An Overview of Blockchain-Based Smart Contract

Satpal Singh Kushwaha and Sandeep Joshi

Abstract This work is motivated by the recent flare-up in the rapid technological developments of the evolving blockchain technology. Blockchain technology provides a mechanism to do transactional communication in a trustless manner, without the need of a trusted third party. An appealing feature or application of blockchain technology is a smart contract. The smart contract is a piece of code which executes on the blockchain technology to enforce the negotiation between two parties in the absence of a trusted third party. This paper aims to provide an ample overview of blockchain-based smart contract, starting with the introduction, architectural overview, working mechanism, application areas and research gaps which can be addressed in future research.

Keywords Blockchain · Smart contract · Distributed ledger · Etherum · Decentralized autonomous organization (DAO)

1 Introduction

Blockchain technology has gained a lot of attention from researchers and stakeholders from around the world after the inception of Bitcoin, [1] which has later become a revolutionary technology. Bitcoin, an application of Blockchain is a well-known term for cryptocurrency that is administered by users across the globe in a trustless manner without the need of a trusted third party.

Smart contract is the another famous use case of Blockchain technology, which was first introduced by Nick Szabo in 1994 [2], is a self-executing contract with the terms of the agreement in between untrusted agreed bodies and is a piece of program code to be executed when certain conditions have been met. The paper is structured in different sections as follows. Section 2 gives a brief introduction of

S. S. Kushwaha (✉) · S. Joshi
Department of Computer Science and Engineering, Manipal University Jaipur, Jaipur, India
e-mail: singh_satpal25@rediffmail.com

S. Joshi
e-mail: sandeep.joshi@jaipur.manipal.edu

S. Smys et al. (eds.), *Computer Networks and Inventive Communication Technologies*,
Lecture Notes on Data Engineering and Communications Technologies 58,
https://doi.org/10.1007/978-981-15-9647-6_70

Blockchain Technology, Sect. 3 describes briefly the smart contract, Sect. 4 describes development platform, Sect. 5 gives a detailed introduction of application areas of smart contract, Sect. 6 describes the research issues in smart contracts, and Sect. 7 concludes the paper.

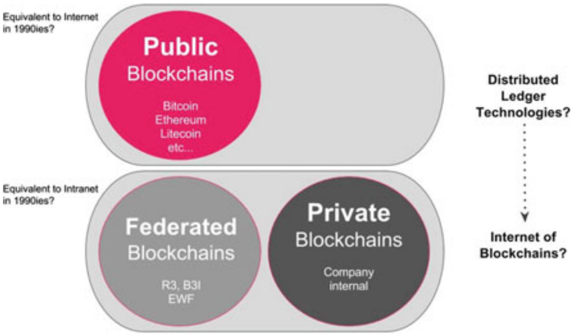
2 Blockchain Technology

Blockchain was introduced in 1991 as a chain of time-stamped blocks by W. Scott Stornetta and his co-author Stuart Haber in his research paper titled "How to time-stamp a digital document" [3]. According to that paper, a document sends by a client to a times tamp server is signed by the server and also linked with the previous document. The term Blockchain became more popular after the invention of Bitcoin. A whitepaper titled "Bitcoin: A Peer to Peer Electronic Cash System" was published by Satoshi Nakamoto in 2008 [1]. This paper was aimed to enable unknown parties to do the transaction with each other directly without the need of a trusted third party like financial institution, means there is no need of relying on a trusted third party.

In 1992, W. Scott Stornetta and Stuart Haber [4] introduced the term "Merkle Trees" into the design of blockchain in his paper titled "Improving the Efficiency and Reliability of Digital Time-Stamping" to allow multiple documents to be collected into a single block. In the Merkle tree, the data structure was patented by Ralph Merkle in 1979 [5], which is used to prove the integrity of the data in the Blockchain transaction. A Blockchain is characterized by decentralization, persistency, anonymity and auditability [6]. The present Blockchain system is classified into three categories: Public Blockchain, Private Blockchain and Consortium/Federated Blockchain [7]. Public Blockchain is permissionless while the Private and Consortium Blockchain are Permissioned [6].

Dubai is planning to be the first Blockchain-powered government in the world by 2020 [8] (Fig. 1).

Fig. 1 Types of blockchain system [38]



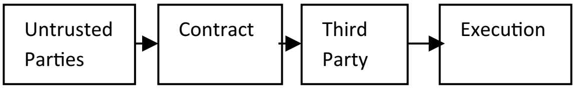


Fig. 2 Traditional contract system

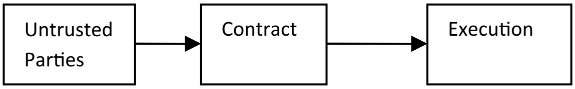


Fig. 3 Smart contract system

3 Smart Contract

The term smart contract was coined by Nick Szabo in 1994 [2]. A smart contract is a piece of computer code used to automatically execute the terms of the contract negotiated in between two untrusted parties after meeting predetermined conditions, in other words, it is an auto-enforceable code. A simple real-life example of a smart contract is the vending machine [9], which is a contract with bearer [a person having coins]. The smart contract technology can change the way business is done in between companies or untrusted parties. Following Figs. 2 and 3 differentiate the tradition contract system with smart contract system.

The traditional contract system is more costly as compared to the smart contract system. Smart contracts are implemented on the blockchain so they also inherit some features of Blockchain-like distributed nature and immutability. A research from Gartner Group which is a leading research and advisory company predicts that by the year 2020, 25% of global organizations all over the world will be using the smart contract [10].

It usually contains four fields that help to construct a smart contract.

i. Address: It is a unique address, by which users or other smart contracts communicate with it.
ii. Balance: It is used to execute a smart contract. After meeting the predefined conditions or terms of contracts, the digital assets are transferred, and the balance is updated.
iii. Code: It is the compiled piece of code which executes the terms of a contract according to the response rules.
iv. Storage: It is an optional field used to store the data.

Figure 4 shows the structure of the smart contract.

4 Smart Contract Development Platforms

Developing a smart contract is not a big deal, but developing a smart contract with no security flaws is a very critical task because a weak smart contract invites an adversary to attack. Parity bug leads to $170 Million of Ether being inaccessible

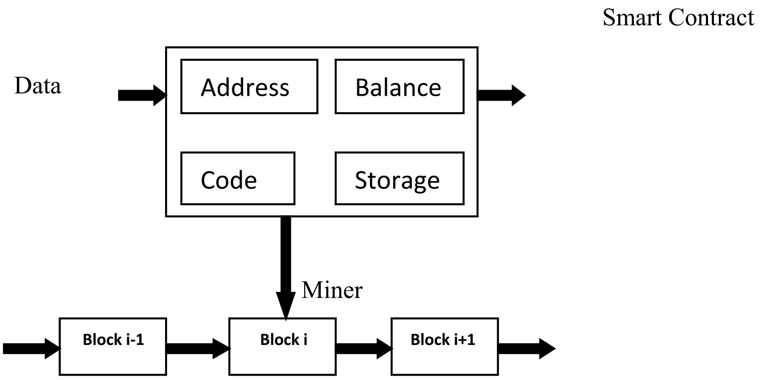


Fig. 4 System of smart contract

[11]. So if the integrity of the smart contract is compromised, then it falls into the area of unsecured immutability. Following are some development platforms which can be used to develop a smart contract.

Etherum: It is a decentralized platform to run and develop a smart contract. It is a distributed public Blockchain network. The solidity programming language, which is very simple to use, is used to create a smart contract on Etherum [12]. The best thing about Etherum is that it is free to use.

Bitcoin: It can be used to create smart contract but the problem with Bitcoin platform is that it cannot be used to create complex smart contract [13].

Hyperledger: It is the Linux foundation platform to develop a smart contract in a high-level language and is completely open-source and free to use. It provides reliable performance with many plug-in components [14]. Go and Java programming is used to code a smart contract.

NXT: It is a public Blockchain platform with templates to facilitate the development of smart contracts [12].

Stellar: If users want high-speed transactions using smart contract, then the Stellar is the best choice because its popularity is rising day by day. The disadvantage of Stellar is that its smart contracts are not turing complete but the advantage is that single transaction cost is approx $0.0000002 [15].

5 Smart Contract Application Areas

Blockchain technology has several application areas, and some of them are as follows:

Financial Services: The use of smart contract reduces the transactional cost because in traditional financial services, a third party is required due to which transactional cost increases [1, 16–19]. State Bank of India announced to use blockchain for a smart contract in November 2017.

Internet of Things (IoT): Smart contracts can be used to automate the functioning of the devices in the IoT, for example, assets tracking in shipping industry [20–22].

Smart Cities: The smart contract can be used in smart city development using sharing service [23].

Smart Transportation Systems: Real-time ridesharing is the excellent application scenario in the social transportation system.

Real Estate: Smart contracts can be efficiently used in renting residential and business buildings [24]. New York's Bapple Realty was amongst the first real estate agencies to use Blockchain smart contracts. (https://dolare.com/blog/post/8-smart-contracts-use-cases).

Digital Identity Management: Digital identity management using smart contract solves the problem of storing personal data because it is stored on the blockchain, so the KYC verification becomes instants [25].

E-commerce: Smart contracts can be used to automate the fulfilment of the orders, and cryptocurrency can be used for payments.

Social Networks: Facebook has collected 300 Petabytes of user data which increase the risk of exposure of data to malware [26]. Blockchain can be used to improve the security of user's data using a personal data management system [27].

6 Research Issues in Smart Contracts

Smart contracts not only have several benefits but also have some major research challenges which should be taken care of. Following are some smart contract issues.

Transaction Ordering Dependency Vulnerability: It is one of the major issues of the smart contract. This occurs when several dependant transactions which are included in one block invoke the same contract [28]. This creates a problem for the miner in which transaction is to be performed first. So if the transactions do not execute in the right order, then an attacker can successfully launch an attack.

Privacy Issue: It is a complicated issue because anybody can access the data on a blockchain. So an adversary can analyse the transactions patterns and may be able to reveal the actual identity [29–32].

Coding of Correct Smart Contract: A smart contract is coded by a coder, who knows the coding language. But if the coder does not able to convert the terms of contract correctly according to the involved parties, then it will create a hazardous condition because the smart contracts have their balances [33, 34].

Time stamp Dependency Vulnerability [28]: Time stamp can be used by smart contracts as a triggered condition to execute transactions. In these typess of scenarios, a miner can do the fraudulent activity by changing its local time.

Immutability of Smart Contract: The smart contracts inherit the property of immutability which makes it unchangeable [35]. But in case of legal contracts, sometimes terms of the contract have to be changed or modify due to the law of the region. So this immutability is a major problem concern to be tackled.

Re-entrancy Vulnerability [28]: This is a condition when an attacker uses a recursive call to conduct multiple withdrawals transactions but his account balance is deduced only once. The distributed autonomous organizations (DAO's) investor had loosed 60 million US dollars due to DAO Bug in June 2016 [36].

Lack of Trustworthy Data Feeds [37]: The smart contracts which are dependent on the external data sources face problems when wrong data is provided by the external sources [39–41].

7 Conclusion

The Blockchain technology-based smart contract work is in a decentralized p2p environment. The smart contract's features are immutability, decentralization, auditability, etc. Make it famous amongst stakeholders all over the world. Smart contract features, development environment, application scenarios and some research challenges have been discussed. Due to the popularity and rapid development in the smart contract, there may be some untouched gaps, which should be identified and addressed in future.

References

1. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system
2. Szabo N (1994) Smart contracts. [online]. Available http://szabo.best.vwh.net/smart.contracts.html
3. Haber S, Stornetta WS (1991) How to time-stamp a digital document. J Crypt 3(2):99–111
4. Bayer D, Haber S, Stornetta WS (1992) Improving the efficiency and reliability of digital time-stamping. In: Proceedings of sequences'91: methods in communication, security, and computer science. Springer, pp 329–334
5. Merkle RC (1979) Secrecy, authentication, and public key systems. Ph.D. dissertation. Stanford University, Stanford, CA, USA. AAI8001972
6. Zheng Z, Xie S, Dai H-N, Chen X, Wang H (2018) Blockchain challenges and opportunities: a survey. Int J Web Grid Serv 14(4):352–375

7. Buterin V (2015) On public and private blockchains. Ethereum blog, crypto renaissance salon, 7th August 2015
8. Lohade N (2017) Dubai aims to be a city built on blockchain. Wall Street J
9. Szabo N (1997) Formalizing and securing relationships on public networks. First Monday 2(9)
10. https://www.gartner.com
11. https://www.theregister.co.uk/2017/11/16/parity_flaw_not_fixed
12. Lewis A A gentle introduction to smart contracts. Available online at https://bitsonblocks.net/2016/02/01/a-gentle-introduction-to-smart-contracts/. Accessed 2/03/2019
13. Wood G (2013) Ethereum: a secure decentralised generalised transaction ledger. Ethereum project yellow paper
14. BigchainDB: the scalable blockchain database powering IPDB, 2017. [Online]. Available https://www.bigchaindb.com/
15. https://hackernoon.com/comparison-of-smart-contract-platforms-2796e34673b7
16. hyperledger (2015) Hyperledger Project. https://www.hyperledger.org/
17. IBM (2016) IBM blockchain. http://www.ibm.com/blockchain/
18. Morini M (2016) From 'Blockchain hype' to a real business case for financial markets. Soc Sci Res Netw
19. Azure (2016) Microsoft Azure: blockchain as a service. https://azure.microsoft.com/enus/solutions/blockchain/
20. Atzori L, Iera A, Morabito G (2010) The Internet of Things: a survey. Comput Netw 54(15):2787–2805
21. Brody P, Pureswaran V (2014) 'Device democracy: saving the future of the Internet of Things. Tech Rep. IBM Institute for Business Value. [Online]. Available http://www935.ibm.com/services/us/gbs/thoughtleadership/internetofthings/
22. Christidis K, Devetsikiotis M (2016) Blockchains and smart contracts for the Internet of Things. IEEE Access 4:2292–2303
23. Sun J, Yan J, Zhang KZK (2016) Financ Innov 2:26. https://doi.org/10.1186/s40854-016-0040-y
24. Karamitsos I, Papadaki M, Al Barghuthi N (2018) Design of the blockchain smart contract: a use case for real estate. J Inform Sec (JIS) 9(3)
25. https://dolare.com/blog/post/8-smart-contracts-use-cases
26. Vagata P, Wilfong K (2014) Scaling the Facebook data warehouse to 300 PB. Technical Report
27. Zyskind G, Nathan O et al (2015) Decentralizing privacy: using blockchain to protect personal data. In: IEEE security and privacy workshops (SPW). IEEE, pp 180–184
28. Luu L, Chu D-H, Olickel H, Saxena P, Hobor A (2016) Making smart contracts smarter. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, CCS '16. ACM, pp 254–269
29. Meiklejohn S et al (2013) A fistful of Bitcoins: characterizing payments among men with no names. In: Proceedings of internet measurement conference, October 2013, pp 127–140. [Online]. Available http://dl.acm.org/citation.cfm?doid=2504730.2504747
30. Ronand D, Shamir A (2013) Quantitative analysis of the full Bitcoin transaction graph. In: Financial cryptography and data security. Lecture notes in computer science. Springer, Berlin, Germany, pp 6–24. [Online]. Available http://link.springer.com/chapter/10.1007/978-3-642 39884-1_2
31. Robinson T (2015) Bitcoin is not anonymous. [Online]. Available http://www.respublica.org.uk/disraeli-room-post/2015/03/24/bitcoin-isnot-anonymous/
32. Coinalytics—blockchain intelligence. Accessed on 15 March 2016. [Online]. Available http://coinalytics.co/
33. Delmolino K, Arnett M, Kosba A, Miller A, Shi E (2016) Step by step towards creating a safe smart contract: lessons and insights from a cryptocurrency lab. In: International conference on financial cryptography and data security. Springer, pp 79–94
34. Bhargavan K, Delignat-Lavaud A, Fournet C, Gollamudi A, Gonthier G, Kobeissi N, Kulatova N, Rastogi A, Sibut-Pinote T, Swamy N et al (2016) Formal verification of smart contracts: short paper. In: Proceedings of the 2016 ACM workshop on programming languages and analysis for security. ACM, pp 91–96

35. Marino B, Juels A (2016) Setting standards for altering and undoing smart contracts. In: International symposium on rules and rule markup languages for the semantic web. Springer, pp 151–166
36. The DAO smart contract. http://etherscan.io/address/0xbb9bc244d798123fde783fcc1c72d3bb8c189413#code
37. Alharby M, Moorsel A (2017) Blockchain-based smartcontracts: a systematic mapping study. arXiv preprint arXiv:1710.06372
38. https://blockchainhub.net
39. Atzei N, Bartoletti M, Cimoli T (2017) A survey of attacks on Ethereum smart contracts (SoK). In: International conference on principles of security and trust. Springer, pp 164–186
40. Christidisand K, Devetsikiotis M (2016) Blockchains and smart contracts for the Internet of Things. IEEE Access 4(2016):2292–2303
41. Luu L, Chu D-H, Olickel H, Saxena P, Hobor A (2016) Making smart contracts smarter. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. ACM, pp 254–269