because these tokens have different properties and a group of owners. ERC1400 is an umbrella standard that incorporates ERC1594 [5] (core functionality), ERC1410 [4] (partially fungible tokens), ERC1643 (document management) and ERC1644 (controller token operation) with some additional constraints to ensure these standards interoperate in a consistent manner.

R-Token. R-token [12] is a permissioned token allowing token transfer to occur only if they are approved by an on-chain regulator service. R-tokens extends ERC-20 tokens for regulated securities. Regulator services can be configured to meet security regulations such as KYC policies and anti-money laundering.

3 Token Attacks

Blockchain is immutable; hence, the contracts which are deployed on blockchain are immutable. Majority of contracts are written in solidity language; therefore, one must understand what are the common attacks in solidity. These attacks have caused huge losses due to loopholes in the contracts.

3.1 Approve Attack

Although 90% of the tokens present in the crypto world are compliant with the ERC20 standards, one of the significant flaws in its interface is in the definition of the $approve$ and $transferFrom$ methods. These methods can be exploited to withdraw

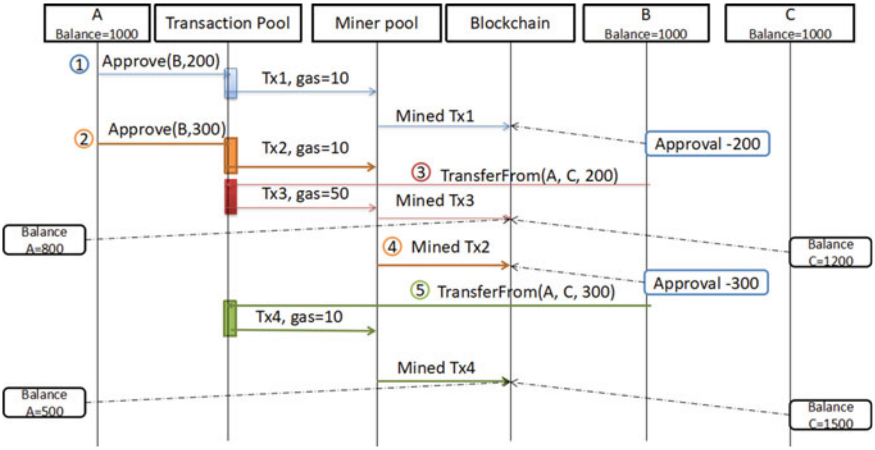


Fig. 1 Illustration of approve transfer attack using sequence diagram

tokens more than given allowance if approvals are given in succession. Approve function permits the given address to withdraw token amount from message sender token balance up to the specified value in single or multiple withdraws. Consider a scenario wherein there are two parties/accounts A and B, where account B also controls account C. It is assumed that all accounts initial balance is 1000 ether and the balance of account A is greater than or equal to $X + Y$ ether, i.e., 200+300 in Fig. 1. The illustration of the attack vector use case scenario is as follows referring Fig. 1.

STEP 1. Account A gives approval of X ether from his wallet to account B.

STEP 2. Account A decides to change the amount of approval from X ether to Y ether for account B and sends Y as the argument in the $Approve$ function.

STEP 3. Account B notices A's second transaction and before it gets mined, B sends the allowance of X ether that it had approved from A to C using $transferFrom$ function with high gas value to prioritize transfer.

STEP 4. If B's $transferFrom$ call gets executed before A's approve transaction, then B will get the ability to transfer another Y tokens to C.

STEP 5. A's attempt to change the allowance from X to Y has lead B to transfer $X + Y$ to C, whereas A never wanted to transfer $X + Y$ tokens to B. **Solution.** A simple change to the present interface may assist in making the ERC20 less error prone. Convert the approve function to a three-argument function from a two-argument function to prevent the above attack.

```
function approve(address spenderAddresss, uint256 currentValueOfAllowance,
uint256 ChangedValueOfAllowanceValue) returns (bool success)
```

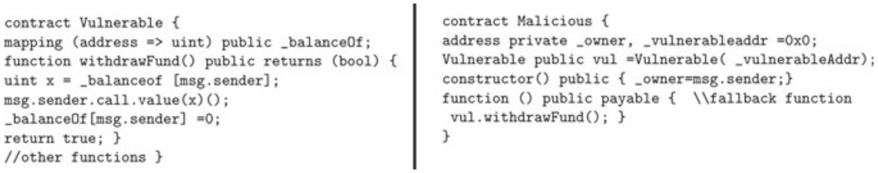
If current allowance for $spenderAddress$ is equal to $currentValueOfAllowance$, then overwrite it with $ChangedValueOfAllowanceValue$ and return true, otherwise return false. Accordingly, events $Transfer$ and $Approve$ can be changed.

3.2 Overflow Exploit Attack

The classical arithmetic integer overflow problem can be exploited to transfer large amount of tokens. This bug was noticed by PeckShield on April 22nd 2018. He found fallacious transaction which involved two large BEC token transactions. The reason was arithmetic overflow in statements like
$balances[msg.sender]+ = msg.value$;
Solution. OpenZeppelin [11] library has provided a transparent solution to counter this attack vector. OpenZeppelin has provided a solidity file known as $SafeMath.sol$, which provides several checks to avoid any overflow during basic arithmetic operations.

3.3 Reentrancy Attack

A non-recursive function of a contract should not enter in the same function before termination of the function. In reentrancy attack, an attacker tries to reenter the calling function of a contract with the fallback mechanism in solidity. In Ethereum, *call* function is used to transfer a value/data or to execute a function of same contract or another contract. The *call* function starts code execution of function and spends available gas for execution. It makes code vulnerable to reentrancy attack. As there is no gas limit for the call function, the fallback function in *call* function can run as long as it exhausts all gas allocated for that function or balance of the account. To understand this attack, consider example as discussed in [10]. Consider a wallet contract, named $Vulnerable$, shown in the following code in Fig. 2. Its statement $msg.sender.call.value(x)()$ will cause reentrancy attack, as it invokes the $message.sender's$ fallback function in order to send his balance of 'x' wei to the sender. A malicious user contract named $Malicious$ is as shown in the code in Fig. 2, where the contract address is initialized with $\_owner$ and another address is vulnerable wallet's address. When the malicious user withdraws balance from$Vulnerable$



```
contract Vulnerable {
mapping (address => uint) public _balanceOf;
function withdrawFund() public returns (bool) {
uint x = _balanceof [msg.sender];
msg.sender.call.value(x)();
_balanceOf[msg.sender] =0;
return true; }
//other functions }
```

```
contract Malicious {
address private _owner, _vulnerableaddr =0x0;
Vulnerable public vul =Vulnerable( _vulnerableAddr);
constructor() public { _owner=msg.sender;}
function () public payable {  \\fallback function
 vul.withdrawFund(); }
}
```

Fig. 2 A sample code to illustrate reenterancy attack

wallet using *Malicious* contract, then wallet calls users contract fallback function and in turn calls fallback function which executes *withdrawFund* function of the wallet contract repeatedly until it fails due to gas limit or balance.
Solution. To mitigate reenterancy attack avoid using *call* function. The *transfer*() and *send*() are safe against reentrancy attack since they limit code execution to 2300 gas units.

4 Utility Tokens Comparison

In this section, a comparative analysis of the utility tokens is presented based on fungibility, backward compatibility, token sale delegation and contract registration. The comparative analysis is shown in Table 1.

Fallacious transactions to anonymous addresses can result in the loss of ether; this issue is resolved by ERC223 (assembly code), ERC777 and ERC721 using a centralized registry ERC 820. ERC777 and ERC721 are backward compatible with the ERC20 standard and thus provide a easy mechanism for upgrading the presently deployed contracts. The major aim of using upgraded token standard is to avoid the loopholes in the present contract and to abate the present gas consumption rate. Except ERC721, other token standards are fungible.

Table 1 Comparison of utility tokens

	ERC20	ERC223	ERC777	ERC721
Fungible	Yes	Yes	Yes	No
Verify contract address	No	Yes	Yes	Yes
Back compatibility with ERC20	–	No	Yes	Yes
Delegate token sale	Yes	Yes	Yes	Yes
Usage ERC820 for contract registration	No	No	Yes	Yes

5 Conclusion

Sharp increase in the number of ICOs in the last two years had lead to the standardization of tokens. Cryptocurrency tokens are evolving and growing community effort to set interoperable standards. This paper discusses a list of representative token standards, their possible attacks and possible solution for the same. Possible loss due to faulty and incompatible token implementations can be decreased using appropriate mechanisms as discussed in this paper.

References

1. Buterin V (2013) Ethereum white paper: a next generation smart contract & decentralized application platform. https://doi.org/10.5663/aps.v1i1.10138
2. Coinschedule: https://www.coinschedule.com/stats.html. Last accessed Jan 2019
3. ERC1400: https://github.com/ethereum/eips/issues/1400
4. ERC1410: https://github.com/ethereum/eips/issues/1410
5. ERC1594: https://github.com/ethereum/eips/issues/1594
6. ERC223: https://github.com/ethereum/eips/issues/223
7. ERC721: Non-fungible token standard. https://github.com/ethereum/EIPs/blob/master/EIPS/eip-721
8. ERC777: https://eips.ethereum.org/eips/eip-777
9. Etherscan: Ethereum Blockchain Explorer. https://etherscan.io/
10. Li X, Jiang P, Chen T, Luo X, Wen Q (2017) A survey on the security of blockchain systems. https://doi.org/10.1016/j.future.2017.08.020
11. OpenZeppelin: https://github.com/openzeppelin/openzeppelin-solidity (2018)
12. Remeika, B., Amano, A., Sacks, D.: The Regulated Token (R-Token) Standard (2018). 10.1007/BF02162388
13. Satoshi, N.: Bitcoin: A Peer-to-Peer Electronic cash system. Bitcoin (2008)

Blockchain Research and Applications: A Systematic Mapping Study

K. S. Sagar Bharadwaj, Samvid Dharanikota, Adarsh Honawad, and K. Chandrasekaran

Abstract Brought to the limelight by the famous Bitcoin, blockchain has since evolved and now sees a lot of use cases apart from cryptocurrencies, such as in distributed storage systems, finance, health care, and so on. It is, therefore, an area of scrutiny by a lot of researchers and application developers. Significant amount of research on blockchain involves the application of blockchain technology to solve problems from various domains or improve the existing architecture of blockchain itself. The recent trend toward the decentralization of the Internet has given rise to many decentralized applications which also rely fundamentally on blockchain. In this paper, we conducted a systematic mapping study on blockchain technologies. The objective of the study is to identify and map various domains of research related to blockchain and recognize possible directions for future research. We do so by formulating a set of well-defined research questions and providing answers to them.

Keywords Blockchain · Systematic mapping study · Blockchain research · Blockchain applications

K. S. Sagar Bharadwaj · S. Dharanikota (✉) · A. Honawad · K. Chandrasekaran
Department of Computer Science and Engineering, National Institute of Technology
Karnataka, Surathkal, Mangalore 575025, India
e-mail: samvid.dharani@gmail.com

K. S. Sagar Bharadwaj
e-mail: sagarbharadwaj50@gmail.com

A. Honawad
e-mail: adarsh2397@gmail.com

K. Chandrasekaran
e-mail: kchnitk@ieee.org

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_13

1 Introduction

The blockchain is a distributed data storage ledger with certain key features that rely heavily on cryptography. This structure, which is replicated over all nodes in a network (dependent on the type of blockchain), is fundamentally a cryptographically linked chain of blocks, similar to a linked list data structure. Each block, along with data, consists of a hash of the previous block in the chain, until the genesis block (the first block) whose hash field is 0.

This property ensures a key aspect of blockchain, immutability, in the following way: Assume the data in block n is being tampered with. This change would need to be followed by a recalculation of that block's hash, which is present in block number $n + 1$. This would lead to re-computation of the hash of block $n + 1$ and the following blocks until the latest block. This implies immutability because the creation of a new block in the network now becomes difficult (dependent on the blockchain protocol), leading to honest nodes which do not tamper with the data.

Blockchain also ensures privacy to an extent as the user identities are just cryptographic keys and not their information or credentials. Hence, the user's personal details are not compromised in case of a breach. The blockchain is decentralized and so no single node manages the network and the blockchain itself. This fact eliminates any single point of failure issues that centralized systems such as most of the current Internet technologies face.

It is to be noted that a blockchain is not complete without the network architecture that buttresses it, just as in any distributed system.

These properties of decentralization, immutability, and privacy make it an attractive architecture to be used in various use cases.

Moreover, as a consequence of the distributed nature of the blockchain architecture, several issues that pertain to standard distributed systems such as consensus and consistency among others also apply here. Moreover, because there is no one perfect solution to the issues in distributed systems, there is always scope for improvement concerning technologies surrounding blockchain.

This paper is a systematic mapping study of the work done in the area of blockchain. Several attempts [4, 66] have been made to prepare such studies earlier, but they are now outdated owing to the rapid progress in the research of blockchain systems. Furthermore, they have included only the papers that are openly accessible. Mapping the published literature, we highlight the areas in blockchain that are actively being researched, and we also highlight the current and potential use cases of blockchain.

Further, Sect. 2 describes the research methodology that we have used in conducting our mapping study. We define the research questions that we attempt to answer and the motivation behind the same. In Sect. 3, we elaborate on basic publication-related information such as the year of publication and types of publications obtained after our search process. Section 4 answers the defined research questions. Finally, we present concluding remarks in Sect. 5.

2 Research Methodology

We have followed the standard procedure for a systematic mapping study, with minor changes as applicable, as defined in [46]. We have documented our entire search procedure along with the results online [51].

2.1 Research Questions

The first step is to identify the research questions that are to be answered with this systematic mapping study. The questions have been identified and elaborated as follows:

RQ1: How have publication amount, frequency, and research topics changed over time?

This question seeks to answer how the trends in blockchain research have changed over time, from its inception with the Bitcoin. We seek to identify areas of research in blockchain that are growing and the areas that are gradually being ignored by answering this research question (Fig. 1).

RQ2: What are the use cases of blockchain technology?

Blockchain is seeing applications in a wide variety of use cases beyond cryptocurrencies, for which it was designed initially, especially with the shift toward the decentralized Web. The answer to this question will outline the domains in which blockchain is being used, providing solutions to the problems in those areas.

RQ3: What are the areas of current research in blockchain?

While RQ2 focuses on the domains where blockchain is used, RQ3 aims to elaborate on enhancements and optimizations that are being made to blockchain architectures themselves. Blockchain architectures in the current scenario are not perfect and have

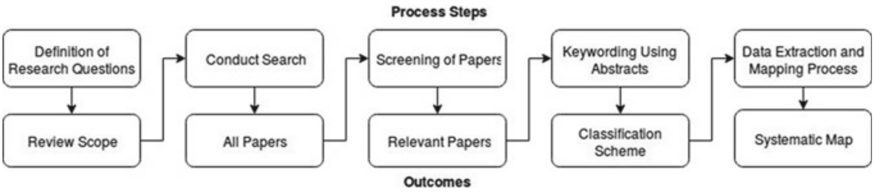


Fig. 1 Research methodology [46]

many drawbacks in terms of scalability and transaction processing speed, among others, and this question addresses the work done toward improvement in these aspects.

RQ4: How is research on blockchain distributed geographically?

We aim to provide an estimate of the number of papers published in different countries. The answer to this research question enables one to study where in the world blockchain research is being carried out.

RQ5: What is the future research direction for blockchain?

This question aims to draw conclusions from the above questions and predicts where the research in blockchain is heading toward and the areas researchers are most likely to pursue. We also intend to point out some areas of research that were lacking attention at the onset of blockchain but are now significant areas of research.

2.2 Selection of Paper Sources

Our aim was to select popular sources that would contain the most significant number of publications. IEEE Explore, ACM Digital Journal, SpringerOpen, and ScienceDirect were considered owing to having an extensive collection of papers in the blockchain domain. Springer, although as popular as the above databases, was not considered because searching for the keyword 'blockchain' resulted in only book chapters and irrelevant papers.

2.3 Conducting the Search

The second stage of the systematic mapping study is to form the search strings that are used for the search of the papers and to conduct the search. The search strings that we used to obtain results from different sources can be found at [51].

In our database search process, only those publications having the keyword 'blockchain' in their title or keywords section were selected. This eliminates the consideration for papers that do not have the word in either field but refer to the same in the content of the paper.

2.4 Screening of Relevant Papers

We then applied filters on the paper databases to extract only journal and conference papers. We selected only those sources that are peer-reviewed. Following the database filtering, we screened the papers first based on their titles and then abstracts and further excluded/included papers based on a set of inclusion and exclusion criteria:

Exclusion Criteria:

- Review/summary/secondary papers—these papers do not pertain to the scope of a systematic mapping study and hence were removed
- Book chapters/keynotes/case studies/work-in-progress/news articles papers—for the same reason as above
- Duplicate papers
- Papers where blockchain is not the main area of focus
- Papers that focus on an economic/financial point of view
- Papers that address issues pertaining to a specific country and do not focus on generic issues
- Papers not written in the English language
- Commentaries/news.

Inclusion Criteria:

- Papers that introduce novelty in blockchain
- Papers that explained the use of blockchain in other domains
- Papers that enhance blockchain.

Due to the large number of papers published on blockchain, we had to stick to rigorous filtering criteria to reduce the number of papers for efficient classification. There is a possibility that we may have missed out some relevant papers.

Table 1 shows the number of research papers that we considered initially and the number of papers left at the end of application of the corresponding filtering criteria.

Table 1 Keyword search results

Filtering Phase	IEEE	ACM Digital Journal	Science Direct	Springer Open	Total
Conducting the search	820	279	183	43	1325
Search filters applied	773	251	116	43	1183
Title screening	639	198	62	22	921
Abstract screening	421	135	45	10	611
Duplicate removal	–	–	–	–	604

2.5 Data Extraction and Mapping

After filtering relevant and vital papers, we have identified categories that each of the papers belongs to, by reading their abstracts. The list of these classifications along with the papers that belong to these classifications is listed in [51]. There are a total of 123 categories that we have identified. We have classified the chosen 604 primary papers under these 123 categories. Several papers were found to belong to multiple categories. Compared to [66], which has classified a total of 41 primary papers into 14 classifications, our mapping study is done on a much larger scale owing to the rapid increase in research on blockchain technologies.

3 Publication Statistics

Papers obtained after all filtering criteria were applied and analyzed to give the following inferences.

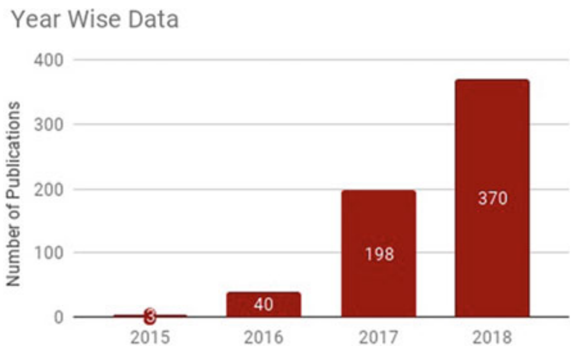
3.1 Search and Selection Results

The search string that was formed based on the relevant keywords, and exclusion criterion was used for searching on the various publications sites. The results were obtained with details like title, abstract, keywords, authors, citations, and other vital details. We began the work on title screening. We excluded several papers marked as Demos and Tutorials. We also found several papers with single pages, which do not have any significant contributions or citations and hence removed them. We also went through the abstracts of each paper that cleared the title screening phase and identified review papers, secondary papers like literature surveys, and papers that do not focus on blockchain. We removed such papers from consideration. The first phase of the classification of papers was also done along with the abstract screening phase. After title screening, abstract screening, and duplicate removal phases, we had a total of 604 primary papers that we could consider for the purpose of this systematic mapping study.

3.2 Publication Year

Blockchain research started gaining its popularity in the year 2015 (according to publication searches). Between 2015 and 2018, the number of papers in the domain of blockchain has risen significantly. One of the attributes of this rise is due to the

Fig. 2 Year of publication



advent of newer blockchain technologies that are more capable than Bitcoin [42] in factors like transaction speed, storage, and the ability to execute code.

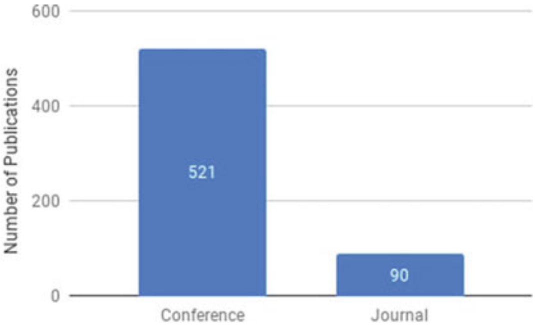
With the increase in such factors, the scope of blockchain has broadened and today blockchain is being used in several areas like health care, supply chain, edge computing, and education apart from just finance. One notable example would be the Ethereum blockchain [63]. Ethereum was proposed in 2014 and brought with it the ability to execute user-defined Turing complete code, called smart contracts. With this, Ethereum swept its way into several Internet of Things (IoT)-based applications. Smart contracts were then utilized in several other domains like health care, energy market, and vehicular networks.

Figure 2 gives a count of the publications with respect to the year of publication after applying all the filtering criteria.

3.3 Publication Type

These numbers were obtained after the application of all the filtering criteria. Figure 3 shows the count of papers obtained from the different publication types. Papers presented in workshops and symposiums have been included under conference.

Fig. 3 Papers classified according to publication type



4 Discussion

Based on the results of our search, we answer the research questions posed above.

4.1 *RQ1: How Have Publication Amount, Frequency, and Research Topics Changed over Time?*

Figure 2 shows the rapid growth in number of publications related to blockchain technology over time. We analyzed the relative percentages of research areas in each year, i.e., how much of the total research in a particular year is conducted in different research fields. In the earlier mapping study on blockchain technologies [66], most of the research was focused on the enhancement of Bitcoin. Around 80.5% of all the papers studied in [66] have focused on Bitcoin. However, the research scenario is largely different today, and the applications of blockchain have diversified.

Figure 4 shows a pie chart of relative percentages of publications classified under each of the research areas. The complete data and list of all the categories or research areas are published in [51]. Compared to [66], this represents a wide range of research possibilities. We also analyzed how blockchain research areas have evolved over time.

Figures 5, 6, and 7 represent the evolution of research areas over time. Many inferences about the evolution of research in blockchain can be drawn based on the data presented in these charts. Bitcoin was a major area of research spanning over 80.5% of all publications during the onset of blockchain research [66]. Bitcoin occupied 10.6% of all research in blockchain in 2016 owing to a boom in the potential applications of blockchain. This percentage reduced to 2.6% in 2017 and is less than

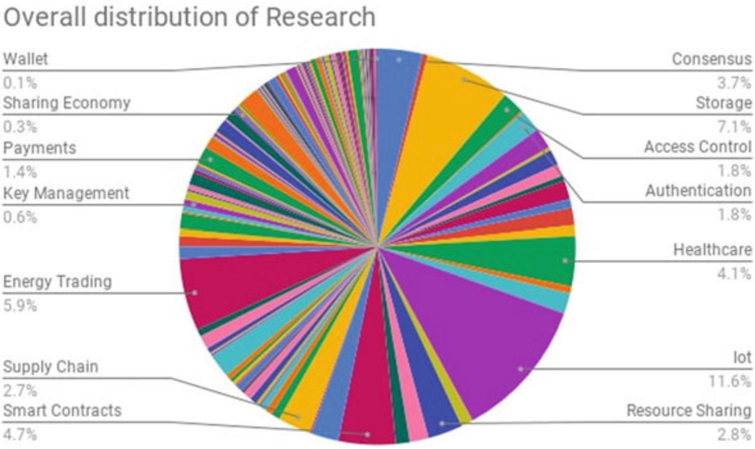


Fig. 4 Distribution of research areas in blockchain

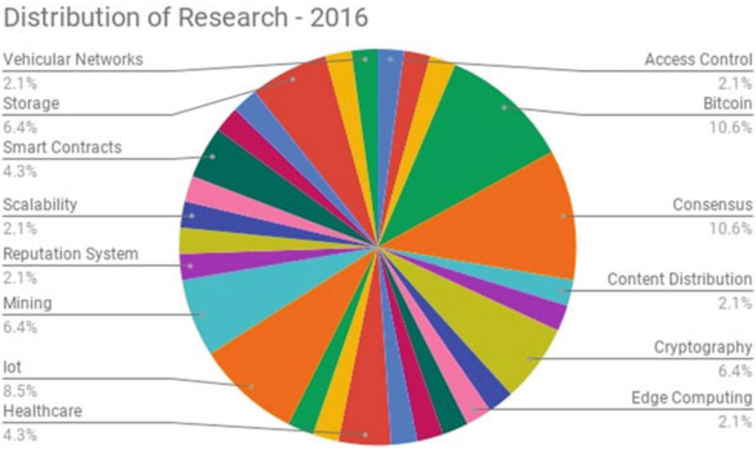


Fig. 5 Distribution of research areas in blockchain—2016

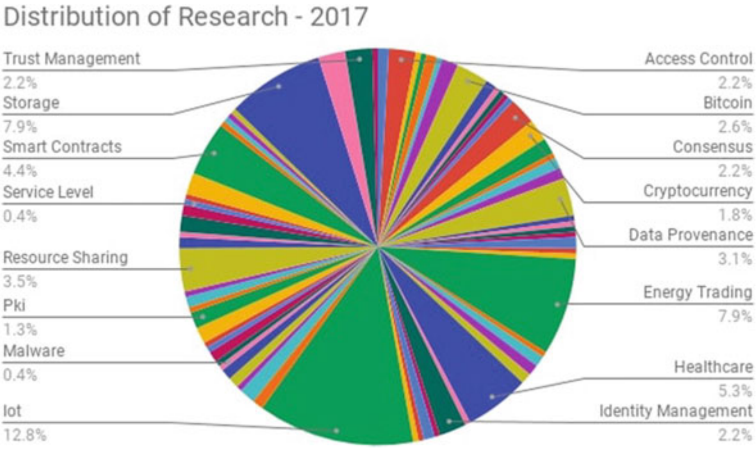


Fig. 6 Distribution of research areas in blockchain—2017

1% in 2018. This not only shows a shift of attention away from Bitcoin-oriented research but also points to the increased efforts put into generalizing the applications of blockchain. Another interesting result is the evolution of application of blockchain in the IoT. Reference [9] summarized different methods of adopting blockchain in the IoT domain, and the number of publications has gone up ever since. The application of blockchain in IoT occupied 8.5% of the total blockchain research in 2016. In 2018, research in blockchain and IoT increased to 11.3% with the increase in the total number of papers published. This shows that blockchain and IoT combination are one area which researchers are actively looking at. The number of research areas has also risen significantly since 2016 as shown by the number of segments in the pie

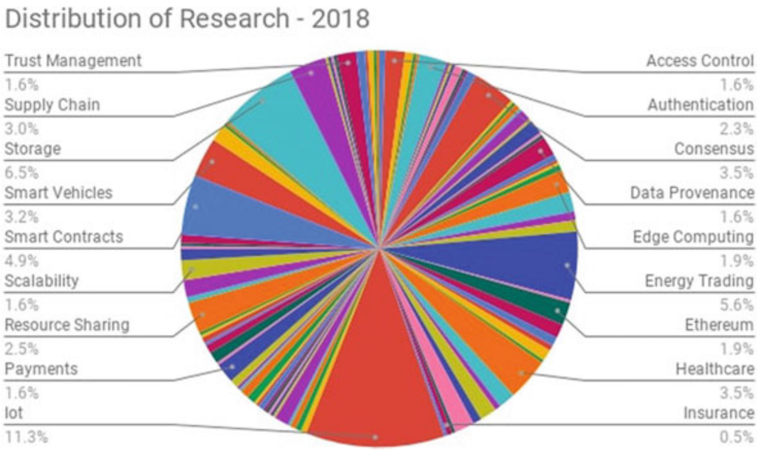


Fig. 7 Distribution of research areas in blockchain—2018

charts (Fig. 4). Many new areas like energy trading on smart grid occupy a significant portion of research today.

4.2 RQ2: What Are the Use Cases of Blockchain Technology?

From various areas where blockchain is being used today, IoT stands above them all (11.6% of the papers) followed by storage solutions (7.1%) and energy trading (5.9%). This is followed by other domains like health care (6.6%) and smart grids (5.7%). Blockchain aids applications in these domains by enhancing fundamental features which include authenticity, security, data integrity, data immutability, data privacy, data provenance, and data ownership among many others.

We have chosen some papers in certain important domains, and we provide a gist of the research conducted in each of those domains. All the papers mentioned in this section are referred to by their IDs in [51].

There has been a significant amount of research happening in the fields of IoT and blockchain, paving the way to many possibilities. In our study, we found 82 papers that directly deal with a combination of blockchain and IoT. [P160] [44] shows how blockchain can be used for access management in IoT scenarios. While current centralized methods do exist, scalability of such methods is limited. Blockchain with its distributed access control system for IoT seems to provide a new and better alternative. [P219] [22] uses blockchain as a method to build trust in consumers to trade their smart devices' data for incentives. They describe how the IoT device will be sold to a user by proving the devices integrity without a third party and provide a secure method of sharing the users data. However, blockchain remains computationally expensive, and one such paper [P561] [12] tackles the problem by introducing an optimized blockchain. The main idea is to create an overlay network of high resource devices to handle the blockchains operations while still providing end-to-end security and privacy to the low resource IoT devices.

Smart grid is vital in today's world to increase distribution of locally produced energy, mostly renewable energy. However, a centralized architecture often poses issues of reliability and privacy or anonymity. In our study, we found a total of 43 papers directly dealing with energy trading through blockchain transactions. A blockchain-based integration into the energy trading society has been proposed successfully by [P505] [3]. [P196] [41] offers a blockchain and smart contracts solution to tackle the distribution of energy from multiple sources and allows to handle payments securely.

Several blockchain-based solutions have been proposed to tackle problems related to data storage on the cloud. [P461] [33] introduces ProvChain which is an architecture to embed provenance data into blockchain transactions. Most of the storage solutions use blockchain architecture as middleware between users of the data and the data itself to enhance several features of the system including privacy, security, data provenance, auditing capabilities, anonymity, and so on. Reference [69] was one of the early papers to give details on the way a blockchain layer can provide enhanced privacy in storage systems. In our study, we have collected 51 papers that utilize blockchain as a storage solution.

Edge computing is used to offload computation required for mining onto edge devices, from mobiles, enabling mobile systems to participate in the blockchain network [P376] [35]. There are also pricing schemes designed for edge service providers (ESPs) [P429] [64]. Edge nodes also make use of distributed control systems, which in turn have function blocks as their main component. Smart contracts are used to implement these function blocks [P141] [57]. Blockchain is also used for trusted data sharing between edge nodes. Ideas are also proposed to reduce work done by mining to replace proof of work by proof of collaboration, catering to the limited computational and storage resources of edge devices.

Blockchains can also replace a traditional CA, as proposed by [P22] [65]. Additional x509 certificate extensions are proposed which facilitate smart contracts to handle the tasks of a CA such as issuing, storing, validating, and revoking certificates. A particular application of blockchain in verification of certificates for SSL/TLS-secured communication is proposed in [P212] [8]. A distributed PKI is proposed in [P211] [49] which supports a distributed certificate library, where the miners in the blockchain environment act as CAs, ensuring the correctness of certificates. Smart contacts have also been used to implement a dynamic trust protocol in PKIs [P592] [2]. [P338] [59] proposes a novel approach of creating a cloud-based PKI using blockchain, where certificate issuing is done on the cloud, and the blockchain is used to record the issued certificates.

We came across several unique use cases of blockchain during our studies. For example, [P286] [50] deals with mixed reality applications. [P276] [62] uses blockchain in a transaction processing system. [P201] [18] uses blockchain in a video surveillance system. [P266] [43], [P360] [26], and [P93] [55] use blockchain to collect and analyze data related to pollution. [P477] [37] and [P486] [25] use blockchain as a reviews framework. [P155] [52] uses blockchain to record work history of employees and aids in corporate management. [P138] [48], [P388] [23], [P467] [67], [P77] [34], and [P95] [24] deal with electric vehicles and charging stations. [P413] [20], [P512] [14], and [P521] [53] use blockchain as a framework to enhance Information Technology Operations (Ops). A list of all the categories and the corresponding papers for each category can be found at [51].

4.3 RQ3: What Are the Areas of Current Research in Blockchain Technology?

While blockchain is used in many domains as seen above, there has been intense research going on to enhance blockchain technology itself. In our study, we identified several papers that deal exclusively with generic blockchain. These papers do not narrow down on a specific use case or a domain but focus on features such as blockchain's storage scalability, transaction scalability, consensus protocols, formal analysis of blockchain, and so on. Figure 8 shows relative percentages of papers according to their primary focus.

We found a total of 26 papers dealing with consensus algorithms. In a decentralized system, we need algorithms to reach consensus among the participating nodes. Bitcoin, one of the first useful implementation of blockchain, uses the proof of work consensus algorithm. However, this algorithm is computationally expensive and faces several security threats. Efforts have been made to come up with new consensus algorithms like proof of luck, proof of stake, proof of trust, and even improvements on existing consensus algorithms. [P453] [40] describes the proof of luck method of consensus in trusted execution environments. The idea is to use a

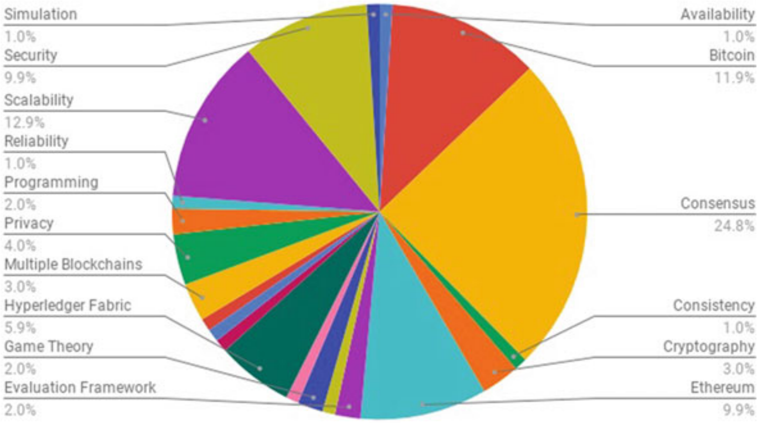


Fig. 8 Distribution of research on generic blockchain

random number generator to pick a consensus leader, offering equitably distributed mining with lower latency and energy consumption. [P52] [54] attempts to solve the 51% majority attack on the Bitcoin network by proposing a modified proof of work consensus algorithm.

[P252] [61], [P429] [64], and [P610] [68] model blockchain from a game theoretic perspective to prove the robustness and security features of some blockchain-based solutions. [P252] [61] models their content caching system based on blockchain as a Chinese restaurant game and analyzes the Nash equilibrium of the game. [P429] [64] models the edge computing service provider as a Stackelberg game. There have not been many studies conducted on rigorous mathematical proofs to prove security and safety properties that blockchain-based solutions claim to offer. [P335] [15] and [P336] [1] focus on formal verification of these properties. [P416] [10] and [P516] [45] discuss alternate solutions for programming languages that can be used on blockchain. Currently, Ethereum uses solidity as the primary programming language to write smart contracts [11]. [P416] [10] proposes a language named obsidian which the authors claim to be safer than solidity. [P516] [45] proposes simplicity, a typed, combinator-based, functional language without loops and recursion to be used in blockchain-based applications.

Ethereum is one of the most widely used platforms not just by decentralized application developers but also researchers to test their ideas. [P313] [5] proposes a query language specific to the Ethereum blockchain based on SQL. This was proposed to extract transaction and block details in the Ethereum blockchain and filter them based on transaction details within the block. Ethereum-specific research is also done in [P394] [21] where a tool is proposed to analyze Ethereum smart contracts for out-of-gas vulnerabilities wherein a smart contract's balance is locked permanently if it terminates abruptly when it runs out of gas, and this abrupt abortion is not handled properly.

In addition to this, several papers are dealing with issues of storage scalability, computational requirements, and faster and scalable consensus algorithms.

4.4 *RQ4: How Is Research on Blockchain Distributed Geographically?*

Figure 9 shows the comparison among the different countries where blockchain-based research papers have originated. The research is being significantly carried out forward by universities and industries in China (23.9%) and USA (13.6%) while all other countries have a contribution of less than 5%.

China seems to be the current hot spot for blockchain research. Blockchain research in China is encouraged by several dominant institutions including the Communist Party, Central Bank, Supreme People's Court, and the Bank of China [16]. Several strategic reasons for significant blockchain research in China have been described in [16].

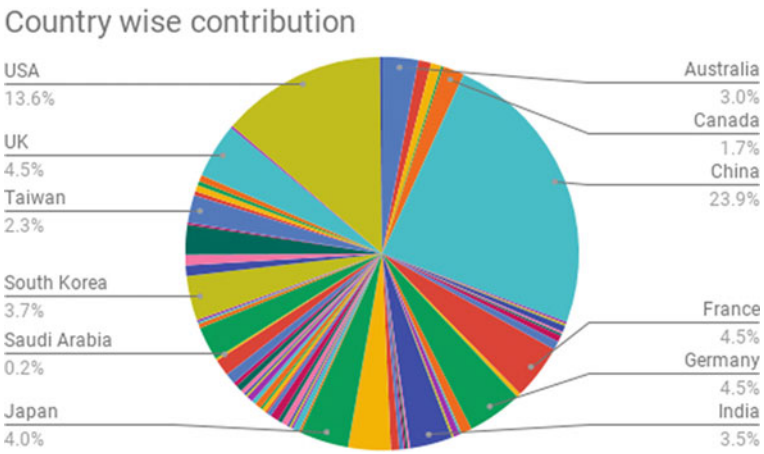


Fig. 9 Research papers classified according to the country

4.5 RQ5 : What Are the Possible Directions for Future Blockchain Research?

The previous research question focused on the current research areas in blockchain. We identify some possible future research areas related to blockchain by identifying the most popular fields that researchers are working on by conducting a search for literature in very recent years (2017 and 2018). This section reviews these research areas by providing a comprehensive overview of the identified fields.

Scalability of blockchains has been a burgeoning area of research interest. In the initial stages of blockchain that was popularized by Bitcoin [42], blockchain consensus algorithms like proof of work focused on scalability with respect to the number of nodes. However, as the number of transactions on the Bitcoin increased, there began considerable work on increasing the throughput of the Bitcoin network. Some early work includes Bitcoin-NG [17] which uses proof of work to elect a leader and allows it to add micro-transactions in the inter mining period. The GHOST rule [56] proposed a new conflict resolution method in proof of work mining that makes it safer to increase the block mining frequency, thereby increasing scalability in terms of the number of transactions. Replacing a linear chain of blocks with directed acyclic graphs (DAG), another method was introduced by [30] to include all mined blocks in the log if they are not conflicting.

Bitcoin lightning network [47] proposes the creation of micropayment channels between two concerned parties to increase scalability by deferring broadcasting transactions to the rest of the blockchain network. Sharding of blockchain has also been proposed as a solution for the scalability problem. Sharding involves different nodes handling different subsets of the blockchain. ELASTICO [36] proposed a sharding algorithm that works in the presence of Byzantine failures. Sharding increases the throughput of the network linearly with respect to the computational power of the network. However, sharding reduces the security provided by the system making it susceptible to attacks as it reduces the number of attackers required to introduce compromised data into the blockchain network. Omniledger [28] proposes solutions to maintain security in a sharded blockchain. Recent solutions include using inspector nodes [7] to reshuffle validator nodes when required to reduce reshuffling overhead. Polyshard [31] claims to introduce scalability in terms of security, storage efficiency, and throughput by using a ‘polynomially coded sharding’ scheme.

Another area of potential future research seems to be the design of consensus algorithms. A change in consensus algorithms can also result in scalable blockchains. Reference [58] details how expensive consensus mechanisms are not needed in permissioned systems, thereby allowing usage of consensus mechanisms that are known to be scalable in terms of performance. Vukolić in [60] has compared proof of work and BFT protocols with respect to scalability. Many blockchain solutions use Byzantine fault-tolerant [29] protocols to construct scalable blockchains. However, such blockchains are not scalable with respect to the number of nodes because of the large number of messages exchanged between nodes. There has been some work done to reduce communication overheads in BFT. Many BFT protocols modeled after practical Byzantine fault tolerance [6] have been used as consensus protocols in blockchain. Stellar Consensus Protocol [38] introduces Federated Byzantine Agreement (FBA), removing the need for nodes to presuppose a unanimously accepted membership list. Algorand [19] proposes a novel Byzantine Agreement (BA) protocol to reach consensus among users. The core of Algorand uses a protocol called BA* that scales to many users, offers reduced latency, and ensures strict safety rule by making sure there are no forks in the blockchain. Ouroboros [27] is a proof of stake-based consensus algorithm where the authors have proved that honest behavior is a Nash equilibrium, thus proving that attacks are quickly neutralized.

Combination of blockchain with IoT, as suggested by our search results, is being considered as one of the most lucrative fields to work. Reference [9] has summarized the usability of blockchain in IoT. Research on blockchain with IoT parallels research on decentralized smart energy grids. There have been some solutions to enable peer-to-peer energy trading among devices in an industrial IoT setup. Reference [32] exploits a consortium blockchain to provide a secure energy trading mechanism. Energy markets where trading of locally produced renewable energy can take place without interference by an intermediary have been proposed and tested [39]. Blockchains are being used for communication between smart home devices [13]. An appropriate combination of blockchain, multi-signatures, and anonymous encrypted message propagation schemes can be used to build a decentralized smart energy grid system that provides increased security and privacy in comparison with centralized systems [3].

5 Conclusions

This paper aims to provide an overall idea of the domains where blockchain has been used to resolve existing issues or provide new innovative solutions. Through our screening process, we selected a set of 604 primary papers to conduct our mapping study. We have provided a broad classification of the areas under which work done can be classified. We have included statistical data regarding the number of papers published in each category, type of publication (Journal, Conference), and country of origin of the research. We included the year-wise distribution of various domains of blockchain research to better understand the change in research over the years. To further understand the research, we selected a few popular papers under the significant classifications and provided a gist about the type of work being carried out in the domain. We attempted to answer all the research questions that have been formulated. We have documented the entire process and published our results online for verification [51].

The frequency at which papers are being published is very high. The same search conducted a few weeks after the publication of this paper may lead to different results than what we have obtained. This is an inevitable drawback. However, we believe that this study will give researchers, both experienced and new, an idea about the work done so far.

References

1. Abdellatif T, Brousmiche K (2018) Formal verification of smart contracts based on users and blockchain behaviors models. In: 2018 9th IFIP international conference on new technologies, mobility and security (NTMS). IEEE, pp 1–5
2. Ahmed AS, Aura T (2018) Turning trust around: smart contract-assisted public key infrastructure. In: 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE). IEEE, pp 104–111
3. Aitzhan NZ, Svetinovic D (2018) Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Trans Depend Secur Comput 15(5):840–852
4. Alharby M, van Moorsel A (2017) Blockchain-based smart contracts: a systematic mapping study. arXiv preprint arXiv:1710.06372
5. Bragagnolo S, Rocha H, Denker M, Ducasse S (2018) Ethereum query language. In: Proceedings of the 1st international workshop on emerging trends in software engineering for blockchain. ACM, pp 1–8
6. Castro M, Liskov B et al (1999) Practical byzantine fault tolerance. OSDI 99:173–186
7. Chauhan A, Malviya OM, Verma M, Mor TS (2018) Blockchain and scalability. In: 2018 IEEE international conference on software quality, reliability and security companion (QRS-C). IEEE, pp 122–128
8. Chen J, Yao S, Yuan Q, He K, Ji S, Du R (2018) Certchain: public and efficient certificate audit based on blockchain for tls connections. In: IEEE INFOCOM 2018-IEEE conference on computer communications. IEEE, pp 2060–2068
9. Christidis K, Devetsikiotis M (2016) Blockchains and smart contracts for the internet of things. IEEE Access 4:2292–2303

10. Coblenz M (2017) Obsidian: a safer blockchain programming language. In: Proceedings of the 39th international conference on software engineering companion. IEEE Press, pp 97–99
11. Chris D (2017) Introducing ethereum and solidity. Springer, Berlin
12. Dorri A, Kanhere SS, Jurdak R (2017) Towards an optimized blockchain for iot. In: Proceedings of the second international conference on internet-of-things design and implementation. ACM, pp 173–178
13. Dorri A, Kanhere SS, Jurdak R, Gauravaram P (2017) Blockchain for iot security and privacy: the case study of a smart home. In: 2017 IEEE international conference on pervasive computing and communications workshops (PerCom Workshops). IEEE, pp 618–623
14. Duan J, Karve A, Sreedhar V, Zeng S (2018) Service management of blockchain networks. In: 2018 IEEE 11th international conference on cloud computing (CLOUD). IEEE, pp 310–317
15. Duan Z, Mao H, Chen Z, Bai X, Hu K, Talpin J-P (2018) Formal modeling and verification of blockchain system. In: Proceedings of the 10th international conference on computer modeling and simulation. ACM, pp 231–235
16. Ehrlich S (2018) Making sense of china's grand blockchain strategy. https://www.forbes.com/sites/stevenehrlich/2018/09/17/making-sense-of-chinas-grand-blockchain-strategy/#75772ce23678
17. Eyal I, Gencer AE, Sirer EG, Van Renesse R (2016) Bitcoin-ng: a scalable blockchain protocol. In: NSDI, pp 45–59
18. Gallo P, Pongnumkul S, Nguyen UQ (2018) Blocksee: blockchain for iot video surveillance in smart cities. In: 2018 IEEE international conference on environment and electrical engineering and 2018 IEEE industrial and commercial power systems Europe (EEEIC/I&CPS Europe). IEEE, pp 1–6
19. Gilad Y, Hemo R, Micali S, Vlachos G, Zeldovich N (2017) Algorand: scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th symposium on operating systems principles. ACM, pp 51–68
20. Graf R, King R (2018) Neural network and blockchain based technique for cyber threat intelligence and situational awareness. In: 2018 10th international conference on cyber conflict (CyCon). IEEE, pp 409–426
21. Grech N, Kong M, Jurisevic A, Brent L, Scholz B, Smaragdakis Y (2018) Madmax: surviving out-of-gas conditions in ethereum smart contracts. In: Proceedings of the ACM on programming languages 2(OOPSLA):116
22. Hardjono T, Smith N (2016) Cloud-based commissioning of constrained devices using permissioned blockchains. In: Proceedings of the 2nd ACM international workshop on IoT privacy, trust, and security. ACM, pp 29–36
23. Huang X, Xu C, Wang P, Liu H (2018) Lnsc: a security model for electric vehicle and charging pile management based on blockchain ecosystem. IEEE Access PP(99):1
24. Huang X, Zhang Y, Li D, Han L (2019) An optimal scheduling algorithm for hybrid ev charging scenario using consortium blockchains. Fut Gener Comput Syst 91:555–562
25. Jan Z, Third a, Ibanez L-D, Bachler M, Simperl E, Domingue J (2018) Sciencemiles: digital currency for researchers. In: Companion of the web conference 2018. International World Wide Web Conferences Steering Committee, pp 1183–1186
26. Khaqqi KN, Sikorski JJ, Hadinoto K, Kraft M (2018) Incorporating seller/buyer reputation-based system in blockchain-enabled emission trading application. Appl Energy 209:8–19
27. Kiayias A, Russell A, David B, Oliynykov R (2017) Ouroboros: a provably secure proof-of-stake blockchain protocol. Annual international cryptology conference. Springer, Berlin, pp 357–388
28. Kokoris-Kogias E, Jovanovic P, Gasser L, Gailly N, Ford B (2017) Omniledger: s secure, scale-out, decentralized ledger. IACR Cryptol ePrint Arch 2017:406
29. Leslie L, Robert SX, Marshall P (1982) The byzantine generals problem. ACM Trans Program Lang Syst (TOPLAS) 4(3):382–401
30. Lewenberg Y, Sompolinsky Y, Zohar A (2015) Inclusive block chain protocols. International conference on financial cryptography and data security. Springer, Berlin, pp 528–547

31. Li S, Yu M, Avestimehr S, Kannan S, Viswanath P (2018) Polyshard: coded sharding achieves linearly scaling efficiency and security simultaneously. arXiv preprint arXiv:1809.10361
32. Li Z, Kang J, Yu R, Ye D, Deng Q, Zhang Y (2018) Consortium blockchain for secure energy trading in industrial internet of things. IEEE Trans Ind Inform 14(8):3690–3700
33. Liang X, Shetty S, Tosh D, Kamhoua C, Kwiat K, Njilla L (2017) Provchain: a blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: Proceedings of the 17th IEEE/ACM international symposium on cluster, cloud and grid computing. IEEE Press, pp 468–477
34. Liu C, Chai KK, Zhang X, Lau ET, Chen Y (2018) Adaptive blockchain-based electric vehicle participation scheme in smart grid platform. IEEE Access
35. Liu M,Yu FR, Teng Y, Leung VCM, Song M (2018) Joint computation offloading and content caching for wireless blockchain networks. In: IEEE INFOCOM 2018-IEEE conference on computer communications workshops (INFOCOM WKSHPS). IEEE, pp 517–522
36. Luu L, Narayanan V, Zheng C, Baweja K, Gilbert S, Saxena P (2016) A secure sharding protocol for open blockchains. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. ACM, pp 17–30
37. Martens D, Maalej W (2018) Review chain: untampered product reviews on the blockchain. arXiv preprint arXiv:1803.01661
38. Mazieres D (2015) The stellar consensus protocol: a federated model for internet-level consensus. Stellar Development Foundation
39. Mengelkamp E, Notheisen B, Beer C, Dauer D, Weinhardt C (2018) A blockchain-based smart grid: towards sustainable local energy markets. Comput Sci Res Develop 33(1–2):207–214
40. Milutinovic M, He W, Wu H, Kanwal M (2016) Proof of luck: an efficient blockchain consensus protocol. In: Proceedings of the 1st workshop on system software for trusted execution. ACM, p 2
41. Münsing E, Mather J, Moura S (2017) Blockchains for decentralized optimization of energy resources in microgrid networks. In: 2017 IEEE conference on control technology and applications (CCTA). IEEE, pp 2164–2171
42. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system
43. Niya SR, Jha SS, Bocek T, Stiller B (2018) Design and implementation of an automated and decentralized pollution monitoring system with blockchains, smart contracts, and lorawan. In: NOMS 2018-2018 IEEE/IFIP network operations and management symposium. IEEE, pp 1–4
44. Novo O (2018) Blockchain meets iot: an architecture for scalable access management in iot. IEEE Internet Things J
45. O'Connor R (2017) Simplicity: a new language for blockchains. In: Proceedings of the 2017 workshop on programming languages and analysis for security. ACM, pp 107–120
46. Petersen K, Feldt R, Mujtaba S, Mattsson M (2008) Systematic mapping studies in software engineering. EASE 8:68–77
47. Poon J, Dryja T (2016) The bitcoin lightning network: scalable off-chain instant payments. See https://lightning.network/lightning-network-paper.pdf
48. Pustisek M, Kos A, Sedlar U (2016) Blockchain based autonomous selection of electric vehicle charging station. In: 2016 international conference on identification, information and knowledge in the internet of things (IIKI). IEEE, pp 217–222
49. Qin B, Huang J, Wang Q, Luo X, Liang B, Shi W (2017) A decentralized pki mitigating mitm attacks. Fut Gener Comput Syst
50. Ryskeldiev B, Ochiai Y, Cohen M, Herder J (2018) Distributed metaverse: creating decentralized blockchain-based model for peer-to-peer sharing of virtual spaces for mixed reality applications. In: Proceedings of the 9th augmented human international conference. ACM, p 39
51. Samvid Sagar A (2018) Systematic mapping study on blockchain research. https://goo.gl/xSUC1i
52. Sarda P, Chowdhury MJM, Colman A, Kabir MA, Han J (2018) Blockchain for fraud prevention: a work-history fraud prevention system. In: 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE). IEEE, pp 1858–1863

53. Sato T, Himura Y (2018) Smart-contract based system operations for permissioned blockchain. In: 2018 9th IFIP international conference on new technologies, mobility and security (NTMS). IEEE, pp 1–6
54. Shi N (2016) A new proof-of-work mechanism for bitcoin. Financ Innov 2(1):31
55. Shih D-H, Shih P-Y, Wu T-W (2018) An infrastructure of multi-pollutant air quality deterioration early warning system in spark platform. In: 2018 IEEE 3rd international conference on cloud computing and big data analysis (ICCCBDA). IEEE, pp 648–652
56. Sompolinsky Y, Zohar A (2015) Secure high-rate transaction processing in bitcoin. International conference on financial cryptography and data security. Springer, Berlin, pp 507–527
57. Stanciu A (2017) Blockchain based distributed control system for edge computing. In: 2017 21st international conference on control systems and computer science (CSCS). IEEE, pp 667–671
58. Swanson T (2015) Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems. Report, available online, Apr 2015
59. Tewari H, Hughes A, Weber S, Barry T (2017) X509cloud framework for a ubiquitous pki. In: Military communications conference (MILCOM 2017). IEEE, pp 225–230
60. Vukolić M (2015) The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In: International workshop on open problems in network security. Springer, Berlin, pp 112–125
61. Wang W, Niyato D, Wang P, Leshem A (2018) Decentralized caching for content delivery based on blockchain: A game theoretic perspective. arXiv preprint arXiv:1801.07604
62. Wang Y, Alexander K (2018) Designing confidentiality-preserving blockchain-based transaction processing systems. Int J Account Inform Syst 30:1–18
63. Wood G (2014) Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper 151:1–32
64. Xiong Z, Feng S, Niyato D, Wang P, Han Z (2018) Optimal pricing-based edge computing resource management in mobile blockchain. In: 2018 IEEE international conference on communications (ICC). IEEE, pp 1–6
65. Yakubov A, Shbair W, Wallbom A, Sanda D et al (2018) A blockchain-based pki management framework. In: The first IEEE/IFIP international workshop on managing and managed by blockchain (Man2Block) colocated with IEEE/IFIP NOMS 2018, Tapei, Tawain, 23–27 April 2018
66. Yli-Huumo J, Ko D, Choi S, Park S, Smolander K (2016) Where is current research on blockchain technology? AS systematic review. PloS One 11(10):e0163477
67. Zhang T, Pota H, Chu C-C, Gadh R (2018) Real-time renewable energy incentive system for electric vehicles using prioritization and cryptocurrency. Appl Energy 226:582–594
68. Zhen Y, Yue M, Chen Z, Tang C, Chen X (2017) Zero-determinant strategy for the algorithm optimize of blockchain pow consensus. In: 2017 36th Chinese control conference (CCC). IEEE, pp 1441-1446
69. Zyskind G, Nathan O et al (2015) Decentralizing privacy: using blockchain to protect personal data. In: Security and privacy workshops (SPW). IEEE, pp 180–184

Blockchain-Based Secure E-Voting with the Assistance of Smart Contract

Kazi Sadia, Md. Masuduzzaman, Rajib Kumar Paul, and Anik Islam

Abstract Voting is a very important issue that can be beneficial in terms of choosing the right leader in an election. A good leader can bring prosperity to a country and also can lead the country in the right direction every time. However, elections are surrounded by ballot forgery, coercion, and multiple voting issues. Moreover, while giving votes, a person has to wait in a long queue and it is a very time-consuming process. Blockchain is a distributed database in which data are shared with the participant of the node and each participant holds the same copy of the data. Blockchain has properties like transparency, pseudonymity, and data integrity. In this paper, a fully decentralized e-voting system based on blockchain technology is proposed. This protocol utilizes smart contracts in the e-voting system to deal with security issues, accuracy, and voters' privacy during the vote. The protocol results in a transparent, non-editable, and independently verifiable procedure. The protocol discards all the intended fraudulent activities occurring during the election process by removing the least participation of the third party. Both transparency and coercion are obtained at the same time.

Keywords Blockchain · E-voting · Hash · Security · Smart contract

1 Introduction

1.1 Blockchain

Blockchain is essentially a distributed database of records or a public ledger of all transactions or digital events that have been occurred and shared among participating

K. Sadia · R. K. Paul
Department of Computer Science and Engineering, American International University-Bangladesh, Dhaka, Bangladesh

Md. Masuduzzaman · A. Islam (✉)
Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, South Korea
e-mail: anik.islam@kumoh.ac.kr

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_14

parties connected within a network [1]. A blockchain is a chain of blocks where blocks are connected to hold data or information regarding any event [2]. Each transaction or activity within the blockchain is verified by consensus of a majority of the participants (i.e., without the approval of the majority network, no activity is acceptable) [3]. Once some data have been inserted into a blockchain, it becomes very difficult to change it due to having an immutability configuration [4]. To rewrite any data, dishonest miners must rewrite the previously broadcasted block and these changes have to be agreed by the other miners in the network [1].

In the blockchain, double spending is prevented by using "proof of work" that requires computer processing power to generate fingerprints to uniquely identify each block [1]. Blockchain technology uses cryptography which ensures the legitimacy of a transaction [5]. Third-party involvement is prevented by the peer-to-peer network validation. Therefore, cost and trust-related issues are resolved [6]. The structure of a block in the blockchain is described below.

Data—The data can be any type of information that is stored in the block.
Hash—The hash is a kind of fingerprint that uniquely identifies a block and is generated based on its contents.
Hash of the previous block—It refers the previous block to form the chain. Any change in data can change the hash of the block.

According to Fig. 1, when a participant intends to add a block to the chain, the peer nodes are responsible for validating the block. After the verification, if the majority agrees to add the block, then the block is added to the blockchain [7]. If the majority denies, then block is discarded.

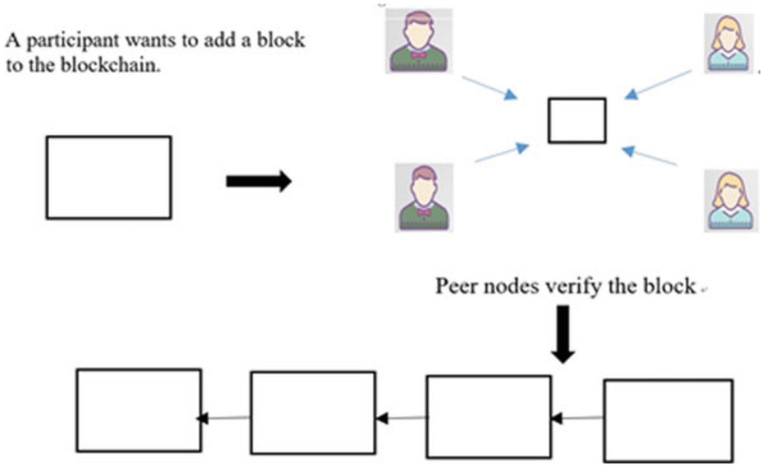

A participant wants to add a block to the blockchain.
Peer nodes verify the block

Fig. 1 Mechanisms of adding a block in blockchain

1.2 E-Voting

In democracy, the main important thing is to secure the election process for the national security and development of a nation. Ever since the candidates were needed to be elected through a democratic process, it was done by voting with pen and paper. Afterward, the result was counted manually and declared. The process of voting with paper ballot and pen required a lot of time and created hustle in maintaining a long queue. Also, the manual process ensues with ballot forgery, coercion, and multiple voting. Now, replacing the traditional process of voting by a new innovative process might be condemned in stopping any sort of duplicity and forgery [8].

E-voting is the new concept proposed to ensure fair and digitalized voting that promises to resolve all the issues related to the traditional voting process. By electronic voting, we generally mean the vote casting process with the help of any sort of computer or computerized voting equipment or the Internet. The tasks are conducted through systems to hereby reduce the involvement of manpower during the election process. Registering the voters, tally ballots, and recording of votes can also be easily done by this electronic system [9].

Electronic voting machine is neither a complex machine nor a harder one to operate. It can be easily understood and operated by both the election officer's in-charge and the voters. EVM has basically three units—control unit, display unit, and ballot unit. The main unit of EVM is a control unit that stores all the data and controls the basic function including voter information. Vote counting is assuredly conducted with possibly less time and accuracy.

1.3 Smart Contract

Previously, contracts between parties were held upon visual meetings. The smart contract is aimed to provide contracts between parties where both parties are given the priority and contracts are conducted upon establishing the conditions of both parties [4]. It is the executable code that runs on top of the blockchain to facilitate the terms required in an agreement of a contract between the two parties. The involvement of any third party is resolved as any medium between parties is not required as contracts are self-executed.

A smart contract is a legal application that runs on a blockchain network [4]. Smart contracts are much like legal contracts. The smart contract can be used in many different things. Banks, for example, could use it to issue a loan, worth for automatic payments, both e-commerce and music rights management can use this. An insurance company could use it for process claims; postal companies use it for payment on delivery and so on.

- No trust issue in a smart contract just like this vendor machine, as shown in Fig. 2. A person itself can put the coin into this and get the desired product.

Fig. 2 Example of smart contract

- No involvement of the third party: the same as this vendor machine. When a person itself involved with this matter can directly interact with it and get the desired product. Moreover, there is not any involvement of third party.
- As the smart contract is distributed in an open ledger, there is no chance of losing or hacking as in an open environment. It is difficult to involve in and manage to steal stuff.

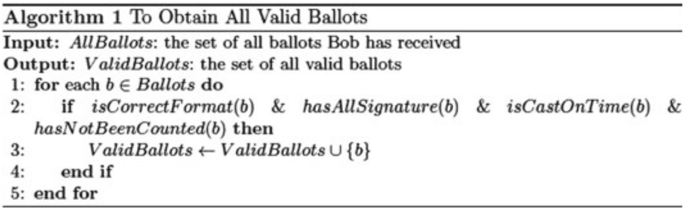
The remaining sections of this paper are organized as follows: Sect. 2 represents related works. In Sect. 3, the proposed scheme is depicted. A security analysis based on different properties is outlined in Sect. 4. Finally, Sect. 5 draws a conclusion from this paper.

2 Related Work

There has been a lot of work on blockchain-based e-voting using cryptography, signatures, and other techniques. In such papers, minimal involvement of the third party observed is significantly less and a problem of coercion and transparency maintenance at the same time is also observed. Additionally, the balancing of transparency and coercion resistance was a possible future work in [10]. Reduction of third party is a major portion of work in an election process as the impact of third-party involvement can have a vulnerable effect on the whole procedure. Moreover, coercion resistance is a difficult task that is to be mapped with transparency.

Lewis et al. [1] described blockchain as an open, distributed ledger of historical records that uses cryptography and digital signatures. In his paper, he also mentioned the logic of blockchain and how does it work. Upon explaining the aftermath of resolving conflicts, he introduced an idea of not broadcasting a block intentionally. Two blocks can be created, and one can be left as being not broadcasted. The unbroadcasted block can be broadcasted when desired. In this paper, we have used this concept to keep the choices of nominees secured until result calculation.

Liu et al. [10] proposed a protocol where the choice was made safe using a random string and choice code. The length of the vote string varies depending on the election requirements. The choice code represents the voter's choice followed by a random string which is an indication of a well-formed vote. According to Liu et al. [10], the phases are pre-voting phase, voting phase, post-voting phase. In the pre-voting phase, the organizer Bob collected all valid ballots. After ending the voting time, Bob generates a set of all ballots which means all the ballots that have been received. Then Bob runs this algorithm 1:



Algorithm 1 To Obtain All Valid Ballots

```
Input: $AllBallots$: the set of all ballots Bob has received
Output: $ValidBallots$: the set of all valid ballots
for each $b \in Ballots$ do
    if $isCorrectFormat(b)$ & $hasAllSignature(b)$ & $isCastOnTime(b)$ & $hasNotBeenCounted(b)$ then
        $ValidBallots \leftarrow ValidBallots \cup \{b\}$
    end if
end for
```

This algorithm runs to gain a set of valid ballots which is set of all the valid ballots.

There are issues regarding an election. Therefore, voters' privacy must be assured. Thus, the concept of public and private keys is used in different papers but with a little modification. Anonymity was ensured by keeping voters' identity private [10, 11]. According to Liu et al. [10] and Hardwick et al. [11], one must authenticate oneself to the central authority (CA) and CA receives a token that proves one's eligibility to vote. In these papers, one central authority or an officer is responsible for initial verification.

The counting phase described in the protocol discussed by Hardwick et al. [11]. Hardwick et al. [11] deal with broadcasting a ballot opening message that contains a value which will represent the voter's choice and the voter's themselves broadcast this. Hardwick et al. [11] stored the information of the list of candidates and voters in the genesis block as the initial storage. The authors revealed the result at the end of the election using the concept of value representation of the voter choice. A voter can vote multiple times, and every time the previous vote was replaced by the current one. By this process, coercion is said to be totally removed. In both the papers, everyone can view the public blockchain and there is no centralized authority. In [12–17], they also proposed a voting mechanism which utilizes blockchain.

3 Proposed Methodology

3.1 Procedure

The basic functionalities of the proposed protocol are shown in Fig. 3. The code is executed on top of the blockchain. Therefore, verifying actions that were supposed to be performed by the third party are performed automatically. Moreover, the peer network connected is in-charge of further verification as mentioned. The figure introduces some unknown terms that are further described below.

Condition 1—Verify whether the voter is in group X and the flag of X is true. Also, check whether the voter is on the eligibility list or not.

Condition 2—Mathematical computation (proof of work) is done. Also, verify whether the voter has cast vote previously or not and check the ballot is in the correct format or not.

Organizer—In this protocol, the organizer is the only representative who is involved within the protocol but for a limited time. The role of the organizer is to arrange and collect the list of nominees, list of eligible voters, start date and time, end date, and time. The start and end (date and time) are decided and announced by the election commission. The list of eligible voters is collected through manual registration.

Ballot string—The string that contains the choice of nominee hidden around random numbers to avoid recognition.

Sibling block—A block that contains the arrangement of choice value.

st—ST—Start Time
et—ET—End Time

Hash (fingerprint)—a hash function that used on the binary value of the voter's fingerprint.

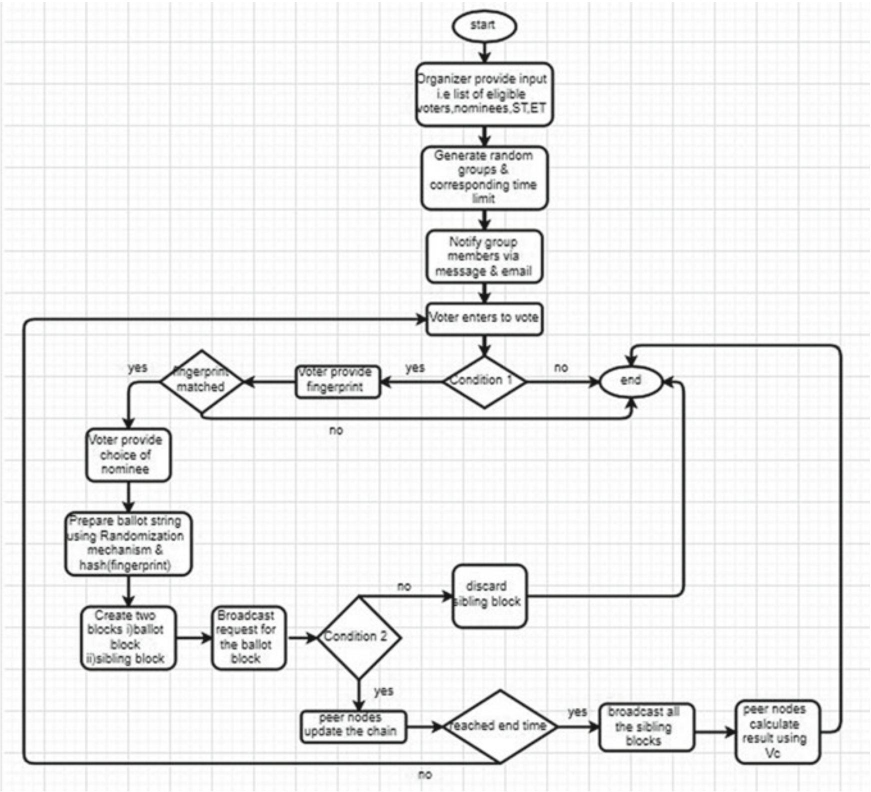

start
Organizer provide input i.e list of eligible voters,nominees,ST,ET
Generate random groups & corresponding time limit
Notify group members via message & email
Voter enters to vote
Condition 1
yes
no
end
Voter provide fingerprint
fingerprint matched
yes
no
Voter provide choice of nominee
Prepare ballot string using Randomization mechanism & hash(fingerprint)
Create two blocks i)ballot block ii)sibling block
Broadcast request for the ballot block
Condition 2
no
discard sibling block
yes
peer nodes update the chain
reached end time
yes
broadcast all the sibling blocks
peer nodes calculate result using Vc
no

Fig. 3 Flowchart representation of the basic functionalities of the protocol

Note: The choice of nominee is hidden in the ballot string. The arrangement of the choice is hidden in the variable V_c. The arrangement is prepared by random number generation. Thus, nobody has any idea of the voter's choice until the end of the election.

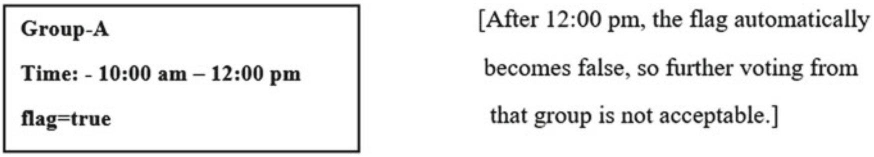
3.2 The Phases of the Proposed Protocol

The protocol is categorized into three phases, in which each phase is dependent upon another. Following are the three phases:

1. Pre-voting phase.
2. Voting phase.
3. Post-voting phase.

1. Pre-voting phase

The organizer is responsible for collecting the list of the eligible voters and nominees based on the desired condition (if any). The list of the voters should contain voters' names, national identification numbers (NID), fingerprint, and any other information based on the direction of the election commission. Organizer provides the list of eligible voters, and their fingerprint coordinates along with the binary value, nominees, start date–time, and end date–time as an input on the genesis block. In the case of people having a problem, an alternative option is considered. A priority list is maintained. In the priority list, the thumb is given the priority and people deprived of thumb can use the grooming finger. For worst case, message verification process can be used. In that process, a pin code is sent to the particular contact number of the voter and the voter has to provide the pin to verify himself as an alternative of the fingerprint. Genesis block is the parent block or the first block of the blockchain. The start date–time and end date–time are mentioned earlier by the election commission. The role of the organizer ends here; as per the result of the code execution, the procedure is carried out. The program (code) is previously integrated within the blockchain as per the concept of smart contract. On reaching the start date–time, one of the pre-defined conditions fulfills (i.e., {if (DateTime.Now==st) start ();}; a function is called which invokes the election procedure to start and corresponding activities are performed). Voters are grouped randomly based on the number of eligible voters. Moreover, other conditions are also provided and random time is generated for each group. Each group holds distinct timing; overlapping is not taken into consideration. Voters of specific groups are notified via email and message; a time limit is set for each group.



Group-A

Time: - 10:00 am – 12:00 pm

flag=true

[After 12:00 pm, the flag automatically becomes false, so further voting from that group is not acceptable.]

The flag is a Boolean property of a group. The flag remains true until the time limit of the specified group expires. The duration of each group is also decided by the election authority. The voting duration for each group must be adjusted in such a way that none of the voters skip to vote due to load/traffic on the network. No one is allowed to vote after the flag becomes false (i.e., the time limit exceeds). The flag becomes false automatically once all the voters within the group are done with their voting which provides further security.

2. Voting phase

As the voter approaches to voting providing his/her public keys, it is verified (within the code) whether the voter is in the group with a flag value of true and whether the voter is in the eligibility list. As a smart contract performs an executable code, it is verified through the code by the call of a function that checks whether the voter

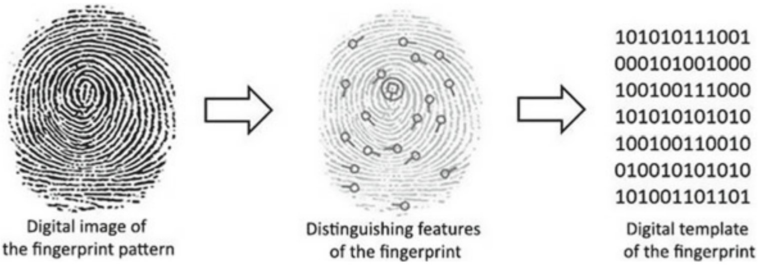
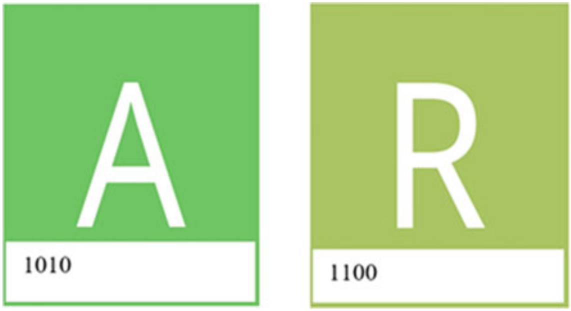

101010111001
000101001000
100100111000
101010101010
100100110010
010010101010
101001101101
Digital image of the fingerprint pattern
Distinguishing features of the fingerprint
Digital template of the fingerprint

Fig. 4 Conversion of the fingerprint pattern to the binary value

entered is eligible or not. Given that, the eligibility lists of the voters are stored on the genesis block. As the voter has proved him/her as eligible, and also, the voter is in the specified group (the group to serve currently); the voter is then to provide his/her private key (fingerprint which is converted to binary data, as shown in Fig. 4); as a need of verification that no other people except the voter is casting his/her vote. This reduces the chance of anyone knowing one's public keys and using the public key to cast vote in the name of the voter. This is the second phase of verification of voters. The fingerprint is matched with the one provided along with the eligibility list in the genesis block. The fingerprint sensor is used to figure out the coordinates of particular voters. The coordinates are then matched with the coordinates provided in the genesis block. If it matches then, according to Fig. 4, the binary value of the coordinates is obtained from the provided list in the genesis block. Conversion of the coordinates into the binary value during the voting process will require time and memory consumption. Thus, this procedure is performed. The hash of the binary value is the unique voter identification in the ballot within the block. Direct voter's identity is avoided to ensure the voter's security. The hash (fingerprint$_{\text{binary}}$) value is the representation of the voter in the block. SHA-256 is used as the secured hash function, hash (fingerprint$_{\text{binary}}$) that cannot be reversed. According to some research, fingerprint is one of the most secure metadata of a person. Thus, fingerprint is used instead of any other metadata in this protocol.

The voter is then provided with the list of nominees each represented by a logo. The voter then selects his/her choice of nominee. The nominees are represented by their representative logo. The logos have a binary value which is basically selected and worked with when chosen. The calculations and workings are done upon distinct binary values. Figure 5 shows an example of the representation. The number of 1's and 0's in representing the nominees must be the same. Otherwise, it is possible to guess the choice of nominee in the ballot string. Upon several workings, it has been seen that an unequal number of 0's and 1's for every nominee may result in the prediction of the selection of a nominee. As a result, the progress of the election is made visible. If the representations do not remain consistent or if it is not possible to allocate different representations of nominee within (N) bits, then increase the number of bits to get different representations for equal numbers of 0's and 1's. For example, three bits with two 1's and a 0 will have representations—110,011,101.

Fig. 5 Binary representation of nominee logos



Therefore, three logos can be represented by these in other words three nominees can be represented.

On choosing the nominee, the preparation of the ballot takes place. A ballot is designed to have a ballot number in it. The ballot consists of the voters' hash ($\text{fingerprint}_{\text{binary}}$) and the ballot string. The ballot string is prepared by the execution of a function inside the code with the concept of smart contract. The ballot string must be different for every voter. The ballot string has two substrings, such as choice string and the random string. The choice string consists of the nominee choice hidden within other randomly generated values. The random string is randomly generated 0/1 values. These techniques are used to prevent viewers from recognizing the choice of nominee. A nominee might get multiple votes. Therefore, to distinguish every ballot strings the concept of random string is used. Generation of the random string results in unique ballot string formation. The ballot string is prepared in two phases, and the following are:

Note: The total number of bits has no restriction. 16-bit is just an example. Greater number of bits is more secure as chances of similar generation of random number decrease. The decision of the number of bits must be taken into consideration before making the decision.

Let us consider a 16-bit ballot string of which 8 bits are choice string (i.e., the red ones) and 8 bits are random string (i.e., the black ones). The ballot string is equally divided into these two parts.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

(i) If n bits are representing each logo, then n random numbers are generated from 0 to 7 as the choice string is between 0 and 7. The binary value of the logo is arranged in the generated random value indexes of the ballot string (i.e., Alice chooses the nominee with a binary value of 1100 and the binary value consists of four bits. Thus, four random numbers are generated to hide the choice of Alice).

Number of bits is representing each logo—4
Random numbers—4,5,7,0 (4)—V_c—opening value

Nominee choice—a binary value of logo—1100

Therefore, the four randomly generated numbers—4,5,7,0 are the indexes to hide the binary value of the nominee's choice. The value is assigned sequentially.

0				1	1		0								
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

(ii) Generate another number between 1 and 0. Fill that number in the other four indexes. Example-1

0	1	1	1	1	1	1	1	0							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

The other indexes of the choice string are assigned with either 1 or 0. However, all the other indexes of the choice string must have the same value to avoid recognition of the choice.

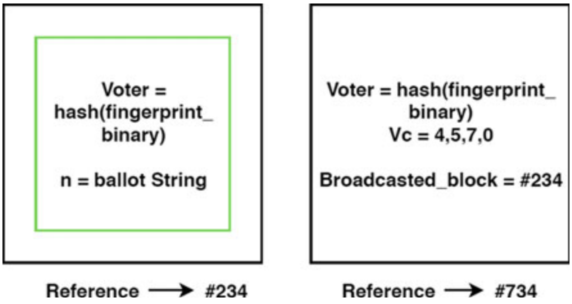
(iii) Generate random numbers randomly between 1 and 0 and put on the indexes (8–15) suppose—11001010

0	1	1	1	1	1	1	1	0	1	1	0	0	1	0	1
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

As 8–15 is the random string part of the ballot string, eight random numbers either 1 or 0 are generated and assigned sequentially to distinguish each ballot string. The ballot string is prepared, and the choice is hidden inside the string. The choice is recognized by the V_c only. Dispose of the V_c can only be result in the consideration of the vote. One block is created containing the ballot, and another sibling block is created that consists of the voters' hash (fingerprint$_{\text{binary}}$), the reference number of the broadcasted block, its own reference number, and the opening value of the choice (i.e., in this case—4, 5, 7, 0). Figure 6 shows the arrangement.

As the voter casts the vote (i.e., the voter broadcasts the ballot containing block), the ballot containing block is requested to add in the chain, whereas the sibling block remains un-broadcasted. The peer nodes start to work for the proof of work for the block. The one, who complete the puzzle, verifies whether the voter has cast vote earlier and whether the ballot is in a correct format. After all the verification, the ballot contained block is added in the blockchain and other peer nodes verify and update their chain. Majority is taken into consideration. If majority disagrees, then the block is discarded.

Fig. 6 Blocks in the blockchain



3. Post-voting phase

Once the ending time is reached, it is checked whether all the voters have voted or not. If not, then they are shortly given a notification to complete their voting within the specified time. Their voting is performed similarly as per the voting phase; but if they fail to do so, then no consideration is taken to be granted. If all the voters are done with their voting, then as instructed in the contract, all the sibling blocks are broadcasted one by one sequentially. Once all the sibling blocks are broadcasted, the peer nodes start to calculate the result referencing the blocks and using the V_c to extract the choice of nominee for every block. Here, all the nodes are supposed to come up with the same result as no blocks are discarded unnecessarily in between and the blockchain supports no changes. Therefore, the voters, in other words, the peer nodes themselves count the votes and broadcast the result preventing the need for counting using third party. The blockchain is transparent, and the accuracy is ensured as everything is made visible.

4 Security Analysis

In e-voting, security is the main concern that must be taken into consideration at first because if the voters are not assured of their safety, then they are not going to involve in the protocol. The following are the certain security goals that can be satisfied with our proposed methodology.

4.1 Anonymity

This protocol uses public and private keys of a voter during the process execution. On the blockchain, only the voter's public key is broadcasted which is hashed previously. Therefore, by excluding the voter's actual identity, no one will be able to recognize any voters within the blockchain. The only identity of voters is the hash of fingerprint that is the binary values of the coordinates.

4.2 Voters' Privacy

Voters are not aware of their timing to vote. Therefore, the chances of manipulation and coercion by fraudulent supporters are reduced. The timing of the voters' voting is only kept within the randomly generated time against each group in the code executed. As a result, manipulators or the party-specific public cannot blackmail or threaten voters.

4.3 Confidentiality

Confidentiality is equivalent to part of privacy. The prevention of sensitive information from reaching unauthorized users while making sure that the right people are aware of it. The most common method ensuring confidentiality is a data encryption, and in this protocol, the data are the voter's identity and the ballot string which is encrypted and can only be made visible to all the participants once the election process is over. Being concerned about the voter's identity, only the voter themselves are aware of their identity and choice. The ballot string is prepared in such a way that without the V_c, and the choice is not understandable. Once the sibling block is broadcasted, the choices are visible to all the peer nodes within the network.

4.4 Ballot Manipulation

In this protocol, inappropriate ballots (i.e., one voter voting more than once) is prohibited by the rejection of the approval of the peer nodes. Upon verifying, the peer nodes reject the ballot and are not further added to the block. Ballots without correct format are also discarded, and it is made sure that the sibling block of the rejected block is discarded simultaneously. Ballots are contained in blocks that is why the modification is not possible. A single change in a block leads to changing the other blocks linked with it.

4.5 Transparency

Blockchain is an open and distributed ledger where each transaction and activity is made transparent for peer verification, validation, and visibility. As things are kept visible, this ensures no fraudulent activities to take place secretly. The fairness and accuracy are obtained through the blockchain's property of being transparent.

4.6 Public Verifiability and Individual Verifiability

Our protocol provides the opportunity to publicly verify activities or the voting process as it is kept transparent with the help of blockchain. Peer nodes or anyone can monitor activities of the participant of the network. Moreover, voters themselves can make sure whether their vote is taken into consideration or not. If the block containing voter's identity is broadcasted, then this ensures that the voter's vote is going to be taken into consideration. Individual verifiability is satisfied through the protocol.

4.7 Auditability

Results are calculated after the ending of the election process, and the whole process is auditable as blockchain keeps the record of the whole thing. The rejected blocks and ballots can be monitored at later stages to have an idea of how often fraudulent activities were intended. Smart contract codes cannot be modified since it is permanently written on the blockchain.

4.8 Consistency and Accuracy

All the peer nodes will have the same record, and at the end, the same result will be obtained by all the participants. For every activity, a consensus mechanism is carried out to satisfy the consistency. Meanwhile, no changes are incurred, and the consensus mechanism makes the protocol accurate.

4.9 Non-repudiation

The process of non-repudiation is that someone cannot deny something. Therefore, the result obtained cannot be claimed as being unfair or of fraudulent activities as every activity is made transparent and verifiable by the majority network. It is not possible to mess with the majority honest network. Activities are performed by the execution of code where there is no possibility of unfair means.

5 Conclusions

E-voting is an emerging concept or solution of voting to carry out activities with accuracy and reliability. Moreover, blockchain is an interesting and attractive technology that provides transparency of data and is a topic of high demand. As the process of election must be handled with care to avoid unusual circumstances and occurring, this protocol reduces the constraints of manual voting and other e-voting systems based on blockchain. Also, the reduction of the third party is proof of a healthy election which is enabled with the assistance of a smart contract. The coercion is also prevented by the concept of random generation of groups using a smart contract. The techniques used in the protocol are quite simpler and easily understandable. Moreover, this protocol is designed to reduce memory and time consumption to make tasks faster. Thus, this protocol fulfills all the previously defined properties of the referred paper along with the prevention of coercion with transparency. The voters can monitor the whole process, and their privacy is also maintained to avoid any sort of privacy issues. Moreover, a replacement of the metadata can be taken into consideration to make this protocol widely used in all areas.

References

1. Islam A, Shin SY (2019) BUS: a blockchain-enabled data acquisition scheme with the assistance of UAV swarm in internet of things. IEEE Access 7:103231–103249. https://doi.org/10.1109/ACCESS.2019.2930774
2. Islam A, Shin SY (2018) Blockchain technology in networking: a survey of the state-of-the-art. In: Proceedings of symposium of the Korean Institute of communications and information sciences, pp 321–322, June 2018
3. Islam A, Uddin MB, Kader MF, Shin SY (2018) Blockchain based secure data handover scheme in non-orthogonal multiple access. In: 2018 4th international conference on wireless and telematics (ICWT), Nusa Dua, 2018, pp 1–5. https://doi.org/10.1109/icwt.2018.8527732
4. Islam A, Shin SY (2019) BUAV: a blockchain based secure UAV-assisted data acquisition scheme in internet of things. J Commun Netw 21(5):491–502. https://doi.org/10.1109/JCN.2019.000050
5. Islam A, Chae S, Shin SY (2018) Social Internet of Things (SIoT) and blockchain: research opportunities and challenges. In: Proceedings of symposium of the Korean Institute of communications and information sciences, pp 326–327, January 2018
6. Islam A, Kader MF, Shin SY (2019) BSSSQS: a blockchain-based smart and secured scheme for question sharing in the smart education system. J Inform Commun Converg Eng 17(3):174–184. https://doi.org/10.6109/JICCE.2019.17.3.174
7. Islam A, Shin SY (2019) BHMUS: blockchain based secure outdoor health monitoring scheme using UAV in smart city. In: 2019 7th International Conference on Information and Communication Technology (ICoICT), Kuala Lumpur, Malaysia, 2019, pp 1–6. https://doi.org/10.1109/icoict.2019.8835373
8. Weaver N (2016) Secure the vote today. Lawfare Blog, Washington, D.C.
9. General framework of electronic voting and implementation thereof at national elections in Estonia. Document: IVXV-ÜK-0.99, 12 January 2017
10. Liu Y, Wang Q (2017) An E-voting protocol based on blockchain, October 2017

11. Hardwick FS, Gioulis A, Naeem Akram R, Markantonakis K (2018) E-voting with blockchain: an E-voting protocol with decentralization and voter privacy. arXiv:1805.10258 [cs.CR], 3 July 2018
12. Cranor LF (2001) Electronic voting. Encyclopedia of Computers and Computer History, Fitzroy Dearborn
13. Kumar DA, Begumn TUS (2012) Electronic voting machine—a review. In: International conference on pattern recognition, informatics and medical engineering, March 21–23, 2012
14. Hanifatunnisa R, Rahardjo B (2017) Blockchain based e-voting recording system design. In: 11th International conference on telecommunication systems services and applications (TSSA), 2017
15. Hjalmarsson FP, Hreioarsson GK, Hamdaqa M, Hjalmtysson G (2018) Blockchain-based E-voting system. In: 2018 IEEE 11th international conference on cloud computing (CLOUD), San Francisco, CA, USA, 2018, pp 983–986
16. Hoque MdM (2014) A simplified electronic voting machine system. Int J Adv Sci Technol 62:97–102
17. Yavuz E, Koc AK, Yavuz E, Cabuk UC, Dalkilik G (2018) Towards secure E-voting using ethereum blockchain. In: 6th International Symposium on Digital Forensic and Security (ISDFS) 2018

Transcripts DApp—A Blockchain-Based Solution for Transcript Application

Shrinivas Khedkar, Akhil Powar, Nikhil Powar, Chethan Kille, and Harsh Kansara

Abstract There is a need for a reliable system with an efficient and simplified process for academic transcript application and procurement along with sufficient measures for authentication of issued transcripts and verification of the integrity of transcripts. The work presented in this paper proposes to create a blockchain-based decentralized application which can be used by students for application of transcripts, which can only be approved and issued by the intended institute, and by universities for the verification of the transcripts issued. The proposed system attempts to leverage the immutability and transparency of blockchain and versatility of programming provided by smart contracts to ensure that there is no avenue for alteration or forgery of transcripts.

Keywords Blockchain · Transcripts · DApp

1 Introduction

The current infrastructure for transcripts generation and validation is largely based on physical processes and paperwork. This often requires the presence of student or some representative to physically deliver required documents and/or collect the issued copies of transcripts. There is extra work required by the university receiving the

S. Khedkar · A. Powar · N. Powar (✉) · C. Kille · H. Kansara
Veermata Jijabai Technological Institute, Matunga, Mumbai 400019, India
e-mail: napowar_b15@ce.vjti.ac.in

S. Khedkar
e-mail: sakhedkar@ce.vjti.ac.in

A. Powar
e-mail: aapowar_b15@ce.vjti.ac.in

C. Kille
e-mail: cpkille_b15@ce.vjti.ac.in

H. Kansara
e-mail: hjkansara_b15@ce.vjti.ac.in

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_15

student's transcripts to verify their authenticity. Also there is an added disadvantage of unregulated cost and waiting time to get the transcripts from universities.

There are a few online systems related to application and verification of transcripts. However, many of these are only partly online, and the transcripts obtained are actually physical documents. Completely digitized systems that use PKI place dependency on a third-party. The few systems that do use blockchain are not truly decentralized. They depend on centralized storage, issuers, or centralized credential stores. Opting for storage on the blockchain itself sharply increases the costs associated with any blockchain-based system. Due to absence of a universal log-in/sign-up method, identities on the blockchain-based applications are tied to a particular blockchain address, which greatly reduces mobility for users as they are constrained to keep utilizing the same blockchain address for all transactions. It also leads to a dependency on providers (such as MetaMask, Mist, etc.) to function.

The application of blockchain technology to tweak the current system will ensure transparency and avoid fraudulent information as the system is consensus based and the ledger is public and immutable. A blockchain-based solution will ensure reduced paperwork and hence reduced cost and waiting times for the applicants. Third-party validation will also be eliminated and a trust-less system can be generated. The DApp is also designed to provide true mobility to users by use of identity contracts.

In the rest of this paper, we provide backgrounds of concepts that enable the proposed system, review previous work in related areas of application, provide an overview of the proposed system design, and its working and then lay out possible avenues for future works based on or related to the proposed system.

2 Blockchain

The blockchain technology was formally introduced by Satoshi Nakamoto in their work on Bitcoin [1]. It is an incorruptible digital ledger that records any sort of transaction in a network. Transactions are grouped together and stored as a "block," which is then added to the chain.

Blockchain uses a consensus algorithm to handle addition of new blocks. The algorithm helps the blockchain to ensure a state of consensus and thus maintain the integrity of the blockchain.

Each block in the chain contains the hash of the preceding block. The hash of the new block is calculated by taking into account the transactions included in the block and the hash obtained from the previous block. Thus, any alteration in a block's data will lead to invalidation of that block and all others that had succeeded it. The comparison of hashes thus ensures the validity of the blockchain.

2.1 Smart Contract

A smart contract is a self-executing computer program that is used to perform required tasks when triggered by some predefined conditions [2]. Smart contracts are used to achieve decentralized automation. Smart contracts are publicly visible to those with access to the blockchain.

A smart contract is immutable; in that, once a smart contract is deployed on the blockchain, it cannot be changed. A change in the contract, if absolutely required, must be brought about by creating an entirely new contract.

A smart contract's transactions, including deployment, are recorded in the blockchain similar to any other transaction. Smart contracts may be programmed to carry out any type of general purpose functions.

2.2 Transcripts

A transcript is a record of the courses taken by a student and the grades earned by the student in those courses. These are official documents which can be verified by an authorized entity/person. Generally, universities charge a fee for each copy of the official transcripts. Applicants send their transcripts via post to the admission office of the university in which they seek admission in.

2.3 Decentralized Storage

The blockchain is not meant for data storage as storing large documents will be very expensive. It is a public ledger meant to record transactions which are inexpensive to store as compared to conventional files, owing to their fixed, limited size. Storing files on the blockchain would require us to store them as transactions. Large files would be spread over several transactions, which may span multiple blocks.

Interplanetary File System (IPFS) [3] is a peer-to-peer file-sharing system. It uses a distributed hash table (DHT) so that the data is spread across a network of computers, and efficiently coordinated to enable efficient access and look-up between nodes. The main advantages of DHT are decentralization, fault tolerance, and ability to be scaled. Nodes do not require central coordination; the system can function reliably even when nodes fail or leave the network, and DHT can scale to accommodate millions of nodes. Together these features result in a system that is generally more resilient than client–server structures.

To maintain the integrity, the data blocks are stored as a Merkle DAG. This is done by organizing data blocks using hash functions. It is simply a function that takes an input and calculates a unique alphanumeric string(hash) corresponding with that input. All content on IPFS can be uniquely identified, since each data block has a unique hash. Plus the data is tamper-resistant because to alter it would change the hash.

2.4 Decentralized Application (DApp)

Decentralized application is an application that has its back-end code running on a decentralized peer-to-peer network. A truly decentralized application includes decentralized storage. DApps, by definition, do not have a single point of failure.

3 Related Work

Traditionally, applications involving document transfer and application process have been implemented using public key infrastructure [4]. These methods have employed centralized certificate authorities to authenticate both ends. As such, they are vulnerable to the problems of a trust-based system including third parties [5].

Transcript application using blockchain aims to address all drawbacks faced by earlier methods. Sony Global Education has one such implementation which uses Hyperledger Fabric as the underlying blockchain for managing transcripts and scores [6]. However, all documents must adhere to one of the predefined structures for validity.

Learning Machine, in collaboration with MIT Media Laboratory, has created the Blockcerts tool-set, which is an open infrastructure for creating, issuing, viewing, and verifying blockchain-based certificates [7]. This can be used for purposes beyond education, with professional document sharing and managing capabilities. This tool-set does not allow for changes in the issued certificates. Any errors in the certificate thus result in revocation of the original certificate and issuing of a new one.

Smart Cert is an initiative based on the Blockcerts tool-set, which are digital certificates registered on a blockchain [8]. These certificates are signed using cryptography and can be shared. With Smart Cert, the certificates are not stored in a decentralized storage, only their hashes are. The certificates themselves are stored in a centralized storage.

C Verification's solution is a blockchain-based recruitment and background verification platform [9]. It allows the user to store verified testimonies of their professional achievements and share it with all potential employers. However, C Verification is still a third-party between the issuer and consumer of the references, which is not desirable in our scenario. Also, C Verification charges a fee from its users for every action, retaining 30–100% of it depending on the use case.

BC Diploma provides a platform to certify diplomas by associating Ethereum technology with cryptography [10]. BC Diploma is a DApp that can be used by institutions of higher education to issue their degrees on Ethereum. Although it is fully decentralized, the diplomas are stored on the chain in text format and in fixed sizes, which requires standardization of diplomas. Such standardization does not exist and forcing current diplomas into constraints would result in loss of information.

A paper published by researchers at Southern Taiwan University of Science and Technology [11] describes an application that manages and verifies digital certificates. In this, the authors have proposed a Web-based application to store digital certificates which are then validated using serial numbers issued by the system or using QR codes. The universities themselves fill out student details while uploading the digital certificates, which can then be shared by the student with other universities or with employers.

4 Transcripts DApp

The following sub-sections describe the proposed system design. Figure 1 depicts the architecture of the proposed system.

4.1 Framework of the System

The system goes through the following steps:

1. The admin adds the academic institute to the institute list in the ‘Entity List’ smart contract. This is usually done by the admin after they have performed some sort of authentication and verification of the institute.
2. The student logs into the system using their sub-domains assigned. For first-time log-in, a sub-domain is created and assigned to the student. The student is hereafter represented entirely by their ID contract. Further details about this step can be found in the next section.
3. The student creates a new application, or proceeds directly to step 5. By creating a new application, a new ‘Transcript Application’ contract is created. Further details about this step can be found in the next section.
4. The institute then approves the application of the student. Further details about this step can be found in the next section.
5. The student then views the application status. If application has been approved, student may access the transcript document directly though it or may choose to download it.
6. The student may also share visibility of their transcript with others.

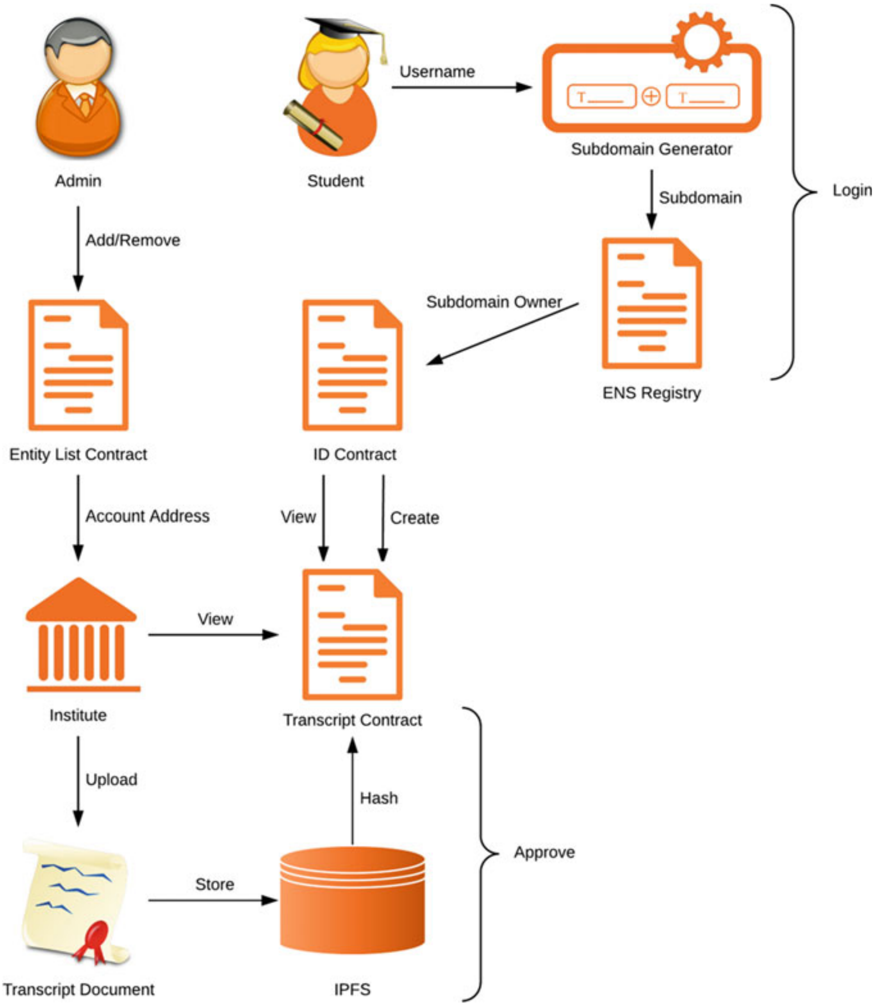


Fig. 1 Architecture of transcripts DApp

4.2 Description of Major Functions

The implemented prototype performs the following major functions:

Log-in and Sign-up

The implemented prototype uses a universal log-in and sign-up procedure as described in Ethereum Improvement Proposal EIP-1078 [12]. The DApp controls a high-level domain name based on the Ethereum Name Service. The username is used to create a unique sub-domain for each user, which is the only input required from the user for log-in. The created sub-domain is assigned to an ID contract, as defined in the Ethereum Improvement Proposal EIP-725 [13]. After the initial creation of sub-domains, the user may log-in using the sub-domain name and using a wallet provider that is registered as a level-1 (Admin) key of the ID contract.

The process of universal log-in and sign-up is as follows:

1. A user who wishes to log-in provides a username, which is prefixed to the domain name to derive a sub-domain.
2. The DApp checks availability of entered sub-domain name with the ENS registrar on the current Ethereum network.
3. If sub-domain name is taken, skip to step 6. Else, if sub-domain is available, the DApp prompts user to verify entered name, or create a new sub-domain
4. For creating a new sub-domain, the DApp creates a new ID contract for the user and assigns the current wallet being used by user as the Admin Key (level-1 key) of the ID contract.
5. The DApp registers the desired sub-domain and transfers its ownership to the newly created ID contract.
6. If user attempts to log into a previously registered sub-domain, the DApp checks if the key is present in the associated ID contract.
7. If the key is already present in the ID contract, the DApp concurs that the user is verified and successfully logs them in. Else, the DApp asks for a new transaction for entering the new key as an Encryption Key (level-4 key).
8. By measures provided in the ID contract, addition of keys can be done successfully only when the transaction is signed by an Admin Key (level-1).
9. The DApp provides a QR code or a URL for the user to sign the transaction using their admin key.
10. After success of the transaction, the user is logged into the DApp.

For institutes and admins, the procedure for registration differs. They are required to contact the current admins at the provided correspondence points. After performing the requisite authentication and verification of their identities, the admins themselves add new institutes and admins. The list of current admins and institutes is stored using a smart contract. The admins also hold the power to remove institutes found guilty of misconduct.

The process of registration and log-in for academic institutes is as follows:

1. The representative of the institute contacts the administrators regarding registration of the institute on the DApp.
2. The administrators, after performing some requisite manual verification, request the Ethereum account address which would be used to represent the institute.
3. The administrator adds the institute as a name–address pair into a Entity List contract.

4. The Entity List contract contains the names and address of all the institutes registered with the DApp.
5. The Entity List contract has built-in checks to only allowing existing administrators, the permission to add new institutes, or remove existing institutes.
6. The institute may then log-in using their registered account address.

Create Transcript Application

The DApp creates a new smart contract for each new application made by a student. The contract stores the details provided by the student and also identifies the institute that is responsible for approving the application. The DApp determines the status of application using the presence or absence of the hash value.

The DApp keeps track of all the created applications using a contract that stores the list of applications. One list is maintained for each user, which stores both unapproved and approved applications.

The process of creating a new transcript application is as follows:

1. The student logs into the DApp using the previously mentioned procedure.
2. The student chooses to create a new application and fills in the required details in an application form.
3. The student may select any institute they desire to obtain their transcripts from, known as the provider, from the list of institutes registered with the DApp.
4. When the student submits the application form, the DApp validates all fields to check for proper input formats for all fields.
5. The application may or may not require a fee for processing, as per the requirements of the institute.
6. The DApp creates a new Transcript Application contract using the data received from the student.
7. The Transcript Application contract has a field which determines the institute that can approve the application.
8. The Transcript Application contract also has a field that stores the hash value of the uploaded transcript obtained from IPFS. A non-existent hash value denotes that the application is yet to be approved.
9. The DApp stores the value of the newly deployed Transcript Application contract in the Transcript List contract.
10. The Transcript List contract has a list of all associated contracts for each user. For students, it stores the addresses of all the Transcript Application contracts that have been created by them. For institutes, it stores the addresses of all Transcript Application contracts that have been created with them as the authorized provider.

Approve Transcript Application

The smart contract is designed such that it can be approved only by the designated institute. The institute merely uploads the transcript document using the DApp. The process of storing the document on IPFS, storing the obtained IPFS hash in the contract, and fetching of the document using the stored IPFS hash are performed by the DApp behind the scenes. Once the file is uploaded and the hash in the smart contract is set, the DApp considers the application as approved.

The process of approving a transcript application and uploading the transcript is as follows:

1. The institute logs into the DApp using their registered account address.
2. The DApp displays a list of all transcripts that mention the concerned institute as the provider using the data stored in the Transcript List contract.
3. The institute may view the details of the application and the applicant as fetched from the Transcript Application contract.
4. The institute may perform some requisite verification to determine whether the applicant mentioned qualifies for the desired transcripts.
5. Once verification is complete, the institute prepares a transcript which may be completely digital or may have physical copies. Either way, the transcripts must be converted into PDF file format for uploading.
6. When the institute uploads the contract, the DApp connects to a remote IPFS node to upload the file onto IPFS.
7. The IPFS upload method returns a hash which is stored in the Transcript Application contract.
8. Due to built-in checks, the Transcript Application contract only permits the institute specified by the user to set the field denoting the IPFS hash of the transcript.
9. The Transcript Application is thus considered as approved and the stored transcript may be accessed by the student.
10. The IPFS hash uniquely determines the stored transcript, and the immutability of blockchain ensures that the hash value once set in the Transcript Application contract cannot be changed by any other means.

This procedure uses the properties of both smart contracts and IPFS to ensure the integrity and authenticity of stored transcripts.

5 Smart Contract Pseudo-Code

The pseudo-codes of the smart contracts utilized in the DApp are described below:

Transcript Application Contract

```
constructor (address _owner, address _provider,
    string _name, string _id, string _courseName,
    int _startYear, int _completionYear) {
  transcriptHash = "Not set";
  transcriptOwner = _owner;
  providingAuthority = _provider;
  name = _name;
  id = _id;
  courseName = _courseName;
  courseStartYear = _startYear;
  courseCompletionYear = _completionYear;
}

function setTranscriptHash (string s) returns (string) {
  if(msg.sender != providingAuthority) {
    return "Error";
  }
  transcriptHash = s;
  return "Success";
}
```

Transcript List Contract

```
mapping(address => address[]) transcriptList;

function addTranscript (address student, address
    college, address transcriptAddress) {
  transcriptList[student].add(transcriptAddress);
  transcriptList[college].add(transcriptAddress);
}

function getTranscripts (address account) returns
    (address[]) {
  return transcriptList[account];
}

function removeTranscript (address transcriptAddress) {
  foreach(entity in transcriptList) {
    entity.remove(transcriptAddress);
  }
}
```

Entity List Contract

```
function addAdmin (address addr) returns (string) {
  if(!isAdmin(msg.sender)) {
    return "Error";
  }
  admins.add(addr);
  return "Success";
}

function addProvidingAuthority (string memory name,
    address addr) returns (string) {
  if (!isAdmin(msg.sender)) {
    return "Error";
  }
  providingAuthorities.add(ProvidingAuthority(name,
      addr));
  return "Success";
}

function removeProvidingAuthority (address addr)
    returns (string) {
  if (!isAdmin(msg.sender)) {
    return "Error";
  }
  providingAuthorities.remove(addr);
  return "Success";
}
```

6 Costs and Scaling

The costs associated with various types of transactions in the DApp are shown in Table 1. These transaction costs were determined through the Ropsten test network based on the value of Ether at the time of writing.

Apart from the costs mentioned in Table 1, institutes may also choose to charge some fees for the transcripts, which can be paid during the submission of application.

The scaling of the DApp is directly tied to the rate of transactions on the Ethereum network. As of writing, Ethereum supports a rate of about 15 transactions per second and a block time of about 10–20 s.

Table 1 Transaction costs

Role	Action	Approximate cost-ether (USD)
User	Registration	0.000311 (0.04)
	New transcript application	0.000291 (0.03)
Institute	Uploading transcript	0.000022 (0.00)
Administrator	Add new institute	0.000021 (0.00)
	Add new administrator	0.000015 (0.00)
	Remove institute	0.000012 (0.00)

7 Conclusions and Future Work

We conclude that the proposed DApp has significant benefits in terms of costs, time required and convenience. The DApp is fully decentralized, which eliminates any dependency on third-party verification agencies. Using smart contracts, we ensure that only the authorized institute can issue transcripts to students. The hash of uploaded transcripts uniquely determines the transcript file on IPFS, thus ensuring that the file cannot be tampered with.

The starting point for future work would be working on a concrete proof of concept which delivers a decentralized trust-less system as proposed in the design while providing at least the same level of security as provided by current system of physical transcripts.

Currently, the proposed system is limited to handle academic transcripts. Possible areas of extension of the proposed system would be design of similar systems to handle other documents such as certificates and performance reports or integration of all of the above in a single system.

References

1. Nakamoto S (2009) Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf
2. Buterin V and others (2014) A next-generation smart contract and decentralized application platform
3. IPFS, https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf. Accessed 10 December 2018
4. Maurer U (1996) Modelling a public-key infrastructure. In: European symposium on research in computer security. Springer, pp 325–350
5. Ellison C, Schneier B (2000) Ten risks of PKI: what you're not being told about public key infrastructure. Comput Security J 16(1):1–7 (2000)
6. SGE Education blockchain, https://blockchain.sonyged.com/. Accessed 10 December 2018
7. BlockCerts, https://www.blockcerts.org/about.html. Accessed 10 December 2018
8. Blockchain Imperative for Educational Certificates, https://gosmartchain.com/whitepaper/SmartChainUniversity-whitepaper.pdf. Accessed 11 December 2018

9. Blockchain-based recruitment and background verification platform, https://cverification.com/src/docs/Cverification_Whitepaper.pdf. Accessed 11 December 2018
10. BCDiploma, https://www.bcdiploma.com/ico/img/BCD-WhitePaper_last.pdf. Accessed 11 December 2018
11. Cheng J, Lee N, Chi C, Chen Y (2018) Blockchain and smart contract for digital certificate. In: 2018 IEEE international conference on applied system invention (ICASI), pp 1046–1051 (April 2018)
12. Van de Sande A (2018) EIP 1078: Universal login/signup using ENS sub-domains. https://eips.ethereum.org/EIPS/eip-1078. Accessed 10 December 2018
13. VogelSteller F (2018) EIP 725: Proxy Identity. https://eips.ethereum.org/EIPS/eip-725. Accessed 10 December 2018

Application of Blockchain Technology in Civil Registration Systems

Vennis Shah, Karnika Padia, and Vivian Brian Lobo

Abstract Smart conurbation treatments are surfacing the route for urbanization, which indicates that government or administrative bodies, citizenries, and corporations need to incorporate smart solutions to expound systems that can implement a specific task in a disciplined way supplanting outmoded systems. Civil registration systems are one of the outmoded systems that can be made straightforward through the application of blockchain. Transparency, immutability, protection, and confidentiality—being some of the conspicuous blockchain characteristics—make it a podium of choice for digitizing civil registration systems. This study aims to design a decentralized application (DApp), i.e., a government or an administrative gateway, to perform various activities of civil registration systems using blockchain. The designed system generates birth as well as death certificates by way of smart contracts on a permission less network using Ethereum—a blockchain-based distributed processing platform. Moreover, the paper presents DApp implementation accompanied by several frameworks such as Truffle, Ganache, MetaMask, Remix Editor, Solidity, and Web3.js. This designed system guarantees a simplified registration process and offers increased data transparency and effective record maintenance.

Keywords Blockchain · Civil registration · Decentralized application

V. Shah (✉) · K. Padia
Department of Information Technology, St. Francis Institute of Technology, Mumbai, India
e-mail: vennis.shah98@gmail.com

K. Padia
e-mail: kpadia05@gmail.com

V. B. Lobo
Department of Computer Engineering, St. John College of Engineering and Management, Palghar, Mumbai, India
e-mail: vbl2781991@gmail.com

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_16

1 Introduction

Blockchain is an emergent record list, known as blocks, that is, linked using cryptography. Every single block comprises transaction data, timestamp, and cryptographic hash of a previous block [1]. Transaction entails receivers' public key and is signed by a sender using his/her private as well as public keys. A transaction is placed in a ledger—a list of all transactions—only after the signature of a legitimate user.

1.1 Characteristics of Blockchain

The main characteristics of blockchain are as follows:

- *Inalterable*: Once a transaction is completed, a block is appended to the chain and that blocks' content remains unchanged. In other words, this indicates high data security.
- *Falsifiable*: When a block is generated, previous blocks' hash is stored in current block—which can be referenced. In addition, the current block contains its own hash that can be used to connect to the next block. This means that every single block can be authenticated autonomously.
- *Distributed unanimity*: It facilitates recording of every single transaction and distributing it throughout the network. All users in the network can verify transactions and have a ledger's duplicate print. Modifications to the ledger are exhibited in all prints in a fraction of seconds or sometimes minutes. Security and precision of assets are cryptographically preserved by means of keys and digital signatures, which are controlled by users [2].

1.2 Classification of Network in Blockchain

- *Centralized network in blockchain*: Herein, all nodes are linked by a single node, and an organization that deploys the blockchain network can determine which node can connect to the network, thereby ensuring security. Nonetheless, data maintenance at a single node is problematic. Attacking one node is simpler than attacking numerous nodes instantaneously. Once an attack is performed on the central node, there is no substitute. This consecutively affects the complete network that can lead to data loss or addition of untruthful data (Fig. 1).
- *Decentralized network in blockchain*: Herein, ledgers are amassed at different locations; therefore, to attack a decentralized blockchain network, an attacker has to maintain a track of every single ledger. In decentralized network, a transaction can be added to a network only when all nodes verify and sign it using their digital signatures. Consequently, unethical ledger modification is practically impossible since legitimate nodes will oppose attacker's transaction, and the attacker's block will not be added to the ledger. This feature of decentralized network in

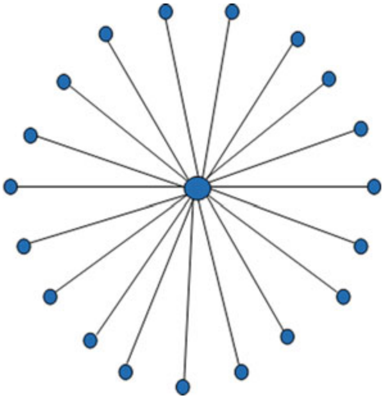


Fig. 1 Centralized network

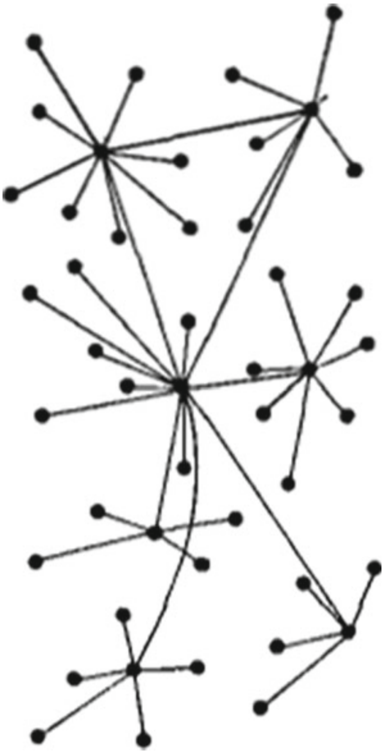


Fig. 2 Decentralized network

blockchain makes it much more secure when compared with centralized network in blockchain [3] (Fig. 2).

In this study, a decentralized smart contract mechanism is used.

- *Decentralized smart contract*: Ethereum is one of the most widely used smart contract platform to create decentralized smart contracts. Ethereum uses ether as its currency. A smart contract [4] represents the idea that legal contracts can be automatically notarized and executed. Solidity programming is the most favorable language that is used to develop smart contracts with Ethereum.

1.3 Confronts in Existing Civil Registration Systems

Absence of public consciousness and less requirement for civil registration papers, i.e., birth and death certificates, are one of the main confronts for low level of registrations. For instance, when an infant is born, parents need to visit a government organization to register an infant's birth. Conventional government paper work includes collecting a registration form, filling appropriate details, and submitting photocopies of required documents over and above carrying original documents for verification. Just in case there is an error, the entire wearisome procedure needs to be performed again, which ultimately leads to a troublesome task of registering an event in government records—thereby making the entire process inefficient and unproductive.

1.4 Resolution

A handful of problems in birth and death registrations can be put to an end by blockchain-based digitization. As mentioned above, blockchain proves to be inalterable, it provides true origin and conclusiveness, which makes it a platform of selection for digitizing birth and death records. Application of blockchain technology in civil registration systems can lessen parent's effort to an enormous extent. Rather than visiting government organizations and adhering to the conventional process, hospitals can give authorities to certain members to verify documents and enter details of an infant during birth. These details can be sent to government organizations (herein, nodes) through smart contracts. By means of transaction details and smart contract address, a government node can mine the block. Such a digital implementation of civil registrations will reduce manual documentation, decrease human errors, save time, keep data tamper-proof, and increase pellucidity between government organizations and hospitals. Similarly, death certificates and other registrations can be carried out without visiting government organizations.

2 Literature Review

Bayu et al. [5] studied up-to-the-minute applications related to blockchain and provided perceptions about blockchain and its associated terms. Friðrik et al. [6] assessed blockchain-as-a-service to execute distributed e-voting systems and proposed a blockchain-based e-voting system that discourses existing system limitations. Moreover, they assessed some blockchain backgrounds for constructing a blockchain-based e-voting system. Heng [7] discussed about blockchain applications in e-governance, its advantages, challenges, and other areas of application. Ahmed et al. [8] explained about blockchain components and its use cases in public sectors. Pinyaphat et al. [9] conducted a review on blockchain technology and determined its challenges. In addition, they provided an overview about Bitcoin. Few existing financial and non-financial blockchain applications were also discussed. Chengjun et al. [10] recapitulated existing blockchain technologies and elaborated the philosophies of designing and instigating secure distributed applications and analyzed security concerns. Peng et al. [11] focused on blockchain requirement in healthcare domain to resolve challenges such as gapped interactions, incompetent medical report distribution, and disjointed health documents. Wei et al. [4] reviewed the history of blockchain and clarified common definitions. Ting et al. [12] conducted a study on Ethereum by graphical examination to exemplify money transfer and smart contract creation and invocation. A cross-graph scrutiny facilitated them in addressing a few security issues in Ethereum.

3 Proposed System for Digitizing Civil Registrations

Figure 3 shows the block diagram of the proposed system for civil registrations that makes such systems uncomplicated and reachable to individual citizenries. Data will be accumulated in the form of blocks (i.e., smart contracts). This data can be straightforwardly mined by government organizations for record purposes. The flow of the proposed system can be described as follows:

- *For birth certificate registrations*:

    1. A citizen *(i.e., an infant)* is born at a hospital.
    2. Hospital authorities verify basic documents.
    3. The authorities login into the registration portal.
    4. Relevant information is placed into a smart contract and an event is registered.
    5. The block is mined by government organizations, which then generates a digital copy of birth certificate.

    Similar procedure is followed for death certificate registrations.

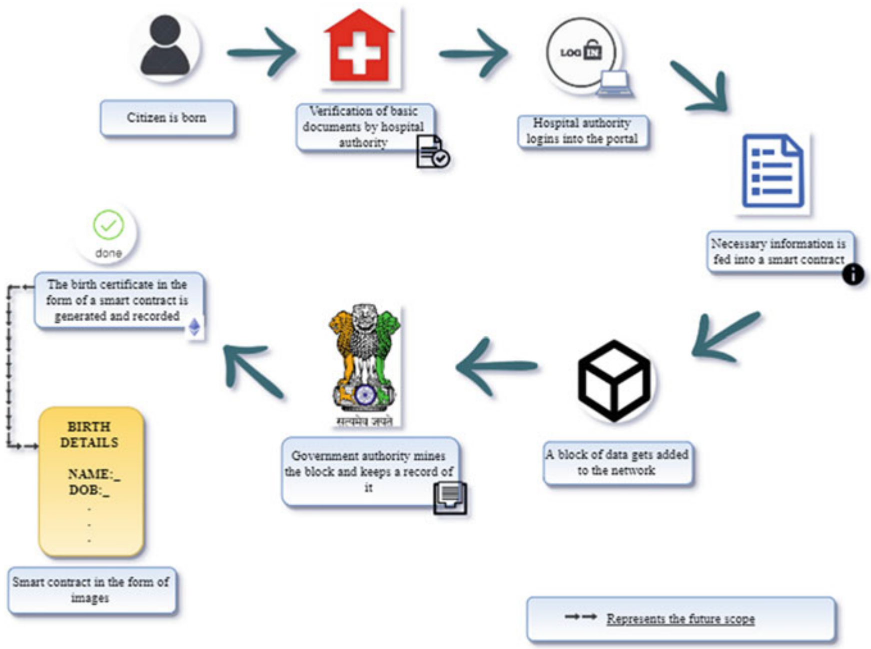

Citizen is born
Verification of basic documents by hospital authority
LOG IN
Hospital authority logins into the portal
Necessary information is fed into a smart contract
A block of data gets added to the network
सत्यमेव जयते
Government authority mines the block and keeps a record of it
done
The birth certificate in the form of a smart contract is generated and recorded
BIRTH DETAILS
NAME:_
DOB:_
Smart contract in the form of images
Represents the future scope

Fig. 3 Proposed system for civil registrations

4 System Design

For designing the proposed system, the following frameworks were used.

4.1 Truffle

It is used for estimating blockchain using Ethereum virtual machine. Some of its qualities include:

- *A migrations framework.*
- *Network supervision for deploying smart contracts to numerous public and private networks and a console for unproblematic contract communication.*

4.2 Ganache

It is a specific blockchain that is used for Ethereum progression. When Ganache is launched, a screen appears that displays server-related aspects. Ganache offers subsequent options, i.e., *accounts option* displays generated accounts and resultant account balances. *Blocks option* shows mined block factors accompanied by block number, amount of gas expended, and business transactions. *Transactions option* provides a list of transactions that are completed, and *logs option* enlists logs for the server.

4.3 MetaMask

It is a suspension bridge that permits a user to visit distributed Web of tomorrow in today's browser. MetaMask for diverse browsers can be operated to execute Ethereum decentralized applications (DApps) in a user's browser devoid of unambiguously running an entire Ethereum node. MetaMask consists of a tamper-proof identity crypt that offers a UI to uphold an individual's identity on a variety of sites and alphanumerically signed blockchain transactions. MetaMask add-on can be installed in Google Chrome, Mozilla Firefox, and Opera. Moreover, it provides a user the freedom to connect to a private or public network.

4.4 Remix Editor and Solidity

Smart contracts can be efficiently written in solidity using Remix tool. Remix Editor registers code and generates corresponding application binary interface (ABI) and byte code. With the help of Remix Editor, a user can deploy smart contracts to a particular address. Once the smart contract is deployed, the editor creates a smart contract address.

Solidity is a high-level programming language that is used for smart contract execution. Smart contracts help in understanding the behavior of Ethereum accounts.

4.5 Web3.js

Web3.js is a compendium of libraries, wherein one can communicate with both a local and remote Ethereum node using hypertext transfer protocol or interprocess communication connection.

5 System Implementation

```
pragma solidity ^0.4.2; //indicates the compiler version
contract Coursetro { //defining variables of type string
   string Name;
   string fName;
   string mName;
   string dob;
   string sex;
   string pob;
   string add
/*function to set the values of variables in the smart
contract*/
function setInstructor (string _Name, string _fName,
string _mName, string _dob, string _sex, string _pob,
string _add) public
{
     Name=_Name;
     fName = _fName;
     mName=_mName;
     dob=_dob;
     sex=_sex;
     pob=_pob;
     add=_add;
}
/*function to get the values of variables in the smart
Contract*/
function getInstructor () public constant returns
(string, string, string, string, string, string, string)
{
     return (Name, fName, mName, dob, sex, pob, add) ;}}
```

Figure 4 shows the home page of government portal and details that need to be filled in government portal for the process of civil registrations. From here, hospital authorities can either register themselves if they wish to get associated with government authorities or already registered hospitals can login to carry out further

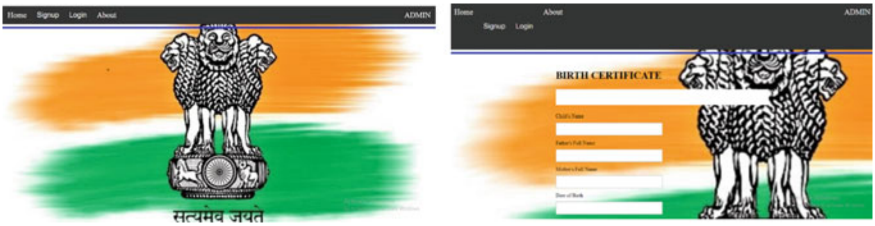


Fig. 4 Home page of government portal and details that need to be filled in government portal

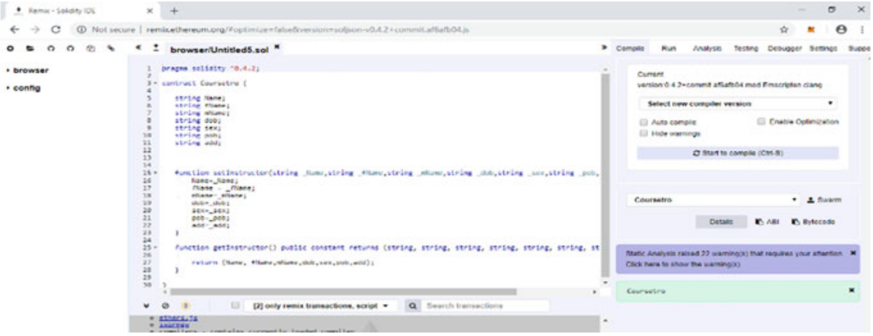
processes. Also, it shows the details that need to be filled in government portal Web page where hospital authorities feed an infant's details, which in turn will be fed into a smart contract.

```
if (typeof web3 !== 'undefined') {
            web3 = new Web3(web3.currentProvider);
        } else {
        //set the provider you want from Web3.providers
web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:7545"));
        }
web3.eth.defaultAccount = web3.eth.accounts [0];
//ABI stands for application binary interface
var CoursetroContract = web3.eth.contract([ABI]);
var Coursetro = CoursetroContract.at('0xc5ae4282b30402fbde783237d55532a599783870');
        console.log(Coursetro);
    Coursetro.getInstructor(function(error, result){
            if(!error)
                {
                    $("#instructor").html('Childs
Name:'+result[0]+','+'Fathers Name:'+result[1]/*+','+'Mothers name:'+result[2]+'Date of
birth:'+result[3]+','+'Sex:'+result[4]+','+'Place
ofbirth:'+result[5]+','+'Address:'+result[6]*/);
          console.log(result);
                }
            else
                console.error(error); });

        $("#button").click(function() {
       Coursetro.setInstructor($("#name").val(),
$("#ffn").val(),$("#mfn").val(), $("#dob").val(),
$("#sex").val(),$("#pob").val(),$("#add").val() );
        });
```

The above code is a smart contract code, which is implemented using solidity programming. This code is used for inserting values entered by hospital authorities into a smart contract using *setInstructor()* and is later fetched using *getInstructor()*. Solidity 0.4.2 compiler version is used.

Figure 5 shows the snapshot of remix editor, which is used to compile smart contract code. ABI is generated by the editor, which is data encoding strategy used in Ethereum to work with smart contracts. The associated byte code is generated.

The above code is the pseudo code for Web3.js. The contract address is also added in the contract. For example, in the above code snippet, the contract address is *0xc5ae4282b30402fbde783237d55532a599783870*. This piece of code is written inside the script tag in an HTML file for the birth certificate form.



```
pragma solidity ^0.4.2;

contract Coursetro {

    string Name;
    string fName;
    string mName;
    string dob;
    string sex;
    string pob;
    string add;


    function setInstructor(string _Name,string _fName,string _mName,string _dob,string _sex,string _pob,
        Name=_Name;
        fName = _fName;
        mName=_mName;
        dob=_dob;
        sex=_sex;
        pob=_pob;
        add=_add;
    }

    function getInstructor() public constant returns (string, string, string, string, string, string, st

        return (Name, fName,mName,dob,sex,pob,add);
    }

}
```

Current
version:0.4.2+commit.af6afb04.mod.Emscripten.clang

Coursetro

Static Analysis raised 22 warning(s) that requires your attention.
Click here to show the warning(s)

Coursetro

Fig. 5 Snapshot of Remix Editor

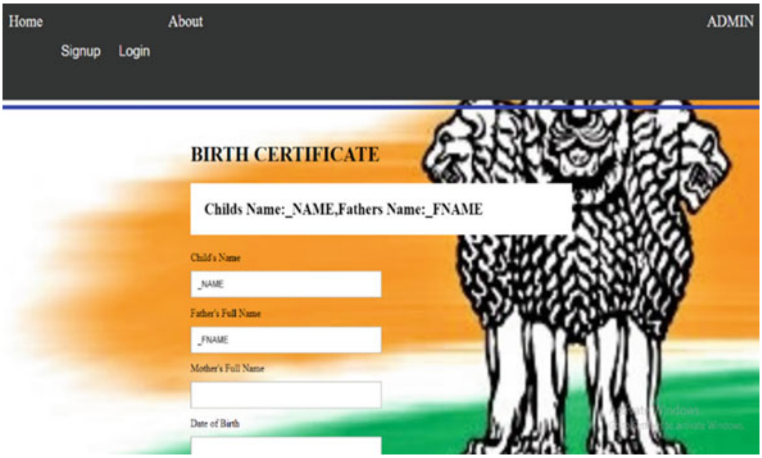
Figure 6 shows that the details that are fed into the smart contract are later fetched and displayed.

Figure 7 shows the snapshot of Ganache, which is a personal blockchain that can be used for Ethereum development. Ganache deploys smart contracts as well as a user can perform specific tests. Herein, ten accounts are created with 100.00 ETH each.

Figure 8 shows the addition of a custom network address http://localhost:7545 that is used by Ganache by means of custom remote procedure call option in MetaMask. Also, connectivity to http://localhost:7545, which is supported by Ganache is shown.

Figure 9 shows that by clicking on the account from Ganache, the account's private key can be obtained, which can be pasted in MetaMask to import this account in MetaMask. It also shows that the account is successfully imported

Figure 10 shows the snapshot of Ganache where list of blocks that are mined



BIRTH CERTIFICATE

Childs Name:_NAME,Fathers Name:_FNAME

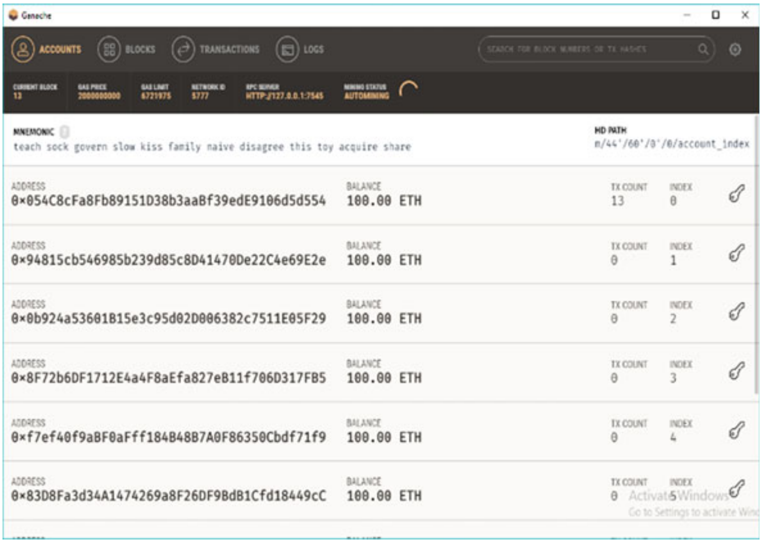
Child's Name
_NAME

Father's Full Name
_FNAME

Mother's Full Name

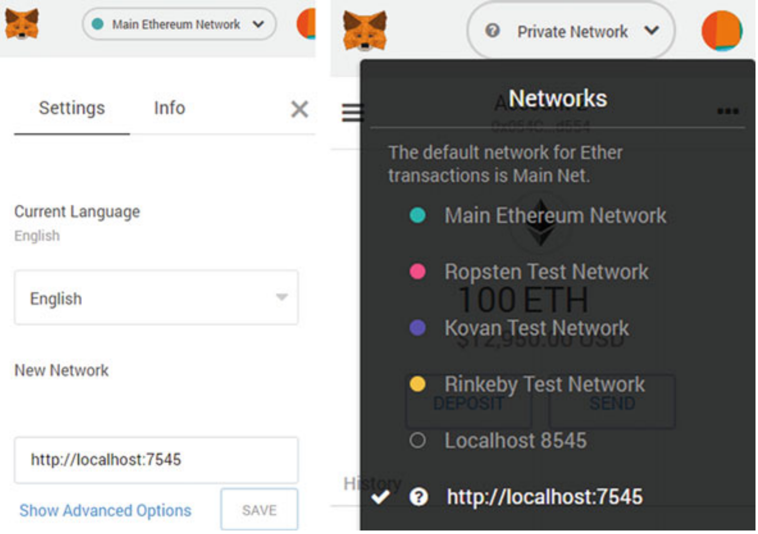
Date of Birth

Fig. 6 Displaying entered values



```
CURRENT BLOCK  GAS PRICE    GAS LIMIT  NETWORK ID  RPC SERVER              MINING STATUS
13             2000000000   6721975    5777        HTTP://127.0.0.1:7545   AUTOMINING

MNEMONIC                                                                  HD PATH
teach sock govern slow kiss family naive disagree this toy acquire share  m/44'/60'/0'/0/account_index

ADDRESS                                      BALANCE      TX COUNT  INDEX
0x054C8cFa8Fb89151D38b3aaBf39edE9106d5d554   100.00 ETH   13        0

ADDRESS                                      BALANCE      TX COUNT  INDEX
0x94815cb546985b239d85c8D41470De22C4e69E2e   100.00 ETH   0         1

ADDRESS                                      BALANCE      TX COUNT  INDEX
0x0b924a53601B15e3c95d02D006382c7511E05F29   100.00 ETH   0         2

ADDRESS                                      BALANCE      TX COUNT  INDEX
0x8F72b6DF1712E4a4F8aEfa827eB11f706D317FB5   100.00 ETH   0         3

ADDRESS                                      BALANCE      TX COUNT  INDEX
0xf7ef40f9aBF0aFff184B48B7A0F86350Cbdf71f9   100.00 ETH   0         4

ADDRESS                                      BALANCE      TX COUNT  INDEX
0x83D8Fa3d34A1474269a8F26DF9BdB1Cfd18449cC   100.00 ETH   0         5
```

Fig. 7 Snapshot of Ganache accounts



Current Language
English

English

New Network

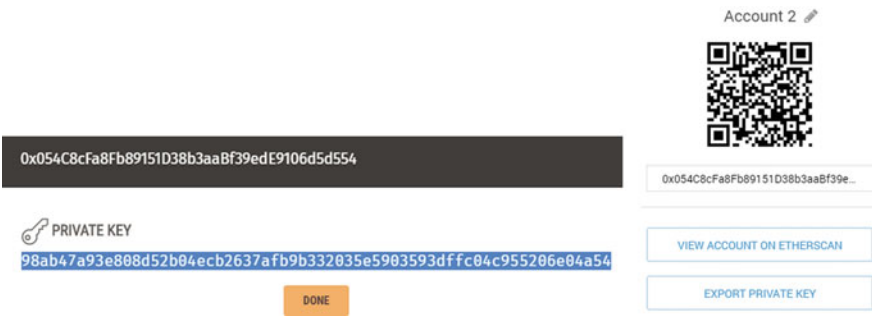
http://localhost:7545

Show Advanced Options SAVE

The default network for Ether transactions is Main Net.

- Main Ethereum Network
- Ropsten Test Network
- Kovan Test Network
- Rinkeby Test Network
- Localhost 8545
- ✓ http://localhost:7545

Fig. 8 Addition of a custom remote procedure call option to the network and selecting the added network, which is a private network

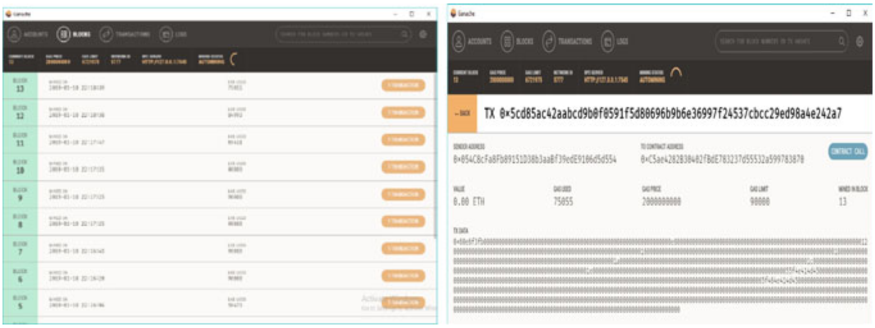


0x054C8cFa8Fb89151D38b3aaBf39edE9106d5d554

PRIVATE KEY

98ab47a93e808d52b04ecb2637afb9b332035e5903593dffc04c955206e04a54

Fig. 9 Private key of account to be imported and successfully importing the account



TX 0x5cd85ac42aabcd9b0f0591f5d80696b9b6e36997f24537cbcc29ed98a4e242a7

0x054C8cFa8Fb89151D38b3aaBf39edE9106d5d554

0xC5ae4282838402fBdE783237d55532a599783878

0.00 ETH

75055

20000000000

90000

13

Fig. 10 Blocks after they are mined and the transaction details of blocks that are mined

can be viewed. It also shows details such as *block number*, *amount of gas used*, and *the date and time when a block is mined*. It also shows transaction details of the block that is mined. It shows details such as sender's address, i.e., the address of the account that is imported in MetaMask to carry out the transaction. This address can be verified with the account address, which is displayed above in Fig. 9.

Table 1 shows an example of a transaction in our developed system for civil registrations.

Table 1 Example of a transaction

TxHash	Block number	Sender address	To contract address	Gas used
0x5cd85ac42a..	13	0x054C8cFa….	0xC5ae42…..	75,055

6 Results and Discussion

As block size increases, blockchain network increases, which results in scalability issues. Note that the scalability can be reduced by reducing block interval to achieve high throughput and efficiency. Moreover, blockchain possesses overhead problems in terms of bandwidth as well as storage space, which is a challenge. In addition, it is assumed that basic document verification is performed by trusted hospital authorities, and there are no malicious activities taking place. In the proposed system, it is stated that government officials will have to mine data blocks, but since Ganache is used, which auto-mines blocks, it is impossible to explicitly illustrate how government officials mine data blocks.

7 Conclusion and Future Scope

This paper presented an application of blockchain technology in civil registration systems that generate both birth as well as death certificates through smart contracts and assures safety, confidentiality, and makes smart contracts tamper-proof. Smart contracts are written using solidity programming and compiled via Remix Editor. The main advantage of using the developed system includes minimum interaction with government organizations, thereby making the entire procedure quicker and reduces human errors. In the near future, the developed system could be improved by including smart contracts in the form of images. In addition, similar concepts could be used to develop other registration applications in land transactions, marriage certificates, and many more sectors.

References

1. Blockchain [Online] https://en.wikipedia.org/wiki/Blockchain (Accessed 15th January 2019)
2. Tech Racers, 4 key features of blockchain, [Online] https://www.techracers.com/blogs/blockchain-key-features/ (Accessed 15th January 2019)
3. Records Keeper, Centralized vs. Decentralized Blockchain, [Online] https://www.recordskeeper.co/blog/centralized-vs-decentralized-blockchain/ (Accessed 15th January 2019)
4. Wei C, Wang Z, Ernst JB, Hong Z, Feng C, Leung VCM (2018) Decentralized applications: the blockchain-empowered software system. IEEE Access 6:53019–53033
5. Bayu Adhi T, Kweka BJ, Park Y, Rhee K-H (2017) A critical review of blockchain and its current applications. In: 2017 International Conference on Electrical Engineering and Computer Science (ICECOS). IEEE, pp 109–113
6. Friðrik ÞH, Hreiðarsson GK, Hamdaqa M, Hjálmtýsson G (2018) Blockchain-based E-voting system. In: 2018 IEEE 11th international conference on cloud computing. IEEE, pp 983–986
7. Heng H (2017) The application of blockchain technology in E-government in China. In: 2017 26th International Conference on Computer Communication and Networks (ICCCN). IEEE, pp 1–4

8. Ahmed A, Nasir Q, Talib M A (2018) Blockchain for government services—use cases, security benefits and challenges. In: 2018 15th international conference on Learning and Technology (L&T). IEEE, pp 112–119
9. Pinyaphat T, Techapanupreeda C (2018) Blockchain: challenges and applications. In: 2018 International Conference on Information Networking (ICOIN). IEEE, pp 473–475
10. Chengjun C, Duan H, Wang C (2018) Tutorial: building secure and trustworthy blockchain applications. In: 2018 IEEE international conference on Cybersecurity Development (SecDev), IEEE, pp 120–121
11. Peng Z, Walker MA, White J, Schmidt DC, Lenz G (2017) Metrics for assessing blockchain-based healthcare decentralized apps. In: 2017 IEEE 19th international conference on e-health networking, applications and services (Healthcom). IEEE, pp 1–4
12. Chen T, Zhu Y, Li Z, Chen J, Li X, Luo X, Lin X, Zhange X (2018) Understanding Ethereum via graph analysis. In: 2018 IEEE conference on computer communications. IEEE, pp 1484–1492

Land Record Maintenance Using Blockchain

Harshita Bhorshetti, Shreyas Ghuge, Athang Kulkarni, and Sukhada Bhingarkar

Abstract Storing data in real time along with keeping it secure is the biggest challenge in industry today. Many land issues arise because there is no database which is protected for tracking the real-time changes in data. Also, land records currently are registered in paper format. This kind of data is thus vulnerable to any changes and maybe destroyed by natural or man-made disasters. The emerging blockchain technology is a boon to store any information in real time and is immune to any changes. In this paper, we propose a solution in the form of distributed app (DApp) which uses the idea of blockchain as distributed database, smart contracts using ethereum platform and Polyline API from Google to mark the land boundaries. Smart contracts allow the performance of credible transactions by using sophisticated cryptography and without interference from third parties. These transactions are traceable and irreversible. Proponents of smart contracts claim that many kinds of contractual clauses may be made partially or fully self-enforcing. In this case, along with the self-verifiable clauses, involving banking parties can perform additional monetary checking. A user can sell or transfer a property that he owns or may buy a new land plot open for sale in desired geographical area. This solution allows maintaining land records easily and in real time without having a single point of failure for the database system. Removal of third-party interventions such as brokers from the process of land title transfer between old and new owners makes the process more transparent and cheaper.

H. Bhorshetti (✉) · A. Kulkarni · S. Bhingarkar
Department of Computer Engineering, MIT College of Engineering, SPPU, Pune, India
e-mail: hbhorshetti@gmail.com

A. Kulkarni
e-mail: athang.kulkarni03@gmail.com

S. Bhingarkar
e-mail: sukhada.bhingarkar@mitcoe.edu.in

S. Ghuge
Department of Information Technology, MIT College of Engineering, SPPU, Pune, India
e-mail: ghugeshreyas2@gmail.com

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_17

Keywords Block chain · Cryptography · DApp · Digital signature · Distributed database · Ethereum · Hash values · Polyline API · Smart contracts

1 Introduction

1.1 What Is Blockchain?

Blockchain [1] is an encrypted, distributed database that records data, or in other words, it is a digital ledger for transactions or contracts—that need to be recorded. It involves creating verification records for digital files. These verification records are the uniquely identifiable hash values calculated by using the data being stored in the actual files and are grouped into an entity called as block' [2]. The block is then added to a chain of blocks such that it includes hash values of the block that is preceded by it. This creates a chain of hash values all the way back to the first block. Therefore, it is impossible to alter information stored in older blocks without changing the subsequent blocks because changing the block would change its hash value and disprove the chain.

Confidence in the original transactions and documents [3] improves when several actors have access to the blockchain's records. When the verification records are open and difficult to manipulate, there is less reason to question them, and trust and confidence in them grow significantly. Hence, blockchain has the ability to create a securely shared history of transaction.

1.2 Ethereum—an Application of Blockchain

Blockchain [4] is one of the most talked technologies in recent years and the best and well-known application of blockchain is 'Bitcoin' [5]. It is a digital currency which is governed by computers for their authenticity and usage, uses peer-to-peer technology to operate with no central authority such as banks or governments, managing transactions and the mining of bitcoins are carried out collectively by this network of computers.

Another publically distributed network is 'Ethereum' [6]. Although, similar to Bitcoin, Ethereum differs substantially in purpose and capability. While the Bitcoin blockchain is used to track ownership of digital currency (bitcoins), the Ethereum blockchain focuses on running the programming code of any decentralized application. Rather than giving a set of limited operations, Ethereum allows developers to create desired contracts. It allows developers to create smart contracts [7], which is an agreement that self-executes and handles the enforcement, the management, performance, and payment of legal tender in some cases.

Ethereum has the following advantages [6] when used as a platform:

- **Immutability**—A third-party cannot make any changes to data.
- **Corruption and tamper proof**—Apps are based on a network formed around the principle of consensus, making censorship impossible.
- **Secure**—With no central point of failure and secured using cryptography, applications are well protected against hacking attacks and fraudulent activities.

1.3 Blockchain in Land Registry

When a purchaser seeks to buy illicit property today, he must make sure that the property he buys is legal and needs lot of paper work for the same. This seems simple on surface but for a large number of residential mortgage holders, flawed paperwork, forged signatures, defects in foreclosure, and mortgage documents have marred proper documentation of property ownership. By using hashes [8] to identify every real estate transaction (thus making it publicly available and searchable), proponents can argue on issues such as who is the legal owner of a property.

Various programs [9] have been launched with an aim to computerize all land records, including mutations, improve transparency in land record maintenance system, update all settlement records, and minimize scope of land disputes. Along with digitization [2], a level of trust needs to be developed among people for creating, enacting, verifying, storing, and securing these digital contracts. Blockchain technology [2] is a method which enables the above-listed functions effectively since it is based on consensus from various nodes after which that particular transaction block is added to every node in the distributed database, i.e., Blockchain. Thus, use of blockchain technology serves as the basis for a more reliable, cheaper, and more efficient land registry.

2 Existing Methodologies

Some organizations, across the globe, have initiated development of platforms and projects to effectively utilize blockchain for maintenance of land records. Propy [10] provides such a platform in the form of a website as well as distributed application (DApp). Testbed [11], part of a Swedish project is another solution for land record maintenance using blockchain.

Propy [10] platform unifies and standardizes property listing in global cities including USA, China, Russia, the Middle-East, and Europe. It automates the purchase process by online land ownership delivery and manages secure payment transactions. This process involves the buyer's broker as well as the seller's broker for interaction between the buyer and seller.

The testbed project [11] in Sweden includes private blockchain that can be run by public and private entities. It includes a software application that manages contracts controlled by and recorded on blockchain. The seller contacts a real estate agent and grants commission to the agent to sell the property via the software app to the interested buyer.

Contrary to Propy [10] and the testbed [11] project, the solution proposed in this paper enables direct buyer–seller interaction without external intervention. This eliminates the need of a government official or a broker and hence enhances transparency. The solution also facilitates the government bodies to access the history of a land property or the details of all the real estate a person owns.

3 Proposed Solution

The solution discussed in this paper aims at using blockchain, smart contracts, and a Web portal for real-time land record maintenance. Any registered user on the portal is allowed to buy plot on sale or sell his land or even tokenize the plot to be inherited by their successors. The seller and buyer will work with ether as their legal tender and all the transactions will be recorded on different blocks over the blockchain sequentially.

3.1 Stakeholders

Stakeholder may be a person, group of individuals, or organization that is interested to carry out the transaction. Important stakeholders included in proposed solution are as follows:

Government Initially, each and every unowned plot or land is registered under the government, with the land's location—co-ordinates and its area (per sq. ft.).

Buyer A registered user can act as a buyer if he intends to buy a land property using the Web app.

Seller A seller is the one who displays all the properties he owns and wants to sell the same. He will be notified if he has any requests from buyers for the property he has opted to sell.

Bank Institution The bank institution will be involved so as to verify the buyer's status and whether he can actually afford the property.

3.2 Registration

Registration is mandatory for a user to be able to buy or sell the property(ies). The user can also be the government who wants to sell land to an interested buyer. The following figure, Fig. 1, describes the process where every user creates an account

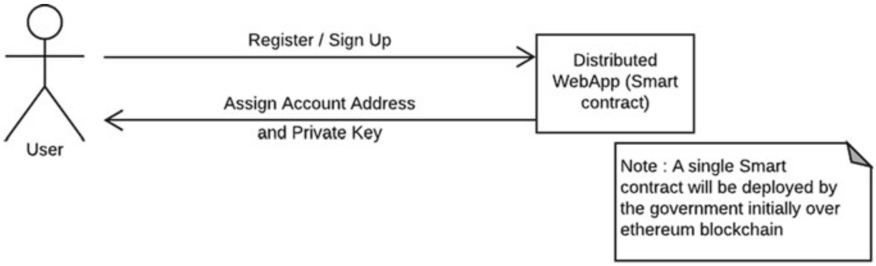


Fig. 1 Register/sign up

and gets a unique private key for further transactions.

Blockchain technology creates a unique address and key for each user of the website post registration. The user has to take a note of this address along with password for the next time when he wants to log into the application. This address also has a corresponding unique private key [4], which will be kept confidential. This private key will be used to sign the transaction(s). Initially, any unclaimed land plot will be registered under the authority of government.

3.3 Selling a Land Plot

A user can have multiple properties registered under to name out of which he might be interested to sell any number of properties—one or even all. The user (seller) is given an option to mark the property as sellable. The following figure describes the process of selling a land plot where buyer can send a request to seller and transaction proceeds only if seller accepts the request and buyer has enough ether (Fig. 2).

On logging into the application, the user puts on display all the properties he wants to sell by clicking on the sell option which will occur next to all the listed properties

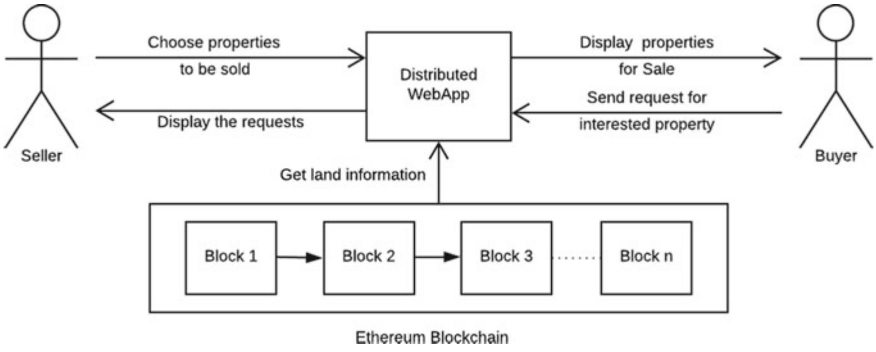


Fig. 2 Selling land plot

he owns. Seller can put the cost of his property in terms of digital currency ether, in this case, which is ethereum's own digital cryptocurrency similar to bitcoin. The trade that would take place if all attributes abide by the rules as per contract, and the seller will receive the ether for his property sold from the buyer.

3.4 Buying a Land Plot

All the saleable properties can be filtered according to the area the buyer wants to buy from. Buyer can view the co-ordinates of the plot, location, its price, and current owner of the plot. The following figure, Fig. 3, describes interactions involved in buying a land property.

The buyer sends request for the property he is interested in buying by using the GUI of the Web app using his digital signature. This assures that the seller does not repudiate the request made by the buyer. Seller then accepts or rejects the buyer's request. The smart contract, if the seller accepts, self-executes to check the required clauses and if all conditions are met, only then the approval is sent to the buyer along with money transfer request. The buyer provides affirmation using his digital signature after which he is notified regarding transaction completion and the land title is transferred to the buyer.

Tax which is paid to the government can be directly transferred from buyer's account to the government using the smart contract itself. This eliminates the government from direct intervention between the users, along with, getting the expected legal tax, and makes the whole system peer-to-peer.

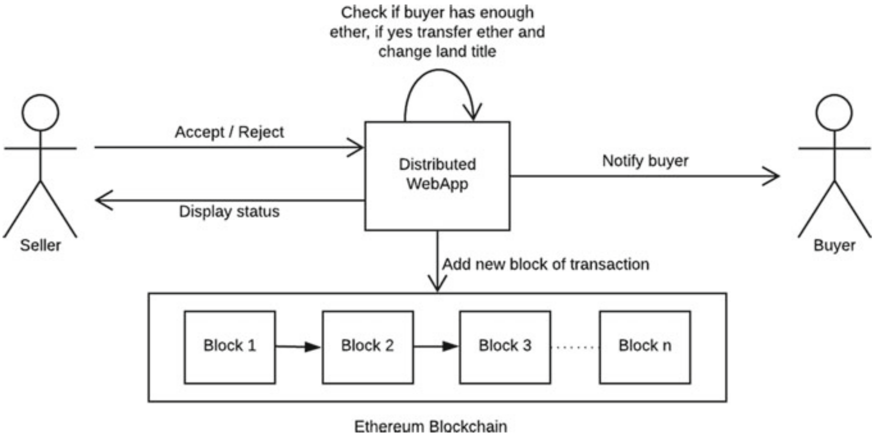


Fig. 3 Change of land title

3.5 Viewing Land History

Another functionality provided by the solution is for the government organizations who want to check details about the owner of a particular land property. They can view the history of owners of a land by entering the ID of that property. Also, the officials can check the list of properties currently owned by a particular user by entering the address key corresponding to the account of that user.

3.6 Using Polyline API

The plot or the land property can be mapped precisely using Polyline API from Google. This API gives the exact co-ordinates of the plot, thus calculating the area of the plot the user intends to sell. The location of the plot can also be easily determined anywhere on the globe, tracing the plot's co-ordinates. This provides to be a very strong verification feature and aims to solve issues related to property area.

4 Experimentation

The previous section gave an overview of the proposed system. The following section gives an idea about the actual implementation of the proposed solution so as to get a better understanding.

4.1 Setup

The following table shows comparison considering different parameters of the existing applications for the same purpose, i.e., for land record maintenance using blockchain (Table 1).

4.2 Implementation

The implementation is explained in three parts. The first part throws light on the back end of the proposed solution. The second part gives an idea about the front end or user interface of the solution. And the last part gives a brief about integration of the front and back end.

Back End
Remix [12] is the backend tool used, which is a powerful and open-source tool that

Table 1 Parametric comparison table

Sr. No	Platform	Involved stakeholders	Cryptocurrency/token	User interface	Backend framework
1	Propy	i. Users(buyer/seller/government) ii. Brokers iii. Notary iv. Banks	Propy Utility Tokens (PROs)	Website: ASP.NET 4, MVC 5, React.js	Truffle
2	Testbed	i. User(buyer/seller/government) ii. Real estate agent iii. Banks	Bitcoin	Distributed App(DApp)	Unknown
3	Proposed solution	i. User(buyer/seller/government) ii. Banks	Ether	Website: HTML, CSS, JavaScript	Remix IDE

helps to write Solidity contracts straight on browser. The huge advantage of using Remix is escaping all the installation hassle. It is written in JavaScript and supports testing, debugging, and deploying smart contracts. Since Remix is a Solidity IDE [13], it provides an interface to write code along with many features like syntax highlighting, auto-recompiling and auto-saving. Solidity is an object-oriented, high-level language used for implementing smart contracts.

The entire code for smart contracts in the proposed solution is written using Solidity in Remix. The code consists of two categories:

- **Users** for storing details like name, contact number, and number of properties the user owns along with the corresponding property IDs. Each user corresponds to a unique address assigned to him once registration is done.
- **Property** for storing details like area, price, current owner, and current status (unclaimed, pending, approved, rejected). Each property corresponds to a unique property ID. Each property also has a corresponding array for storing the owner history of that particular property.

Following are the functionalities for each user of the system:

- **The buyer** can search for a particular property by entering the property ID in the search box from all sellable properties on display. The '*SearchProp*' function will then search for that ID in the existing list of sellable properties. When the buyer is requesting seller for a property ('*RequestProp*') using his digital signature, a contract will get deployed. The buyer will have to pay a minimal amount for the same. After receiving approval from seller, the buyer initiates another function '*Title Transfer Request.*' This function will verify whether the address of message sender and buyer matches, post which a transfer of currency in the form of ether to the seller's account.

- **The seller** can change the price of owned properties by using '*SetPrice.*' This function checks whether the address of requestor and the address of owner are same and only then allows price to be modified. The seller has the choice to approve requests received by him for a particular property through the '*ApproveRequest*' function. After receiving request for title transfer, the '*Transfer Property*' function is initiated by the seller using his digital signature. This function checks if the message sender is the seller and if the ether balance in the buyer's account is sufficient. If not, then the bank balance of the buyer can be checked for verifying if the buyer can actually afford the property at that point of time. Once all the clauses are fulfilled, the transfer of ether takes place from buyer's account to seller's account and a proof of title transfer and transaction is sent to both users.
- **The government** has two additional functionalities. Along with acting as buyer or seller, the government can keep a strict eye on the properties that a user currently owns. This can be done using the '*CheckOwnedProperties*' function by passing the address of the user. This function will check if the sender's address is also the government's address. This will help make the government's tax procedure system more efficient and corruption-free. Another functionality of the government is '*CheckOwnerHistory.*' This feature allows government to get information of owner's history about a property.

Front End

The User Interface (UI) of website was created using Hyper Text Markup Language (HTML) and Cascaded Style Sheet (CSS). JavaScript (JS) was used for front end validations of input fields while creating an account.

5 Results and Discussion

From the implementation of our prototype, we found that, it can be efficiently used for land record maintenance with enhanced security using blockchain for storage of data. It also has an edge over existing technologies like Propy and testbed where broker was playing a major role in every transaction. Our prototype completely eliminates involvement of third-party and brings about complete transparency between buyer and seller. Hence, it results in cost reduction by cutting down the charges to be paid to the broker. Another crucial part in buying and selling off land is the area. Use of Polyline API provides added advantage for approximate calculation of area which can help in cross verifying area using the co-ordinates of mapped land pointed in an android app.

6 Conclusions and Future Work

From implementation of our prototype and study of existing methodology, we came to a conclusion that, nowadays, land records are still maintained in paper format or databases that can be easily tampered. The main advantages of this are enhanced security, elimination of brokers, transparency between buyer and seller and Polyline API for calculation of land area. Once the data is stored on blockchain, it becomes immutable. This can help in resolving issues related to ownership of a land in case of ambiguity and can help detecting false claims. Thus, our prototype can potentially replace not only physical data storage but also ensure high end security with increased transparency.

In future, this website can be converted into a DApp by using required technology. Once the DApp is created, the Polyline API can be integrated with it instead of using it. Also, the bank institutions which are presently checking the balance of buyer can be used to verify other clauses like the loan amount the account holder is already holding. Another advancement could be the prices of the properties automatically modifying with respect to the current market rate based on that location and area.

References

1. https://www.happiestminds.com/Insights/blockchain/
2. Morgen P (2017) Reinforcing the links of Blockchain. https://blockchain.ieee.org/images/files/pdf/ieee-future-directions-blockchain-white-paper.pdf
3. Pai R (2018) Digitization of land records: benefits for property owners and the sector [Online].https://housing.com/news/digitisation-land-records-benefits-property-owners-sector/. Last Accessed 10 Jan 2019
4. Zheng Z, Xie S, Dai H, Chen X, Wang H (2017) An overview of blockchain technology: architecture, consensus. In: IEEE 6th international congress on big data
5. Nakamoto S, Bitcoin: a peer-to-peer electronic cash system
6. https://blockgeeks.com/guides/ethereum/
7. Buterin V, A next generation smart contract & decentralized application
8. https://www.coindesk.com/blockchain-land-registry-solution-seeking-problem
9. Thakur V, Khadanga G et al (2003) Land records management system in India—technical framework. In: Map Asia conference 2003, land records information systems division national informatics centre, New Delhi
10. Team Propy (2017) Global property store with decentralized title registry
11. Lantmäteriet, Bank Landshypotek, SBAB, Telia company, ChromaWay, Kairos Future (2017) The land registry in the blockchain—testbed
12. https://remix.readthedocs.io/en/latest/
13. https://www.sitepoint.com/remix-smart-contracts-ethereum-blockchain/

Tenders 2.0 – A Stake-Based Blockchain Solution for Tender Procurement System

Pranamya Jain, Siddhesh Gangan, Siddhesh Rane, Yash Jain, and Dhiren Patel

Abstract Tendering is a procedure taken up by organizations for procurement of goods and services. Tender process when followed properly allows bids from multiple vendors and thereby facilitates healthy competition and fair outcome. Almost all the countries follow this process in standard and similar fashion. Lack of transparency in the tendering process and assignment of contracts to preferred agencies (rather than to a deserving bidder) has led to unwise use of money and the spirit of speculation. Blockchain is a decentralized ledger that makes records immutable, facilitates the secure exchange of digital currency, and performs deals and transactions. Each member of the blockchain network has access to the latest copy of encrypted ledger so that they can validate a new transaction. In this paper, we propose a blockchain-based tender procurement system with secure, fair, and reliable bidding architecture that can transparently manage the whole bidding process and perform an unbiased evaluation of bids. This system allows citizens to evaluate the process with a single click through an auditing application, which accounts for some stake to be put in the system against a false claim of task completion and thus actively help in making the democratic system stronger.

Keywords Blockchain and Distributed Ledger Technologies · Tender Procurement System · Smart contracts · Fraud Detection

1 Introduction

Public procurement processes are often complex and have a very limited transparency. It encompasses all purchases of goods and services by public institutions in the country. It also involves contracts between the government and the private sector in various areas like health services, military, construction, etc. Tenders of worth INR 82,433 crores were floated in 2012–13, INR 189,279 crores in 2013–14, INR 212735.5 crores in 2014–15, INR 404176.6 crores in 2015–16 and INR 543820.5

P. Jain · S. Gangan · S. Rane · Y. Jain (✉) · D. Patel
Veermata Jijabai Technological Institute, Mumbai, India
e-mail: yashjain864@gmail.com

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_18

crores in 2016–17 by government bodies. If summed up, a huge amount of INR 14 lakh crores have been spent in the last 5 years for which suppliers were selected through tender process. This constitutes about 30% of India's GDP every year [1]. Reality has dawned on the government that the suppliers in the country know each other and also the officials, so it is easy for them to take advantage of the system's weakness. As a result of misconducts, there are huge scams, suspicious modifications of tender requirements and deadlines, selection of unworthy bidders, etc. Also, the voices raised against such misconducts await terrible fates. Procurement accounts for a large part of public resources, and thus, it is important that the tender process occurs in an accountable, transparent and well-managed way. This paper introduces a new decentralized blockchain-based approach to make public procurement process transparent, free from misconduct, making it accountable and enhancing common people participation for uninterrupted and morally right completion of the contracts assigned.

Rest of the paper is organized as follows: In Sect. 2, generic tendering framework of government is discussed. Section 3 discusses blockchain and smart contracts. In Sect. 4, we give details of the proposed framework and architecture of blockchain-based tendering system. In Sect. 5, implementation details with algorithms of our framework are discussed with Conclusion and References at the end.

2 Generic Tendering Framework (of Government of India)

Here, we describe a general tender procurement framework that is in place currently [2, 3]:

1. Based on the requirements, the government releases the full tender specification through different mediums such as Web sites, newspaper ads, or any industry-relevant news media.
2. The tender specification includes various terms and conditions of the requirement, information necessary for an acceptable bid, and bid evaluation criteria.
3. The organization then hosts the tender specifications on a host.
4. The interested bidding organizations download the tender specification from the tendering host, review the requirements, and prepare for a bid.
5. The interested organizations then bid for the proposed tender before the submission deadline.
6. Submission of the bids would be open for a limited period, depending upon the tender specification.
7. When the deadline has passed, the tender host will shut down the bid submission portal. All bids received after this point would be rejected.
8. (a) Tendering organizations will evaluate all the submitted bids as per the evaluation criteria stipulated in the tender specification.

   (b) Based on the evaluation, the best bid would be selected and notified by the tendering organization.

From steps 1–8, citizens are not involved and have no visibility. However, after the tender is concluded, they can request for the data associated with the respective tendering process.

3 Blockchain Technology

Blockchain [4] is a distributed, tamper-proof digital ledger. Transactions are verified through consensus—participants confirm changes with one another and cryptography ensures the integrity and security of the information. This eliminates the need for a central certifying authority.

Transactions are broadcasted to a network of computers (point-to-point) each of which is called a node. The transaction as well as the status of users get validated in this network using the existing algorithms. Transaction which gets verified can consist of contracts, crypto token transfer, or other records and information. A block is created by combining verified transactions, comprising information for the ledger. In order to mine a block, the nodes must follow a consensus protocol (e.g., Proof of stake, Proof of work, etc.) depending on the underlying chain. Once this block is mined, it becomes immutable and is permanently added to the existing chain of the blocks. Each of these blocks contains a cryptographic hash that is linked to the previous one, resulting in the chain that is build using consensus mechanism and ensures integrity and security.

Smart Contracts

Blockchain and smart contracts go hand in hand just like the Internet and the email. A smart contract [5] is a code in the blockchain that stores rules of an agreement. It automatically verifies fulfillment and eventually executes the agreed terms. It eliminates the reliance of a third party when making an agreement. As built and implemented within a blockchain, they possess its immutability and distribution properties.

Smart contracts can be used by governments to ensure easy and efficient delivery of government services. Without the traditional means of government transaction, a citizen will be able to access quick and sufficient service delivery. The smart contract will also be able to reduce fraudulent activities in processes like tender because of the easy accessibility of the contracts for verification by any person.

Smart contracts can hold funds in escrow in case of transfer between addresses as they have state and memory storage. In a nutshell, smart contracts enable you to exchange money, shares, property, and many other things in a way which is transparent and peaceful without the requirement of a third party.

4 Proposed Framework and Architecture

4.1 Actors

- *Government/Organization*: The one who uploads tender specifications.
- *Bidder*: Interested organizations who desire to take up the task or provide the required service.
- *Contractor:* The bidder who is awarded the contract.
- *Verifier*: Person with specific domain knowledge related to the contract.
- *Participants*: Anyone in the network. Even a common citizen can be a participant.

4.2 Architecture

An organization or government creates a tender and puts it on immutable blockchain (see Fig. 1). Once the tender is up, prospective bidders can submit their bids within a specified deadline. At the time of bid, submission bidders will generate a key for symmetric key encryption, encrypt the bid with that key so that it cannot be viewed by others, and then submit the bid. Smart contract will continue to accept bids until the deadline. The bids, in encrypted format, will be stored on the chain itself. All the bids placed after the deadline would not be accepted. Once the deadline is reached, a key submission period is provided where the bidders will submit their keys so that the bids can be decrypted and viewed for evaluation. The valid bids will then

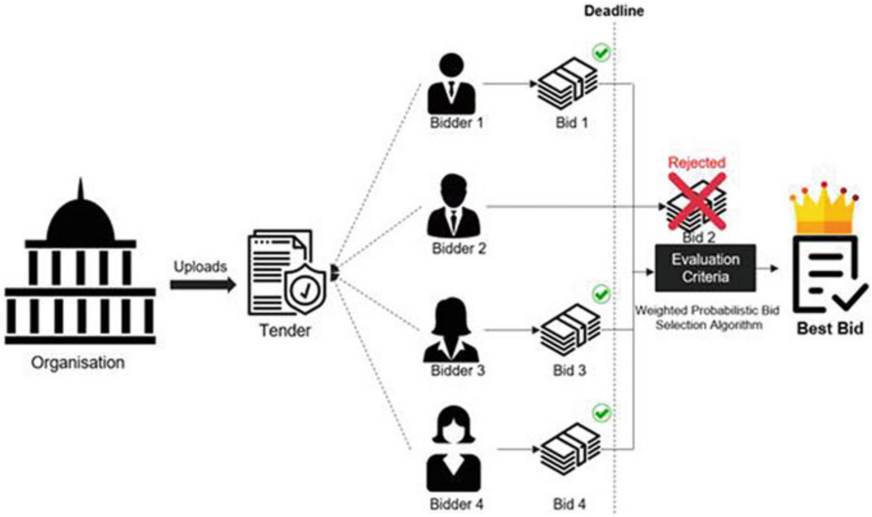
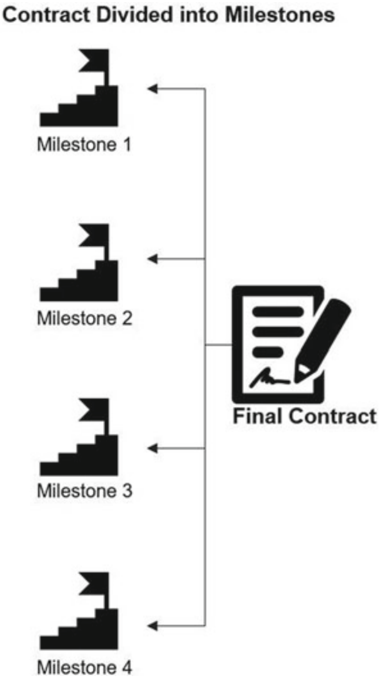


Fig. 1 Tender upload and bid selection

Fig. 2 Milestone creation



be evaluated based on the evaluation criteria specified by the organization at the time of tender creation. The selection of best bids will take place on the platform without human intervention. The bid fee, earnest money deposit (EMD) is refunded to the ones who are not selected. For the selected ones, EMD is accepted as a part of performance security in the form of a certificate of deposit.

Once the best bid is selected, a formal contract will be created on the blockchain. The contract would then get divided into set of milestones (see Fig. 2)—which are successive tasks that need to be completed incrementally in order to complete the contracted work.

Completion of each milestone will be verified in a two-step manner

- First, through the third-party verification done by anyone in the network having domain knowledge about the work and who is willing to put in specified stake.
- Secondly, by the officials whose stakes are automatically involved.

On getting successfully verified (see Fig. 3), the actual transfer of tokens allocated to a particular milestone will take place via smart contract on the blockchain, thus achieving continuous and timely payment to the contractor.

Same process follows for each milestone iteratively until all the milestones are accomplished and the task is completed.

People in the network will be monitoring each milestone completion and on detection of fraud, they can raise a claim with proof and small stake (see Fig. 4). Proof

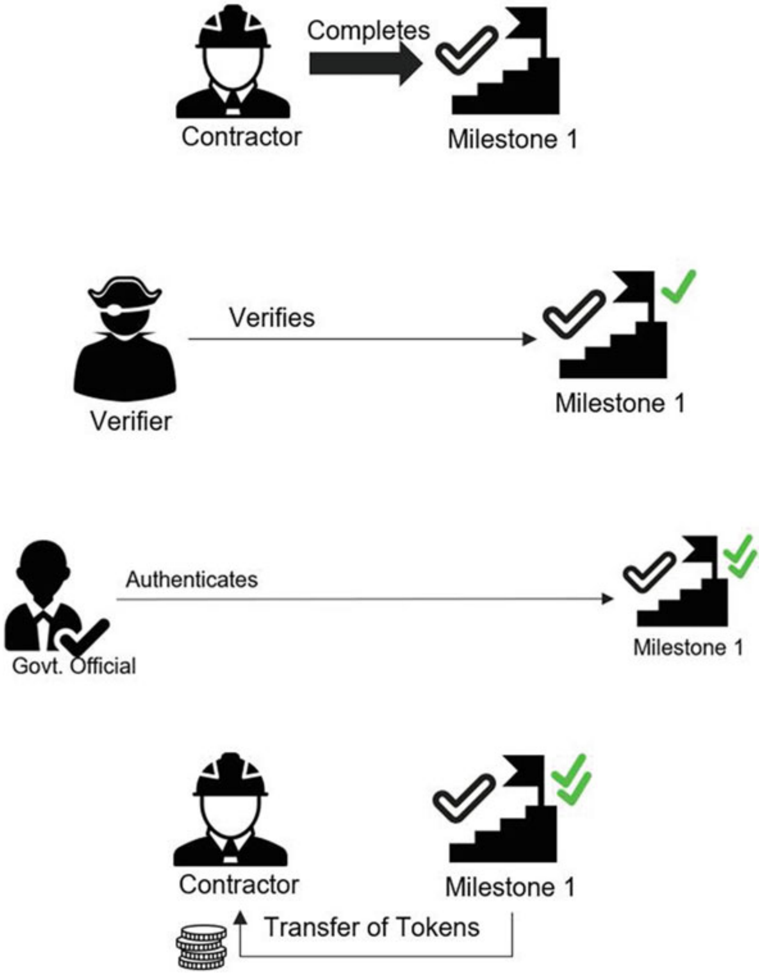


Fig. 3 Verification of milestones

can be submitted in the form of images, documents, statistics, or video supporting the claim. These proofs will be stored off-chain on IPFS [6] and the corresponding hash for retrieving the files will be stored on the chain.

The involvement of stake prevents spamming by malicious actors that deliberately raise false claims. In order to raise the claim, certain amount of stake has to be put. Also, for voting in favor of the claim, the citizen has to put certain stake. As stakes are involved, citizen is discouraged to cast a false vote in fear of losing his money. This claim will be pushed onto blockchain and can be viewed by other participants in the network. They can show their support by voting in favor of the claim and putting in certain amount of stake. Once the claim crosses a particular threshold amount, PIL would be filed and an official investigation would take place.

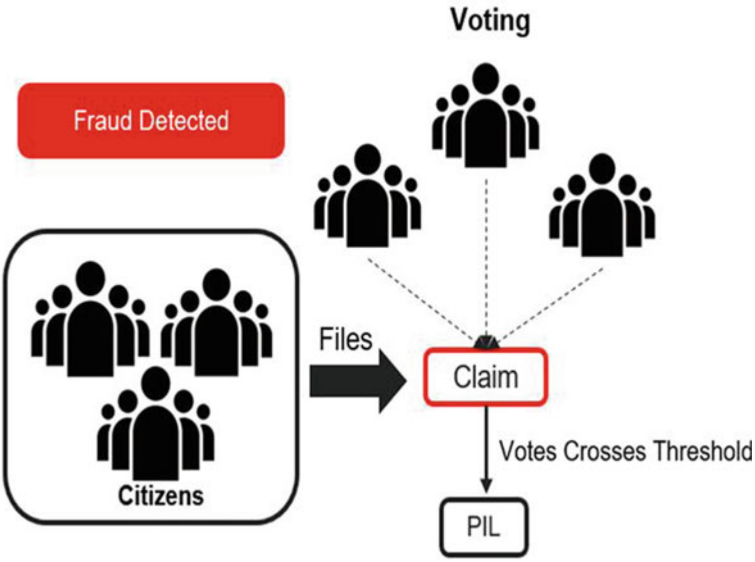


Fig. 4 Fraud detection

Based on the result of the investigation carried out by concerned department

- If the claim is proved, then the stakes of official and verifier would get slashed (see Fig. 5). The participants who raised the claim and those who supported it would

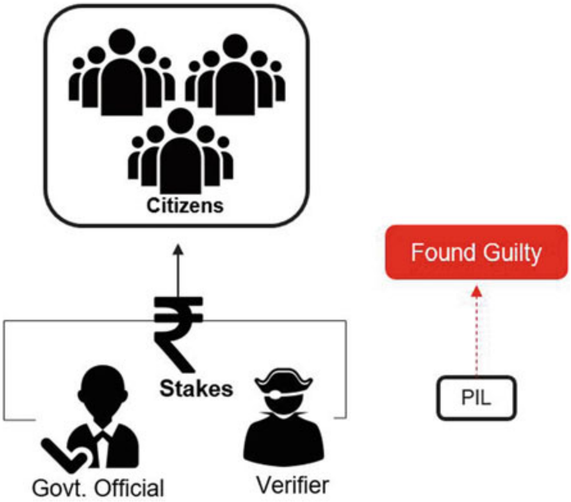


Fig. 5 Slashing of stakes

be rewarded in the form of tokens. The performance security of the contractor is deducted.
- If the claim is not proved, then the stakes of the participants who raised and supported the claim would be slashed and transferred to the government or organization. The government then uses this amount to mitigate the cost of investigation as well as rewarding the verifiers for their work.

5 Implementation and Validation

For implementing the tendering system, we have decided to use Ethereum [8] blockchain platform since it is open-source, massively adopted and has easy-to-implement APIs. We used Truffle [7] framework and contracts were written using Solidity [8] language. They were deployed locally using Ganache-client (private blockchain) [6]. Interaction of DApp with blockchain was done using Web3.js [9] (API to interact with blockchain) and Metamask.

Initiating a tender (Algorithm 1) is done by government official or organization which stores the tender details along with opening and closing date on the blockchain.

Algorithm 1 Creating A Tender

```
procedure CREATE TENDER
        (_data, _gvt_pbk, _bidOpenDate,  _bidCloseDate)
    data ← _data
    govtPublicKey ← _gvt_pbk
    bidOpeningDate ← _bidOpenDate
    bidClosingDate ← _bidCloseDate
```

The contractor places a bid (Algorithm 2) by transferring EMD, specifying the amount he is willing to pay for each milestone. Bids of each contractor are stored on the blockchain. Once deadline is reached, evaluation and selection of bids take place and best bids are selected. The best bid and evaluation criteria would be public, so that other bidders/citizens can verify it.

Algorithm 2 Placing A Bid

```
procedure PLACE BID
        (_contractor, _data, _milestone[], _EMD)
    contractorAdd ← _contractor
    data ← encrypt( _data )
    milestones ← _milestone
    contractorAdd.transfer(_EMD)
```

Creation of contract (Algorithm 3) takes place after selecting the best bid, thereby assigning the tender to the contractor and setting the start date of the contract as of now on the blockchain. The status of all the milestones of the contract is set as **PENDING**. The status of contract is set to **IN_PROCESS**.

Algorithm 3 Creating A Contract

```
procedure CREATE CONTRACT
        (_contractor, _tenderData, _start, _end)
    contractorAddress ← _contractor
    tenderData ← _tenderData
    contractStartingDate ← _start
    contractEndingDate ← _end
    contract.status ← IN_PROCESS
```

On completion of a particular milestone of a contract, the contractor marks it as completed (Algorithm 4) by uploading a proof of the same. Here, the status of the milestone is changed to **REPORTE_ COMPLETE**.

Algorithm 4 Contractor Marking Milestone as Complete

```
procedure MARK COMPLETE
        (_contractAddress, _proof, _milestoneID)
    milestone ← getMilestone(_contractAddress, _milestoneID )
    milestone.proof ← _proof
    milestone.status ← REPORTED_COMPLETE
```

If a domain expert wants to participate as a verifier (Algorithm 5), he can do so by staking a small amount for each milestone he verifies. This changes the milestone status to **PARTIALLY_VERIFIED.**

Algorithm 5 Verification of Milestone done by Domain Expert a.k.a Verifier

```
procedure VERIFY MILESTONE
        (_contractAddress, _milestoneID, _stake, _verifierAddress)
    milestone ← getMilestone(_contractAddress, _milestoneID )
    milestone.verifiedBy ← _verifierAddress
    milestone.verifiedBy.stake ← _stake
    milestone.status ← PARTIALLY_VERIFIED
```

If the government officer finds the work satisfying, he reports the milestone as **COMPLETE** (Algorithm 6) and transfer the amount associated with that milestone to the contract which holds the money till the contractor withdraws it. Thus, the two-step verification by verifier and government official has taken place.

Algorithm 6 Verify and Transfer Milestone amount to Contractor

```
procedure VERIFY TRANSFER
        (_contractAddress, _milestoneID, _govtAddress)
    milestone ← getMilestone(_contractAddress, _milestoneID )
    milestone.govtVerifiedBy ← _govtAddress
    milestone.status ← COMPLETE
    gvt_pbk.transfer(_contractAddress, milestone.amount)
```

The contractor can now withdraw the amount for the completed milestone (Algorithm 7). Once the contractor withdraws the funds, the status of the milestone is set to **CONTRACTOR_PAID**, thereby completing one milestone payment completion cycle

Algorithm 7 Withdraw Funds

```
procedure WITHDRAW FUNDS
        (_contractAddress, _milestoneID, _contractorAddress)
    milestone ← getMilestone(_contractAddress, _milestoneID )
    contract.transfer(_contractorAddress, milestone.amount)
    milestone.status ← CONTRACTOR_PAID
```

If a citizen finds misconduct in work done by contractor for a particular milestone, he can raise a claim (Algorithm 8) by staking certain amount and uploading proofs of the same. A new claim is raised for this milestone and status of the claim is set to **RAISED**. All the other citizens can now view this claim.

Algorithm 8 Raise New Claim

```
procedure RAISE NEW CLAIM
        (_contractAddress, _milestoneID, _citizenAddress, _proof, _stake)
    milestone ← getMilestone(_contractAddress, _milestoneID )
    supporter ← newSupporter(_citizenAddress, _proof, _stake)
    supporters ← [ ]
    claim ← newClaim()
    claim.status ← RAISED
    claim_supporter_mapping.put(claim, supporters)
    if milestone_claim_mapping.get(milestone) = NULL then
        claims ← [ ]
        claims.add(claim)
        milestone_claim_mapping.put(milestone, claim)
    else
        claims ← milestone_claim_mapping.get(milestone)
        claims.add(claim)
```

If a citizen wants to support a particular claim (Algorithm 9), he too can do so by staking certain amount, uploading his set of proofs, thereby incrementing the support vote count. Once the count crosses the threshold set during creation of the contract, a

PIL is filed by associated government Officer or organization and status of the claim is set to **PIL_FILED**.

Algorithm 9 Support Claim

```
procedure SUPPORT CLAIM
        (_contractAddress, _milestoneID,  _citizenAddress, _proof, _stake, _
                claimID)
    milestone ← getMilestone(_contractAddress, _milestoneID )
    claim ← getClaim(milestone, _claimID)
    claim.numVotes ← claim.numVotes + 1
    claim.amount ← claim.amount + _stake
    supporters ← claim_supporter_mapping.get( claim )
    supporter ← new Supporter(,  _citizenAddress, _proof, _stake)
    supporters.add( supporter )
    if claim.amount >= claim.threshold then
        claim.status ← PIL_FILED
        NotifyGovtOfficer()
```

If after investigation, a claim made by citizen is found to be true then the verifier's stake and government official's stake are slashed. This stake, in proportion, along with the original stake of each citizen is rewarded back (Algorithm 10) to the citizens. Even EMD of contractor is slashed and awarded to organization or government. The status of the claim is set to **CITIZENS_REWARDED**

Algorithm 10 Reward Citizens

```
procedure REWARD CITIZENS
        (_contractAddress, _milestoneID,  _claimID, _EMD)
    milestone ← getMilestone(_contractAddress, _milestoneID )
    claim ← getClaim(milestone, _claimID)
    supporters ← claim_supporter_mapping.get( claim )
    verifierAmount ← getStake(milestone.verifiedBy)
    govtAmount ← getStake(milestone.govtVerifiedBy)
    rewardAmount ← verifierAmount + govtAmount
    gvt_pbk.transfer(_EMD)
    for each sup in supporters do
        reward ← sup.stake + (sup.stake/claim.amount)*rewardAmount
        transfer(sup.address , reward)
        claim.status ← CITIZENS_REWARDED
```

If after investigation, a claim made by citizen is found to be false then the stakes of all citizens who supported the claim are slashed (Algorithm 11) which used to reward verifiers. The status of the claim is set to **STAKES_SLASHED**.

Algorithm 11 Slash Citizens stake who supported the false claim

```
procedure SLASH CITIZENS STAKE
        (_contractAddress, _milestoneID, _claimID)
    milestone ← getMilestone(_contractAddress, _milestoneID )
    claim ← getClaim(milestone, _claimID)
    slashAmount ← claim.amount
    transfer(milestone.gvt_pbk, slashAmount)
    claim.status ← STAKES_SLASHED
```

Once all milestones have been successfully completed and all claims have been resolved, the contract is successfully completed and its status is updated to **COMPLETE** (Algorithm 12). Once the contract is fulfilled, the EMD (performance deposit) is transferred back to the contractor. Both the verifiers are rewarded by the organization/government for the successful work.

Algorithm 12 Completion of Contract

```
procedure CONTRACT COMPLETION
        (_contractAddress, _contractorAddress, _verifierAddress,
_gvtAddress, _rwd1, _rwd2)
    gvt_pbk.transfer(_gvtAddress, rwd1)
    gvt_pbk.transfer(_verifierAddress, rwd2)
    transferEMD(_contractorAddress, EMD)
    contract.status ← COMPLETE
```

Helper Functions

Retrieve a Milestone of Contract

```
function GET MILESTONE(_contractAddress, _milestoneID)
    contract ← getContractByAddress(_contractAddress)
    milestones ← contract.tenderData.milestones
    milestone ← milestones.get(_milestoneID)
    return milestone
```

Retrieve a Claim of Milestone

```
function GET CLAIM(_claimID, _milestone)
    claims ← milestone_claim_mapping.get(_milestone)
    claim ← claims.get(_claimID)
    return claim
```

6 Conclusions and Future Directions

There is no accountability in the present tendering scenario for the contracts to be given and completed in a fair and transparent manner. Through this paper and model, we designed and proposed a blockchain-based tendering framework, in which deadlines and requirements cannot be altered, and submitted bids are immutable. These bids are fairly evaluated through algorithm, thus eliminating human-introduced frauds and errors. Milestones help in continuous payments and motivation for timely completion. Involving people through incentives helps in continuous monitoring of the system and also senses of satisfaction for people as they see and evaluate the use of their money. Penalization keeps frauds in check.

Some of the future developments include:

- Enhancement of evaluation and selection algorithm
- Insurance incorporation in the system

References

1. Public Procurement in India. Assessment of institutional mechanisms, challenges and reforms. https://www.nipfp.org.in/media/medialibrary/2017/07/WP2017204.pdf
2. Heeks R, Bailur S (2007) Analyzing e-government research: perspectives, philosophies, theories, methods, and practice. Gov. Inf. Q. 24(2):243–265
3. McDermott P (2010) Building open government. Gov. Inf. Q. 27(4):401–413. Special Issue: Open/Transparent Government
4. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system, [online] Available: http://bitcoin.org/bitcoin
5. Wood G (2014) Ethereum: A secure decentralised generalised transaction ledger. Ethereum Proj Yellow Paper 151:1–32
6. Benet J. IPFS—Content addressed, versioned, P2P file system (DRAFT 3)
7. Wimmer C, Wurthinger T (2012) Truffle: a self-optimizing runtime system. In: Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity. ACM, pp 13–14
8. Solidity documentation. http://solidity.readthedocs.io/en/latest/index.html
9. Web3.js documentation. https://web3js.readthedocs.io/en/1.0/

Current Indian Judicial System: Issues and Blockchain Solutions

Neerajkumari Khairwal and Ronak Shah

Abstract Indian Judicial System is the strongest institution of the world's largest democracy. Its purpose is to give justice to innocent and punish the guilty. Yet pendency of thousands of cases is the matter of serious concern. Shortage of judges and infrastructure has been highlighted as one of the major reasons for the same. Undeniably corruption and counsel-side delays are other prominent reasons for prolonged delay. To mitigate such delays, there is a need for more accountable and transparent judicial system. This paper aims to overcome the current loopholes in Indian Judicial System with an application of blockchain system. The paper discusses sample issues faced by common man in court along with the alternatives available and provides blockchain solution as the substitute. This approach implicitly limits the advocates, agents, and court officials to exploit judicial system for the benefit of individual and culprits. The proposed system will help regain the trust of common man in the Indian Judicial System.

Keywords Indian Judicial System · Blockchain

1 Introduction

Nowadays, Bitcoin is the latest buzz in the payment system and has been a huge success in financial services. It is a decentralized cryptocurrency designed for the electronic transaction by Satoshi Nakamoto in 2008. All the transactions are recorded in a distributed ledger called blockchain. Blockchain is a continually growing list of linked records secured through cryptography. It is immutable and persistent [1]. These features of blockchain have made it a suitable technology for other domains including Internet of things, health care, identity management system, real estate and many more.

N. Khairwal (✉)
Thakur College of Engineering and Technology, Kandivali, Mumbai, India
e-mail: neeraj_khairwal@yahoo.co.in

R. Shah
Tata Consultancy Services, Mumbai, India

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_19

Another significant yet ignored area of research is Indian Judicial System. Even though judiciary is a powerful tool, there are many loopholes in the existing system. All the players in the field, including judicial officials, advocates, police, and individuals, are exploiting these loopholes for individual benefits. It remains the very fact that, over a period of time, people have lost trust in Indian Judicial System. Pendency of cases for several years has been one of the major reasons for the same. A couple of crucial ingredients to stimulate the judicial process are power and money. A common man is generally devoid of both. Hence, Indian Judicial System is highly biased.

To overcome this sorry state of Indian Judicial System, blockchain is a very powerful technology. Inherent features of blockchain such as immutability can be used to make judicial process irreversible, undeniable, and unchangeable. This will expedite the judicial process to a larger extent. Officials will be accountable for any discrepancies. Secure transactions in the distributed ledger will reduce the malpractices in this field to a significant level. Persistency will help to store the record for a longer period of time and provide transparency. Accountability and transparency would hopefully expedite case proceedings, and hence reduce delay in closing cases.

2 Motivation

See Table 1.

Judicial system is the strongest institution in India. It not only provides the justice but implements the amendments and new laws passed by parliament. All citizens including actors, businessmen, politicians, saints and ministers come under jurisdiction and must abide the court order. Therefore, it is the most powerful system.

However, still crores of cases are pending in Indian courts. There are thousands of cases that are pending for more than 10 years [2]. It is the very fact that few people have lost their lives while fighting court case, struggled yet failed to get justice. Another important point to make note of is 80% cases filed are civil cases. Not surprising, then, World Bank's Ease of Doing Business ranks India at 77th position out of 190 countries [3] when it comes to enforcing contracts, which also includes quality of judicial process.

What is the reason that so many cases are pending in court? As per the report from Supreme Court and then Chief Justice, delay was due to an acute shortage of judges.

Table 1 Pendency of cases in Indian Judicial System [2]

Court (as on)	Pending cases		Cases pending for more than 10 years	
	Civil	Criminal	Civil	Criminal
Supreme Court (19 Feb 2016)	48,418	11,050	1132	84
High Courts (31 Dec 2014)	3,116,492	1,037,465	589,631	187,999
Subordinate Courts (31 Dec 2014)	8,234,281	18,254,124	611,658	1,432,079

This fact is supported by a Law Commission's report of 2009 stating, in Delhi High Court, 464 years would be required to clear the arrears with the present strength of judges [4]. However, can we say the shortage of judges is the only reason for delaying justice?

Let us dig more into it. Vidhi Centre for Legal Policy has conducted a study on 8086 orders passed by the Delhi High Court between 2011 and 2015. They have classified the cases as delayed if the cases have been pending for more than 2 years. As per Vidhi Centre, there are two perspectives for delay: Court-side delays and Counsel-side delays. The study concludes that 82% of the delays could be attributed to lawyers [5].

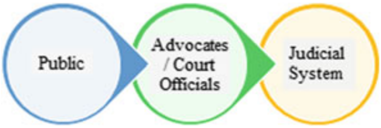
If we dig more into the ground reality, the current records including the statements and evidences get modified, updated, or even misplaced and deleted, for the purpose of delay and defeating the ends of justice. There is a very famous quote, 'Justice delayed is justice denied'. Means delayed relief is equivalent to getting no relief at all. Nevertheless, what's the way ahead? Can we deal with the core judicial issues to expedite the judicial process? Answer is yes. And it is possible through the emerging blockchain technology. Even if we are able to make 1% change in the existing judicial system, it will have social impact on millions of people. And people will actually start believing and regain faith in the judicial system. Thus, blockchain-based judicial system is the most prominent field to go ahead for discussion in this paper.

3 Existing Indian Judicial System

Judicial system is an important part of democracy and plays a crucial role in how democracy works. In current Indian Judicial System, there are two ways that a common man can approach court, viz. via advocates or through court officials as represented in Fig. 1.

In both the cases, legal procedure is an expensive affair. Common public has no or very limited knowledge about legal proceedings. They completely rely on advocates or court officials for any legal proceedings. There might be various reasons for same such as no legal knowledge, illiteracy, too busy or no interest to gain knowledge regarding legal process and so on. This blind trust on intermediaries might result in expensive dealings or in extreme case, cheating.

Fig. 1 General process to approach judicial system



4 Current Challenges

Below listed are the prominent challenges faced by current Indian Judicial System:

Accountability: None is held responsible for delay in current Indian Judicial System. Even though court officials might be justified in being overburdened, finally it is the common man getting pissed off.

Provenance: Provenance of data is lacking currently in Indian Judicial System. Originals can be easily forged and hence, questionable.

Transparency: Despite having privileges of certified copy for legal documents, Indian Judicial System lacks transparency mostly due to malpractices conducted by court officials.

Data Integration: Currently judicial officials have data records of only respective zones. Integration of criminal records across country will have better opportunity to track criminals and their past records to restrict unfortunate events in future.

Scalability: Current judicial system is not scalable enough to maintain a record of cases across the state.

In the Sect. 5, let us have an overlook of the proposed architecture for blockchain-based Indian Judicial System.

5 Proposed Indian Judicial System

Blockchain being an incorruptible distributed ledger technology can be used to overcome above-mentioned challenges in Indian Judicial System.

Current challenges highlighted in Sect. 4 can be tackled using blockchain-based Indian Judicial System depicted in Fig. 2. As a part of blockchain solution, police and lawyers will have to register through court official on blockchain. They will get walletAddress. Similarly, an applicant will have to register when he comes to court for the very first time. He will also get a walletAddress. Based on scenarios, these stakeholders can validate transactions in blockchain.

An applicant, through his lawyer, can file a petition through front end provided using web3.js. This will lead to saving the petition file on IPFS and generate IPFS hash for petition. It would trigger event for court official to file the petition by providing relevant details such as IPFS hash of petition, applicant, accused and lawyer details, type of case and sections charged. These details will get stored on blockchain and applicant will get case number, court number and date for proceeding. By using the smart contract [6, 7], event will be triggered to intimate accused regarding filed petition through e-mail as depicted in Fig. 3. Next section discusses few sample issues faced by common man in Indian Judicial System and how they can be restricted by use of blockchain technology [8–10].

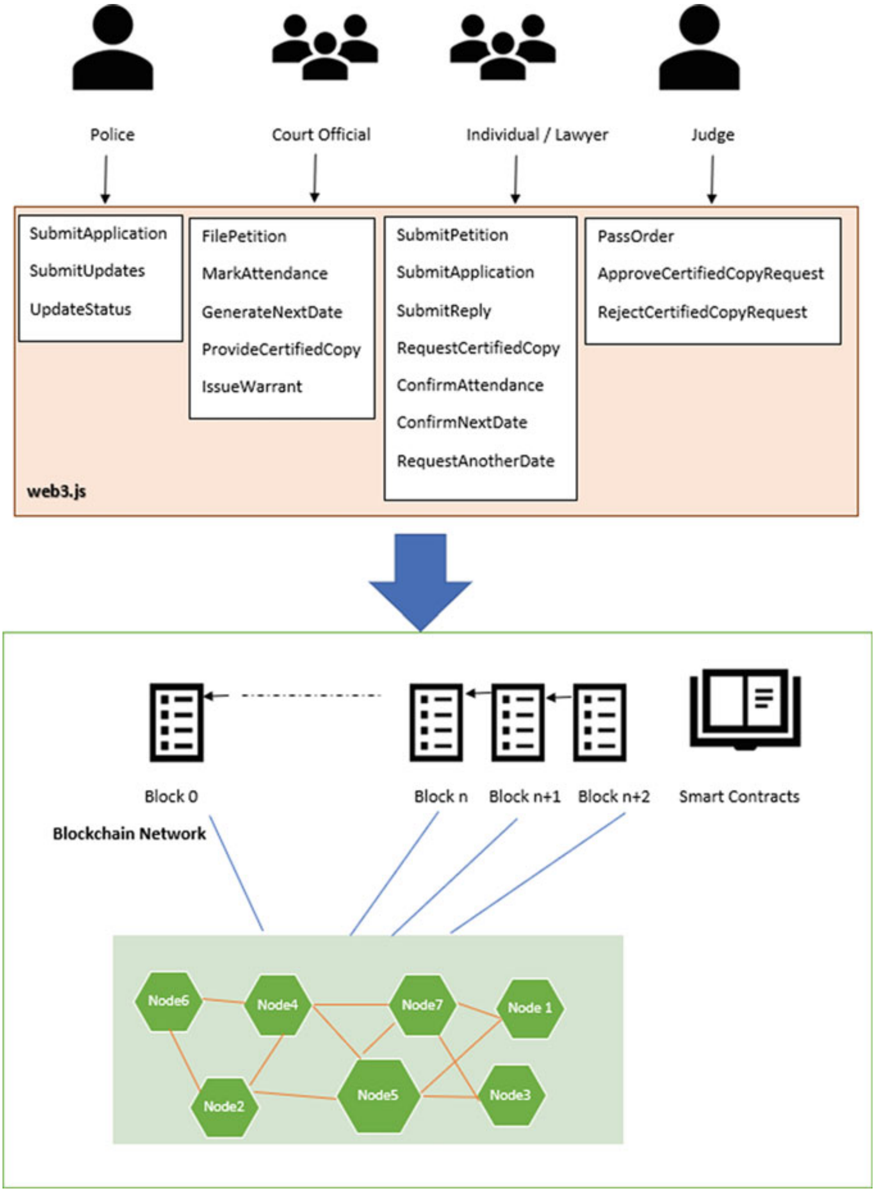
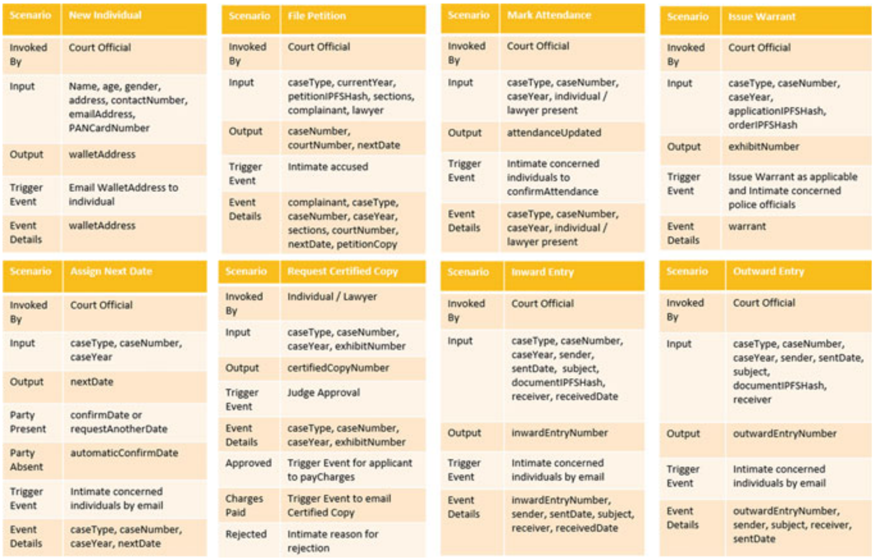


Fig. 2 Proposed Indian Judicial System



Scenario	New Individual
Invoked By	Court Official
Input	Name, age, gender, address, contactNumber, emailAddress, PANCardNumber
Output	walletAddress
Trigger Event	Email WalletAddress to individual
Event Details	walletAddress

Scenario	File Petition
Invoked By	Court Official
Input	caseType, currentYear, petitionIPFSHash, sections, complainant, lawyer
Output	caseNumber, courtNumber, nextDate
Trigger Event	Intimate accused
Event Details	complainant, caseType, caseNumber, caseYear, sections, courtNumber, nextDate, petitionCopy

Scenario	Mark Attendance
Invoked By	Court Official
Input	caseType, caseNumber, caseYear, individual / lawyer present
Output	attendanceUpdated
Trigger Event	Intimate concerned individuals to confirmAttendance
Event Details	caseType, caseNumber, caseYear, individual / lawyer present

Scenario	Issue Warrant
Invoked By	Court Official
Input	caseType, caseNumber, caseYear, applicationIPFSHash, orderIPFSHash
Output	exhibitNumber
Trigger Event	Issue Warrant as applicable and Intimate concerned police officials
Event Details	warrant

Scenario	Assign Next Date
Invoked By	Court Official
Input	caseType, caseNumber, caseYear
Output	nextDate
Party Present	confirmDate or requestAnotherDate
Party Absent	automaticConfirmDate
Trigger Event	Intimate concerned individuals by email
Event Details	caseType, caseNumber, caseYear, nextDate

Scenario	Request Certified Copy
Invoked By	Individual / Lawyer
Input	caseType, caseNumber, caseYear, exhibitNumber
Output	certifiedCopyNumber
Trigger Event	Judge Approval
Event Details	caseType, caseNumber, caseYear, exhibitNumber
Approved	Trigger Event for applicant to payCharges
Charges Paid	Trigger Event to email Certified Copy
Rejected	Intimate reason for rejection

Scenario	Inward Entry
Invoked By	Court Official
Input	caseType, caseNumber, caseYear, sender, sentDate, subject, documentIPFSHash, receiver, receivedDate
Output	inwardEntryNumber
Trigger Event	Intimate concerned individuals by email
Event Details	inwardEntryNumber, sender, sentDate, subject, receiver, receivedDate

Scenario	Outward Entry
Invoked By	Court Official
Input	caseType, caseNumber, caseYear, sender, sentDate, subject, documentIPFSHash, receiver
Output	outwardEntryNumber
Trigger Event	Intimate concerned individuals by email
Event Details	outwardEntryNumber, sender, subject, receiver, sentDate

Fig. 3 Blockchain judicial solutions

6 Judicial System Issues and Blockchain Solutions

Let us have a brief overview as to where and how the intermediaries have the chance to affect the judicial processes in the current scenario.

1. Next Date
2. Warrant Issuance
3. Certified Copy
4. Marking Attendance
5. Inter Department Communication
6. Regional Language Limitations

6.1 Next Date

General Process: As per the current judicial system, Hon'ble Judge assigns the next date for case proceedings. Generally, the court official, sitting next to the judge, notes down the next date on paper for respective case. Hon'ble Judge might also note down next date for respective case on his reference paper but there is no hard and fast rule. **Deviation**: Advocates of the court generally have a good rapport with court officials. Advocate can bribe the court official and ask him to pen down next date as per advocate's will and wish. For instance, Hon'ble Judge has given next date as 27th January for a case but court official has noted down next date as 15th May. Now, loophole in the current judicial system is there is no video recording or audio recording allowed in court. System reflects the next date entered by court official. Now as a victim, one might appear in court as per the next date given by Hon'ble Judge, i.e. on 27th January and find that the case is not on board that day. A victim or his advocate can appeal for 'hearing on board', i.e. hearing on the same day but is mostly rejected by Hon'ble Judge due to no substantial reason to take the matter on board. Rather case hearing is done on next date reflected in system, i.e. on 15th May. The victim is at loss. Very easily a delay of more than hundred days is introduced.

Alternatives: Victim or his advocate may challenge the discrepancies observed. But, generally government department supports its staff. In rare case, a departmental enquiry is initiated. Even though the enquiry process is carried out and the judicial official is found guilty, the enquiry is not closed. And, in Indian Judicial System, there is no time limit to close departmental enquiry. In the worst case, if the enquiry is closed and judicial official is not found guilty, the same person cannot file another complaint against the same judicial official throughout his tenure.

Blockchain Solution: Design an algorithm to generate next date for proceedings based on factors such as case type, sections, case age, number of cases pending in court, priority, individuals involved, holiday and judge vacation. Let the individual and/or lawyer, present in court, confirm next date. They can request for another next date in case of unavailability on assigned next date by providing considerable reason for unavailability. Intimate all concerned through e-mail regarding next date.

6.2 Warrant Issuance

General Process: In current court proceedings, judge issues an order for say recovery of fine. As per the process, judge passes an order against the application filed by complainant or respondent. The order simply states that based on given scenario, issue recovery warrant against the accused for recovery. This warrant needs to be issued by court official to concerned police as the case may be. Police needs to take appropriate action and report in considerable time for further action.

Deviation: Even though Hon'ble Judge has passed the order to issue recovery warrant, the court official does not issue the recovery warrant for months unless followed up by concerned individual or advocate. It keeps pending for several months if not traced. If issued, it is difficult to keep track as to when it was dispatched to police for further action. Also, police may delay the action due to several reasons thus defeating the purpose of justice.

Alternatives: Chase court official, court constable and concerned police officials for prompt action.

Blockchain Solution: For Indian Penal Codes section, under which a warrant is issued, there are pre-defined warrant format issued by the state government. Smart contracts can be developed that, on issuance of order, automatically generate the warrant through smart contract and add an action for concerned police blockchain. Police can submit the report to court through blockchain so that it cannot be modified later.

6.3 Certified Copy

General Process: If certified copy of any case related document is required by an individual, he needs to either request his advocate or apply for the same through agent. One can also apply for any certified copy on his own. The application needs approval from concerned Hon'ble Judge and only then the certified copy is issued.

Deviation:

1. In Indian Court, currently, charge for one page is ₹4. Therefore, for copy of an interim order, any application or a reply of 5 pages, ₹20 are to be paid. But advocates charge ₹500 minimum which is very costly. Similar is the case to get certified copy through agents. Positive part is, through advocates or agents, the certified copy might be available in a day or two based on the rapport of the advocate/agent with the court official.
2. Another problem faced is while providing a certified copy, a page or two might be skipped specially when the document requested is a large set. For example, a chargesheet. To get the missing pages, individual has to file a fresh application for certified copy.

Alternatives: On the other hand, if an individual applies for certified copy on his own then generally court officials are reluctant to entertain such applications. Even if they do, one needs to follow up for months to get a certified copy.

Blockchain Solution: Let the soft copy of all the legal documents filed in court be stored online using IPFS and hash of each document be stored in blockchain. Since an individual generally has access to all the documents filed in court for the respective case, except the Xerox submitted, provide an interface to apply for certified copies, pay online and an interface for Hon'ble Judge to approve or reject the request. On approval, e-mail the digitally certified copy to the individual. An individual can track the status online and neither he has to pay extra nor court official can delay it. Moreover, no possibility of any valid document being skipped in the certified copy since document's integrity can be verified using hash stored on blockchain.

6.4 Marking Attendance

General Process: In several Indian Courts, till today attendance of the complainant, respondent, advocates, public prosecutor, etc., for a case is marked on paper by court official, the case paper referred to as 'Roznama'. Court official simply writes today's date stating who all are present. A brief regarding today's proceeding and next date. This is later signed by concerned Hon'ble Judge.

Deviation: A court official, if bribed by an advocate, easily marks the absent as present by stating say for example accused no. 1, 2, 3, 4 are present even if accused no. 3 is absent. Since there are around 150–200 cases per day, it is not feasible for

the Hon'ble Judge to check whether the attendance marked, and entry made by court official is correct or not, they blindly sign it relying on court official.
Alternatives: If such a blunder is traced by individual, one can complain to concerned Hon'ble Judge or to higher authorities such as District Judge. In best scenario, even if one can prove the absence of the person marked present, the case papers are not updated, modified or rectified in any situation.
Blockchain Solution: Let the court official make entry for each individual present or absent in blockchain. And others present for respective case, confirm it.

6.5 Inter Department Communication

General Process: Documents being submitted by police are directly given to concerned court official without any means for concerned individual to trace. Documents from other departments, courts when received, an entry is made in Inward Register. Similarly, documents sent from court to police are directly given to court constable, whereas documents to other department are sent after an entry in Outward Register.

Deviation:

1. Documents given to or received from police officials may be missed or may not be traceable. Moreover, concerned individual has no knowledge regarding any progress in this front.
2. Concerned individual is not informed regarding any letter received in court regarding his matter. Concerned court official may not take necessary action for received letter.

Alternatives: Keep a check on Inward/Outward entry if any communication from other department is expected. This is generally not allowed for common public.
Blockchain Solution: Entry for any document related to a case should have an entry in blockchain. This will enable provenance and traceability. Concerned individuals will be informed. Officials accountable for delay can be held responsible.

6.6 Regional Language Limitations

Scenario 1
General Process: In State of Maharashtra, most of the documentation work by court officials is done in regional language Marathi. All the legal documents such as warrants are issued in Marathi.
Issue: The problem arises when inter-state communication is required. An order in Marathi is incomprehensible for officials in Kerala.
Alternatives: Beforehand, concerned individual needs to inform judicial officials to issue warrant in English.

Blockchain Solution: Since English is designated as the official language of the Government of India by the Constitution of India, English should be preferred for inter-state communication. By using event-driven smart contracts, this can be easily achieved.

Scenario 2
General Process: In State of Maharashtra, lower courts have court stamps available in regional language, i.e. Marathi. For any affidavit, if court stamp is required in English, process is that concerned official, for example, Police, needs to justify the need for English Stamp. The request for new stamp by concerned officials is forwarded from lower courts to District Court. After the approval of Principal District Judge, the request to create stamp is forwarded to Government Printing Press. Court official needs to collect the stamp from Government Printing Press.
Deviation: Concerned official, i.e. police may delay to put up the request to concerned Hon'ble Judge. After forwarding the request to District Court and Principal District Judge's approval, there may be delay in putting up the request to Government Printing Press. Finally, even though stamp is created by Government Printing Press, the stamp might not be collected by court official. Court official might be reluctant to visit Government Printing Press since they are not officially intimated of stamp being ready by Government Printing Press. Government Printing Press does not inform the concerned department as they handle state-level stamp request for all Government offices. Overall, this delays the entire process by several months or even year if not traced.
Alternatives: Individual needs to track the concerned police and court officials to check availability of new stamp. In the worst case, individual himself needs to visit Government Printing Press to ensure that stamp is ready. In any case, court official needs to visit Government Printing Press to collect the stamp.
Blockchain Solution: Provide an interface to raise request for new stamp. Let entry be made in blockchain for new request, Principal District Judge's approval, request to Government Printing Press, availability of new stamp and collection of stamp by court official. None can deny the fact that they were not notified or informed. Since blockchain is immutable, the concerned can be held accountable for delay if any.

Thus, blockchain can be used in an efficient way to improve the judicial process.

7 Conclusion

Judiciary is one of the strong pillars on which democracy stands. Pendency of cases for several years defeats the very purpose of justice. As rightly said, justice delayed is justice denied. This delay can be reduced to a considerable level by limiting the way current Indian Judicial System is exploited by advocates and court officials. Current Indian Judicial System lacks accountability and transparency. This can be efficiently catered by use of blockchain framework for Indian Judicial System. Immutability of blockchain paves a way ahead for tamper-resistant legal record management. None can modify, delete or add a new entry in past nor deny the existing ones. Cryptographic hash in blockchain enables the provenance of legal data. Streamlining the judicial process with blockchain makes the system more traceable and secure. This will benefit the society at large as none will have privilege to derelict the duty and misuse the judicial system for individual advantage. Regaining the trust of common man in Indian Judicial System will be the ultimate aim of the proposed blockchain solution.

References

1. Bashir I. Mastering Blockchain. Packt
2. Press Information Bureau, Government of India, Ministry of Law and Justice, Pending Court Cases (2016)
3. Press Information Bureau, Government of India, Ministry of Commerce and Industry, India Improves Rank by 23 Positions in Ease of Doing Business (2018)
4. Law Commission of India (2009) Government of India. Reforms in the judiciary—some suggestions
5. Nitin K., Shalini S, Sumathi C. (2017) Inefficiency and judicial delay. Vidhi Centre for Legal Policy
6. Peters G, Panayi E (2015) Understanding modern banking ledgers through blockchain technologies: future of transaction processing and smart contracts on the internet of money. arXiv: 1511.05740v1
7. Antonopoulos A, Wood G. Mastering ethereum: building smart contracts and DApps. O'Reilly
8. Zheng Z, Xie S, Dai H, Chen X, Wang H (2017) An overview of blockchain technology: architecture, consensus, and future trends. In: IEEE 6th international congress on big data
9. Dinh A, Liu R., Zhang M et al Untangling blockchain: a data processing view of blockchain systems. In: IEEE transactions on knowledge and data engineering
10. Koumidis K., Kolios P, Panayiotou (2018) Optimizing blockchain for data integrity in cyber physical systems. In: Proceedings of ICS & SCADA, BCS Learning and Development Ltd

Blockchain-Based Security for Super-peer Wireless Sensor Networks

Kushal Patil, Nirman Sonawane, Ekta Patil, Kshitija Kulkarni, and Puja Padiya

Abstract Wireless Sensor Networks (WSN) have now become an integral part of everyday human life. It constitutes the core of the Internet of Things (IoT) which has now become the most promising field for new innovations. WSN finds its application in a plethora of fields ranging from health care, home application, transport to forest monitoring, surveillance, and military as they are robust and can withstand harsh natural conditions. As the use of WSN increases in applications where confidential information is being transferred, the security of the system becomes a major concern. As the sensor nodes presently work on limited energy and computation power, the ways in which the system can be protected gets limited to a certain spectrum. This paper proposes an innovative decentralized approach for authentication in a super-peer network by the means of the blockchain. The blockchain is an open, distributed ledger that can record transactions efficiently and in a verifiable and permanent way. By allowing the information to be distributed but not altered, the blockchain has created the backbone for a secure decentralized network. Finally, the proposed system's performance will be checked by implementing the protocol in the AVISPA tool and checking the performance against various backend tools provided by the tool.

Keywords Blockchain · Decentralized · IoT · AVISPA · Super-peer · WSN

1 Introduction

Internet of Things is a connection of various types of devices that interact with each other over the internet and exchange data. IoT facilitates a more direct integration of the physical world with the internet. This results in efficiency improvement, monetary benefits, and reduced human efforts. Moreover, there is no restriction on the type of data that can be shared among various devices in IoT. For this purpose, Wireless Sensor Networks (WSN) are used. WSN has evolved enormously over recent years.

K. Patil · N. Sonawane · E. Patil · K. Kulkarni · P. Padiya (✉)
Ramrao Adik Institute of Technology, Nerul, Navi Mumbai, India
e-mail: puja.padiya05@gmail.com

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_20

WSN refers to a group of spatially dispersed sensors that send information about the physical conditions of the environment and organize the collected data at a central location. This data that is being transferred may sometimes contain some valuable information and as the applications of WSN increase, a major issue that arises is the security and privacy of the data collected from the nodes. Also, node failure and data manipulation are practical scenarios that can hamper the security and launch an attack on the system to access valuable information.

In scenarios where the information being transferred is not of much value, any node can easily send the data to the Cluster Administrator (CA) without any prior authentication. But since the information being shared holds some importance in the system, the nodes need to be authenticated prior to sending information.

This paper presents a new way of providing security to the WSN using blockchain. Blockchain found its limelight when it was initially used in the Bitcoin system. In simple terminology, it can be described as a chain of blocks where each block has a body and some header value. The body may consist of encrypted text or plain information. The header value is some information about the previous block. Hence, in this way, a chain of linked blocks is formed. There is a dedicated procedure for adding a new block to the system. This can be exploited to use in a WSN and can be used for providing authentication to the nodes. Following, in Sect. 2 some related works are studied to secure the IoT. In Sect. 3 the detailed operation of the system is explained using diagrams and algorithms. Expected results of the solution are drawn in Sect. 4. Finally, in Sect. 5 contains the conclusion and Sect. 6 contains future work.

2 Related Work

In this section, an overview of the attacks which can occur due to poor authentication measures is given, along with the traditional measures that are used currently and its comparison with the proposed system.

Commonly occurring authentication attacks are as follows:

- Man-In-The-Middle Attacks: in this attack, the attacker intercepts a communication between two systems.
- Sybil Attacks: a Sybil attack consists in that a single node presents multiple identities to other nodes.

Some traditional methods used for key management and authentication are as follows:

- Eschenauer and Gligor [1] propose a probabilistic key predistribution scheme. This scheme consists of three phases, namely, predistribution, shared key computation, and path key development. These stages authenticate the sensor nodes first and then set up the secret keys. This scheme faces problems like difficulty in signature key distribution over large networks and replication of revocation messages which hamper the security proposed by this scheme.

- Zhang et al. [2] proposed a GPSRReV scheme, which is also known as a centralized key revocation scheme. This scheme uses GPSR protocol, to broadcast messages of authentication within a particular range, these messages are then multicast to the nodes outside the range. As it is a centralized scheme, the drawbacks of centralized networks become applicable here. The attacker has a specific target for attack and if the central node gets compromised then the whole system is accessible to the attacker.
- Chan et al. [3] proposed a decentralized system based on votes. This system requires local broadcast and the nodes are authenticated by the neighboring nodes only when all the neighbors know about the new node. This is practically a complex task to be accomplished.

After a study of the papers related to providing security to the system by providing authentication measures, the following conclusions can be drawn:

- A complete model for accessing information, security, and privacy is given by none of the research papers [4].
- When a central authority in charge of authentication, it can be easily ensured that good security and privacy level is present in the network. This has a major drawback; the master authority becomes the central part of the network security and thus becomes a critical point of vulnerability in the network [5].
- Centralized systems may not support the flexibility required by multiple users with varied needs as it requires users to access information on the network uniformly using the same processes [6].
- Security is a critical component of any network system where the information being transferred is of critical importance. Centralized systems often produce centralized targets. Regardless of the security measures taken to protect this network, a centralized system is always easier to attack than a decentralized system as the attacks can be area specific.

Thus, according to the survey, there are various systems for providing authentication security to the network, but none of the methods have experimented with the decentralized architecture. By implementing a new security model and its protocol based on the blockchain technology, validity, and integrity of cryptographic authentication of nodes, from the beginning to the end of the sensor network lifetime.

The proposed system will overcome the shortcomings of the present systems in the following manner:

- This system uses the blockchain as the database to store public keys to achieving a higher security level. It uses digital signature and peer-to-peer information, to allow each Cluster Administrator (CA) of the network to validate information about every other node in the network.
- Only authenticated nodes can mine new blocks, and only if they are not the ones issuing a payload to be included in the block.

- The payloads present in each block of the blockchain are used as an indicator of the node behavior in the network over time. This way, it is ensured that a node cannot fool others by means of data tampering or masquerading.

Thus, the proposed system outlines how blockchain focuses on the usability of the blockchain as a secure decentralized data structure for providing node authentication and security enhancement in Wireless Sensor Networks (WSN).

3 Proposed System

This section describes the structure of the proposed system. It primarily describes the structure of the blockchain and the contents that will be stored in each block. Later, it describes the structure of the system and the algorithms which will be used for various processes.

3.1 Blockchain Architecture

See Fig. 1.

Blocks in the blockchain can be considered to be data structures whose purpose is to bundle a set of entities and distribute it to all nodes in the network. These blocks are created by miners. The miners in this scenario will be the Cluster Administrators (CA). Only those nodes which are validated by 51% or more CAs will be added to the network (Fig. 2).

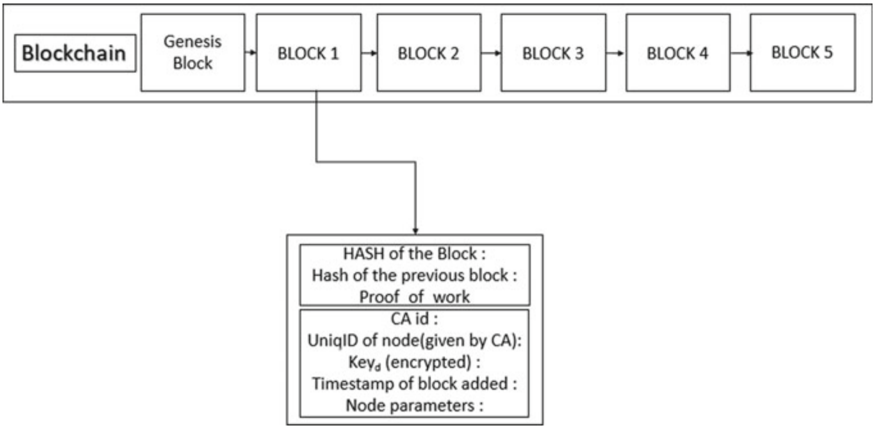


Fig. 1 Blockchain architecture

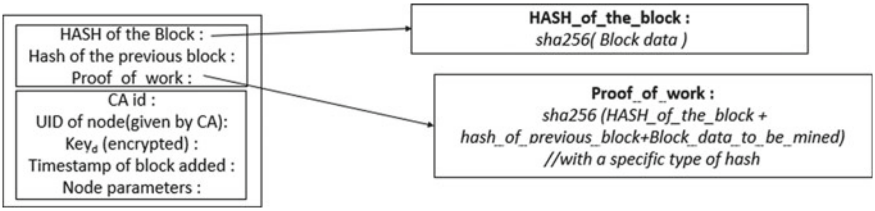

HASH of the Block :
Hash of the previous block :
Proof of work :
CA id :
UID of node(given by CA):
Key$_d$ (encrypted) :
Timestamp of block added :
Node parameters :
HASH_of_the_block :
sha256(Block data)
Proof_of_work :
sha256 (HASH_of_the_block + hash_of_previous_block+Block_data_to_be_mined)
//with a specific type of hash

Fig. 2 Block diagram

The block is divided into two parts:

(a) The Block constants and header
(b) The Data payloads.

The block constant consist of the hash value of the previous block, Merkle root value, timestamp and proof of work value. The Merkle root hash value will be changed and then invalidated even if a single change occurs in the block payload [7]. This solution thus provides secure and reliable storage distributed among all peers. Both the hash calculation algorithm and bandwidth are affected by the size of data payloads which are used to maintain the blockchain. Due to bandwidth restriction in WSN, the total block size is limited to approximately 5 MB so that enough storage can be ensured which is very important for security and avoiding overloading of the network with large size blockchain control data.

Blockchain was mainly used as a database for key management until recently. The proposed system exploits its use as a repository for storing the authenticated nodes and verifying their identity in case they migrate from one cluster to another. Moreover, the key imparted to the node will be stored in an encrypted form which can only be decrypted by using a two-factor authentication method. Hence, in this way, the key can be managed securely throughout the network of nodes and can be securely used for authorization. Thus, the proposed approach is an extension to the existing use of the blockchain and hence coexistence of the existing and the proposed system can be ensured easily.

3.2 System Architecture

This paper presents a new application using blockchain as secured decentralized storage for cryptographic keys and also issues on trust information with respect to Wireless Sensor Networks [8]. Blockchain Authentication describes a way to use the immutability feature of the blockchain to provide solutions to highly complex problems in the field of decentralized ad hoc networks [9]. More accurately, it is shown how a complete solution can be built providing authentication mechanisms and trust evaluation in a self-organized and evolutionary network.

In this paper, WSN is represented by a set of clusters of limited capacity nodes each cluster is managed by an unlimited one, which is the CA (Cluster Administrator). There will be mutual authentication between the device and the CA belonging to the same cluster using personalized shared keys.

The blockchain ledger will contain information about all the nodes in the network so that each can validate information about different nodes in the network through the ledger. At the beginning of the network when there is no authenticated node, the chain is empty. The genesis block should be designed and mined very crucially. The values of the first block that will be mined contain all the mutable values and data because there is no other block to validate the only block in the ledger (Fig. 3).

When a new node enters a cluster and wants to get associated with the network, the following communication will take place between the CA and the node:

- The node sends an association request to the CA.
- CA sends a problem to be solved by the node.
- CA solves the problem and sends the computed value to the CA.
- CA itself solves the problem and compares the computed value with the value obtained from the user.
- If these values are equal, the node is considered to be safe to be added to the blockchain. If not two more attempts are given to the node to compute the solution, if the node fails three times, it is considered to be malicious and is added to the blacklist.

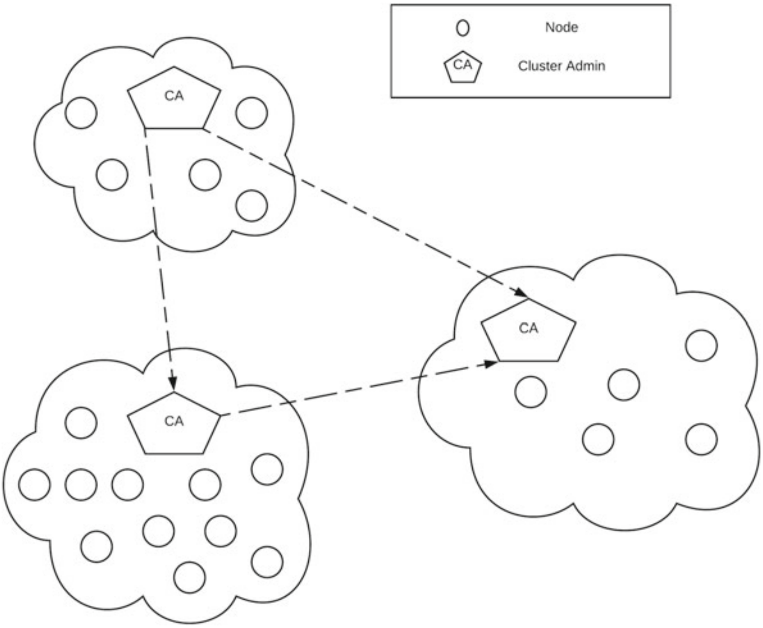


Fig. 3 Super-peer WSN Topology of a cluster of nodes

- If the node is safe, the CA imparts a unique id to the node.
- The node is then added to the CA authenticated list and is broadcasted to the other nodes in the network.

This blacklist is public and available for all the CAs. Hence, once a node is added to this list it cannot be authenticated at any other CA in the entire network.

After the node is considered safe, it can be added to the blockchain. The information to be added is the details of the node, which is the unique id given by the CA. The following steps occur during this process:

- Start with nonce 0, hash the block's header using SHA-256 and check if the hash is under the target value.
- If no, then increment the nonce value and calculate the hash value again until a value under the target is obtained.
- After the target is obtained broadcast the calculated header for validation by other CAs.
- Every CA in the network will verify the header and only then will it approve the association.
- Once 51% or more CAs approve the association, the block is mined and gets added to the network.
- After the block is added the CA sends its unique ID to the node in an encrypted manner.

Suppose, a previously authenticated node wants to migrate to another cluster, then there is no need for the node to be authenticated separately in the cluster. The system will work in the following manner, similar to the algorithm proposed by Hammi et al. [10]:

Suppose, we have a device a1 belonging to the cluster managed by the CA A,

- The initial association of a1 with A is established in a secure procedure.
- When a1 migrates to the cluster that is managed by B,

  i. It sends an association request to B.
  ii. In this request, a1 sends the encrypted unique id obtained from A.
  iii. B decrypts the value and checks it with the list of CAs.

- If this unique id is found in the ledgers of more than 50% of CAs, then the id is considered to be legitimate,

  i. B checks which CA is the parent of this node and the request for sharing the symmetric key (key_d), for authenticated encryption of data between CA and node.
  ii. B checks validity of the node by sending it a verification message encrypted with the key.
  iii. If the node is previously authenticated, it will have the symmetric key to decrypt the verification message and revert it back to the new CA, where the message would be validated.

This way the authentication is done in the network and it ensures that no malicious node can harm the data of the nodes.

4 Implementation

The system will be simulated in AVISPA. AVISPA stands for Automated Validation of Internet Security Protocols and Applications. The AVISPA Tool provides a suite of applications for building and analyzing formal models of security protocols. Protocol models are written in the High-Level Protocol Specification Language or HLPSL. The HLPSL language is used for describing security protocols and specifying their intended security properties, as well as a set of tools to formally validate them (Fig. 4).

The HLPSL language is role-based, which infers that the roles of the entities that will be participating in the implementation of the protocol will be described. The sessions in which the entities will communicate also needs to be specified. The knowledge of each entity consists of the data that is available with the entity prior to the execution of the protocol. This includes the public keys, private keys, symmetric keys and the data present with the entity that it wishes to share with the other entities. The code written in HLPSL language is then translated into the Intermediate Format (IF), using hlpsl2if. The model-checkers then process IF specification to analyze if the given protocol is satisfying the security goals. Four different verification back end tools are used to analyze the IF specification namely CL-AtSe (Constraint-Logic-based Attack Searcher), OFMC (on-the-Fly Model-Checker), SATMC (SAT-based Model-Checker), TA4SP (Tree Automata-based Protocol Analyzer) [11]. These back end tools are used to check for the different types of flaws that may be present in the protocol. The SPAN tool is an animator tool for AVISPA which gives a visual representation of the execution of the protocols. It also shows the visual simulation of an attack that can be performed on the system if the protocols are deemed unsafe by any of these back end tools.

This combined protocol was divided into two parts while specifying using the HLPSL language. These are as follows:

- Initial authentication
- Migration.

The initial authentication protocol and the migration protocol were written in HLPSL language and their performance was checked using the various backend tools provided by the software. In the initial authentication protocol, there are four entities at work, namely, the new node, the Cluster Administrator (CA) of the cluster in which the node enters and the other cluster administrators are considered as a single entity as their role is similar while this protocol is implemented. K_{nc} and K_{cc} are the symmetric keys for encrypted communication between the new node and the CA, and the CA and the other cluster admins, respectively. The goal of this protocol

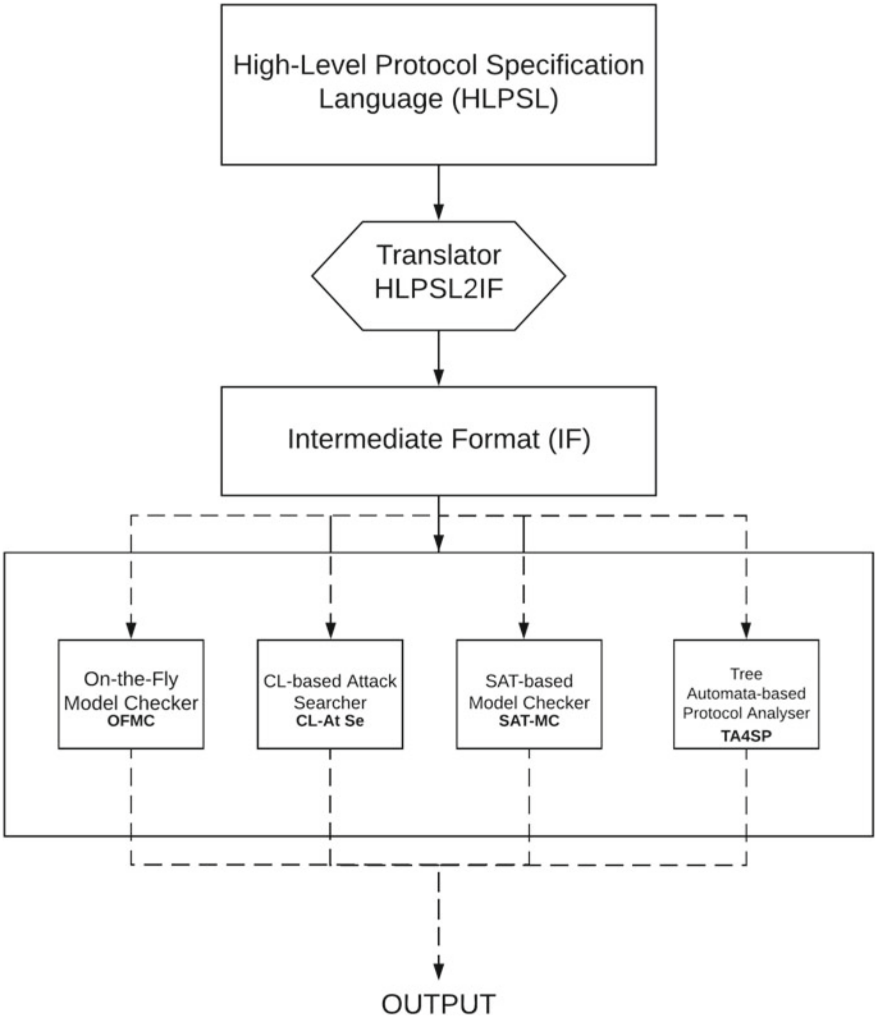

High-Level Protocol Specification Language (HLPSL)
Translator HLPSL2IF
Intermediate Format (IF)
On-the-Fly Model Checker OFMC
CL-based Attack Searcher CL-At Se
SAT-based Model Checker SAT-MC
Tree Automata-based Protocol Analyser TA4SP
OUTPUT

Fig. 4 Structure of the AVISPA tool

is that the CA securely authenticates the new node on the unique ID generated by the CA.

In the migration protocol, the entities consist of a node that has migrated from one cluster to another, a CA of the new cluster in which it has entered, a CA of the cluster in which it was previously present and the blockchain. Although being a ledger to store authenticated nodes in the actual scenario, it has been considered as an entity in this specification as, during the execution of this protocol, the role of the blockchain is passive, i.e., it is only used to check and retrieve the details of the node that wishes to migrate from one cluster to another. There is no modification or change occurring in the blockchain while this protocol is being executed. Thus, the role of the blockchain is similar to that of an entity participating in the protocol execution. K_{nb}, K_{bp}, and K_{BC} are the symmetric keys used for encrypted communication between the node and the new CA, the new CA and the previous parent CA, and the new CA and the blockchain, respectively. The goal of this protocol is to establish and maintain the secrecy of communication between the node and the new CA.

The HLPSL code of both protocols has been provided in the Appendix.

For validation purposes, a module of the above-mentioned protocol was implemented in python 3. For this purpose, various modules and environments are created. For each use case, a separate module is created. This was done to provide a better understanding of the working of the system. Therefore, there are basically five different modules for blockchain implementation purpose:

- Node Management Module
- Blockchain Ledger Management Module.
- Hash Module
- Transaction Module
- Verification Module.

For public and private key management, CRYPTO RSA Package was used. Every CA gets a private and public key, which is generated by this module.

For unique identification of each node, UUID4 module was used. Every node when it sends an association request is given a unique ID by which it is identified on the network. A universally unique identifier (UUID) is a 128-bit number used to identify information in computer systems.

When generated according to the standard methods, UUIDs are for practical purposes unique, without depending on their uniqueness on a central registration authority or coordination between the parties generating them, unlike most other numbering schemes. While the probability that a UUID will be duplicated is not zero, it is close enough to zero to be negligible [12].

For Managing THE PKI, a virtual environment is created in ANACONDA for CRYPTO packages.

All the hashes are performed using SHA-256 bit algorithm. SHA-256 is a novel hash function computed with 32-bit words. It uses shift amounts and additive constants.

The signing of each block is done by CRYPTO.SIGNATURE PKCS1_v1_5 module. The PKCS #1 standard defines the mathematical definitions and properties that RSA public and private keys must have.

The Security layers of Blockchain are designed as follows:

- Each block knows each other and performs a basic block manipulation check after every new block/ledger is mined and distributed across the network.
- Mass-Manipulation of the block is not possible as mined block require a valid proof of work to be added and verified in the ledger.
- For preventing the manipulation of node data while the mining phase, the node transaction need to be signed using PKCS1_v1_5 module.

Table 1 Simulation results of initial authentication protocol

Backend tool	Result	Description
OFMC	SAFE	DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/hlpslGenFile.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00 S searchTime: 0.16 S visitedNodes: 38 nodes depth: 8 plies
CL-AtSE	SAFE	DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/InitialAuth.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 21 states Reachable : 12 states Translation: 0.04 S Computation: 0.00 S
SATMC	INCONCLUSIVE	DETAILS ERROR
		PROTOCOL hlpslGenFile.if BACKEND SATMC

(continued)

Table 1 (continued)

TA4SP	INCONC LUSIVE	DETAILS: NOT_SUPPORTED PROTOCOL: /home/span/span/testsuite/results/hlpslGenFile.if GOAL: SECRECY BACKEND: TA4SP COMMENTS: For technical reasons about non-left-linearity in term rewriting with tree automaton, this protocol cannot be checked. Sorry. STATISTICS: Translation: 0.02 S

5 Results

Oracle VM box was installed in order to use the SPAN AVISPA tool. The backend tools provided by AVISPA were used to find traces of any possible attacks on the system, and there simulation if any attack was possible.

The following results were obtained when the above-mentioned protocols were tested against the four backend tools provided by the SPAN AVISPA:

(a) Initial Authentication Protocol (Table 1).
(b) Migration (Table 2).

The above results are expected to be obtained because of the following reasons:

- Blockchain uses block creation cost as a safety measure against attacks by using mining. While calculating proof of work, other central administrators are also involved. Thus, for a malicious node to assume various identities at once is a computationally complex task and will require high processing power which is presently absent in wireless networks.
- The system has an inbuilt trust model wherein central administrators with a minimum number of malicious nodes have higher trust value while validating a block. Thus, this unequal reputation among the nodes decreases the shareholding powers of those administrators with more number of malicious nodes.

Thus, the simulation results show that the proposed protocol is safe. The system can be considered for use when the nodes transfer confidential information in a

Table 2 Simulation results of migration protocol

Backend Tool	Result	Description
OFMC	SAFE	DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/hlpslGenFile.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00 S searchTime: 1.36 S visitedNodes: 386 nodes depth: 12 plies
CL-AtSE	SAFE	DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/hlpslGenFile.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 8080 states Reachable : 1360 states Translation: 0.06 S Computation: 0.15 S
SATMC	SAFE	DETAILS STRONGLY_TYPED_MODEL BOUNDED_NUMBER_OF_SESSIONS BOUNDED_MESSAGE_DEPTH PROTOCOL hlpslGenFile.if GOAL

(continued)

Table 2 (continued)

		%% see the HLPSL specification.. BACKEND SATMC COMMENTS STATISTICS attackFound false boolean stopConditionReached true boolean fixedpointReached 10 steps stepsNumber 10 steps atomsNumber 0 atoms clausesNumber 0 clauses encodingTime 0.05 S solvingTime 0 S if2sateCompilationTime 0.15 S
		ATTACK TRACE %% no attacks have been found..
TA4SP	INCONCLUSIVE	DETAILS: NOT_SUPPORTED PROTOCOL: /home/span/span/testsuite/results/hlpslGenFile.if GOAL: SECRECY BACKEND: TA4SP COMMENTS: For technical reasons about non-left-linearity in term rewriting with tree automaton, this protocol cannot be checked. Sorry. STATISTICS: Translation: 0.03 S

WSN. Hence, in this way, the system provides a transparent yet more secure system compared to the existing systems.

6 Conclusion

This paper presents blockchain as secured decentralized storage for cryptographic keys and also trust information. This system brings out blockchain's various features like flexibility, storage capacity, and immutability for security mechanisms in Wireless Sensor Network (WSN). In Blockchain, Immutability is a feature that maintains and secures data forever which turns out to be crucial as it solves various problems in the field of decentralized ad hoc networks. This system focuses on both the security as well as the authentication of the nodes in the network, which is very crucial to avoid many types of attack which are possible on the WSN. Thus, this system shows how it will be possible to build a complete solution that will provide an authentication mechanism and security enhancement in a network.

7 Future Work

The size of the network in the proposed model is small which impacts the performance of the system and therefore, new approaches in node management can be a solution. The scalability of the system can also be improved in the future. As the proposed system presents blockchain-based authentication on a local system, it can be revolutionized to a global level system. The new approach for mining of the first block can be a step further in improving the overall system. The network nodes' attributes can be better defined by new approaches in the future.

References

1. Eschenauer L, Gligor VD (2002) A key-management scheme for distributed sensor networks. https://dl.acm.org/doi/proceedings/10.1145/586110
2. Zhang W, Song H, Zhu S, Cao G (2015) Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks. In: MobiHoc '05: proceedings of the 6th ACM international symposium on mobile Ad Hoc networking and computing. ACM Press, New York, NY, USA, 378–389
3. Chan H, Perrig A, Song D (2003) Random key predistribution schemes for sensor networks. In: Proceedings of IEEE symposium on security and privacy. Oakland, CA, USA, 197–213
4. Medaglia CM, Serbanati A (2010) In: Giusto D, Iera A, Morabito G, Atzori L (eds) An overview of privacy and security issues in the internet of things. Springer New York
5. Article on authentication techniques for wireless sensor networks. Retrieved October 26, 2018, from https://pdfs.semanticscholar.org/698b/a57e7674053517f60fd936d8484c742efc29.pdf

6. Patil S, Kumar V, Singha S, Jamil R (2013) A survey on authentication techniques for wireless sensor networks. Int J Appl Eng Res 7(11). ISSN 0973-4562
7. Chowdhury AR, Chatterjee T DasBit S (2014) LOCHA: a light-weight one-way cryptographic hash algorithm for wireless sensor network. Procedia Comput Sci 32:497–504
8. Moinet, A., Darties, B., Baril, J.L. (2017) Blockchain based trust & authentication for decentralized sensor networks. arXiv:1706.01730v1 . [cs.CR]
9. Jesus EF, Chicarino VR, de Albuquerque CV, de Rocha AA. A survey of how to use blockchain to secure internet of things and the stalker attack
10. Hammi MT, Bellot P, Serhrouchni A. BCTrust: a decentralized blockchain-based authentication mechanism
11. Vigan‘o L (2006) Automated security protocol analysis with the AVISPA tool. Electron. Notes Theor. Comput. Sci. 155:61–86
12. Article on universal unique identifier, from https://en.wikipedia.org/wiki/Universally_unique_identifier

Author Index

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9