Journal of Network and Computer Applications 163 (2020) 102635

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Security of Cryptocurrencies in blockchain technology: State-of-art, challenges and future prospects

Arunima Ghosh [a], Shashank Gupta [a], Amit Dua [a], Neeraj Kumar [b,c,*]

[a] Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani, India
[b] Computer Science and Engineering Department, Thapar Institute of Technology and Science, Patiala, India
[c] Department of Computer Science and Information Engineering, Asia University, Taiwan and King Abdul Aziz University, Jeddah, Saudi Arabia

ARTICLE INFO

Keywords:
Blockchain
Distributed ledger
Decentralization
Cryptocurrency
Digital currency
Consensus algorithms
Smart contract
Security.

ABSTRACT

In contemporary era of technologies, blockchain has acquired tremendous attention from various domains. It has wide spectrum of applications ranging from finance to social services and has greatly influenced the emerging business world. Since, blockchain technology is getting embedded in the e-commerce services, the crypto-currencies are gaining huge prevalence. Bitcoin and ethereum are few such crypto currencies, which have utilized decentralized nature of blockchain. Blockchain can be considered as a distributed database system containing immutable ledgers, which are prone to attack by malicious users. Although, from the initial digital currency to the present smart contract, the utilities of blockchain have been harnessed, the innovative technology has to rely on cryptography for its security. There are several reports, which emphases on the vulnerabilities and security of blockchain, however, there is a lack of a comprehensive and methodical survey in both application and technical views. In this survey article, the authors cover various aspects related to blockchain including its taxonomies and the situations in which a particular category of blockchain should be applied. The authors also focusses on the structure of blockchain and the working of the ongoing transactions in the cryptocurrency network. In addition, the authors also specify various categories of consensus protocols, smart contracts, forks, techniques for generating the consensus. A detailed taxonomy of blockchain along with their features and related real-world applications is also discussed. In addition, existing key platforms of blockchain related to the cryptocurrencies, hyperledger and multichain are also discussed. Existing emerging vulnerabilities of blockchain related to the recent attacks on bitcoin and etherum is also presented along with the defensive methodologies and future trends in blockchain.

1. Introduction

1.1. Background

The blockchain technology refers to the immutable public ledgers, which are constructed using decentralized techniques and generally do not contain a trusted authority. This remarkable technique was implemented for enabling the advent of cryptocurrencies in which the exchange of digital assets was take place in decentralized systems. Subsequently, a number of digital currencies has also emerged like Ripple, Bitcoin, Litecoin, Ethereum etc. Blockchain and the cryptocurrencies involved, permitted entities to accomplish economic transactions in the absence of a central authority. It further act as a third party for authentication, while presenting a data storage technique, which is available to all and are legitimate (Nakamoto, 2009). In addition to these features, this exceptional technology hampers any change in the publicized transactions (Rathore et al., 2017, 2018).

In the year 1991, a chain of data, containing digital signature, was utilized as an automated ledger, which signed the documents in a way to assure that any adversary did not tamper the documents in the chain in any way (Narayanan et al., 2016). This was the primary concept for the emergence of blockchain technology. This stupendous technology was first implemented for electronic currency in the year 2008 in a research article which discussed Bitcoin cryptocurrency (Nakamoto, 2008a). The original authors of this technology are still unknown since, the aforementioned paper was pseudonymously published by Satoshi Nakamoto.

* Corresponding author. Department of Computer Science and Information Engineering Asia University, Taiwan.
E-mail addresses: h20180272@pilani.bits-pilani.ac.in (A. Ghosh), shashank.gupta@pilani.bits-pilani.ac.in (S. Gupta), amit.dua@pilani.bits-pilani.ac.in (A. Dua), neeraj.kumar@thapar.edu (N. Kumar).

https://doi.org/10.1016/j.jnca.2020.102635
Received 3 December 2019; Received in revised form 3 March 2020; Accepted 20 March 2020
Available online 20 April 2020
1084-8045/© 2020 Elsevier Ltd. All rights reserved.

This time onwards, blockchain and Bitcoin go hand in hand and blockchain is frequently expected to be utilized for financial transactions.

A number of digital currencies came into existence before Bitcoin, however they could not be operated so extensively. After the blockchain technology was incorporated in bitcoin, the results were splendid as, it attained fascinating features which in turn enhanced its consumption. Bitcoin incorporated with blockchain, was deployed in a distributed environment and hence, single user authority was not provided. Consequently, single point of failure ceased to exist and there was direct transfer of funds among clients in the absence of a third party. In addition to this, it not only permitted fair distribution of funds among the entities (miners), who maintain the blockchain but also reduce the transaction cost in order to utilize system. A self-policing methodology was generated by utilizing a decentralized blockchain technology as well as consensus methodology-based maintenance system, which guaranteed that only legitimate transactions are appended in the blockchain system.

1.2. Motivation

Since, blockchain consists of the above-mentioned features, thus, apart from economic communications there can be several applications of this technology. Some of them include IoT, supply chain management, distributed independent agencies, decentralized cloud storage, healthcare, proprietorship and rights distribution. Recently, the blockchain technology is fascinated by not only the commercial sectors but also gain attention in academia. Some other fields in which this ground-breaking technology is applicable are medical (Ekblaw et al., 2016; Azaria et al., 2016; Yue et al., 2016), finance (Huckle et al., 2016; Bylica et al., 2015; Hurich, 2016; Jindal et al., 2019), IoT (Dorri et al., 2017; Chaudhary et al., 2019; Zhang and Wen, 2016), software engineering (Lee Kuo Chuen, 2015; Buterin, 2014), etc. Fig. 1 focusses on the various domains and the shares of the responder who use blockchain in the corresponding area (financial-institutions). Since, various domains have embraced blockchain technology at a very high rate, various blockchain applications have sprung up and this has led to the transformation of banking and economic services. Fig. 2 discloses the quarterly increase in the number of users who are using the blockchain wallet (wallet-users).

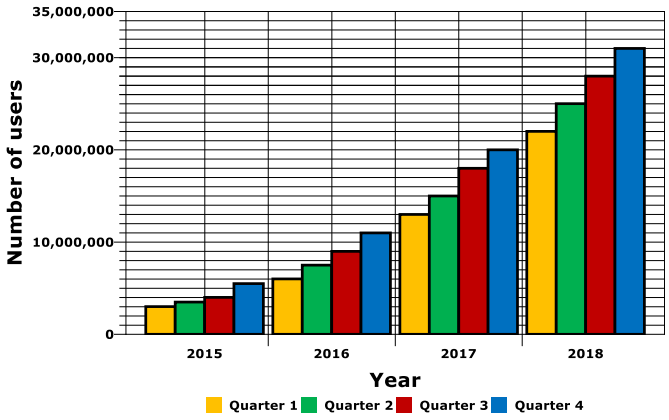

Fig. 2. Statistics in terms of increase in number of users of blockchain wallet.

1.3. Blockchain in bitcoin

There were many cryptocurrencies launched and among them Bitcoin was the most publicized and successful. It has a special kind of data structure used for storage and transactions in its network can occur without involving a third party. The primary technique used in the construction of Bitcoin is the blockchain technology, which came into existence in the year 2008 and its implementation was performed in the year 2009 (Nakamoto, 2008b). Bitcoin was surveyed as the highest operating currency in the year 2015 (Desjardins, 2016) and the greatest operating product in the year 2016 (Adinolfi, 2016). In the same year, (i.e., in 2016) blockchain is accounted to have reached 10 billion dollars in its capital market. In the year May 2017, it was reported that bitcoin has transactions greater than 300 K (blockchain.info, 2017) on daily basis.

1.4. Blockchain in ethereum

With the inception of programming languages which are turing-complete, few languages like solidity and serpent came into existence which enabled the users to design smart contracts which will execute on the blockchain and thus, the era of blockchain 2.0 began. With the advent

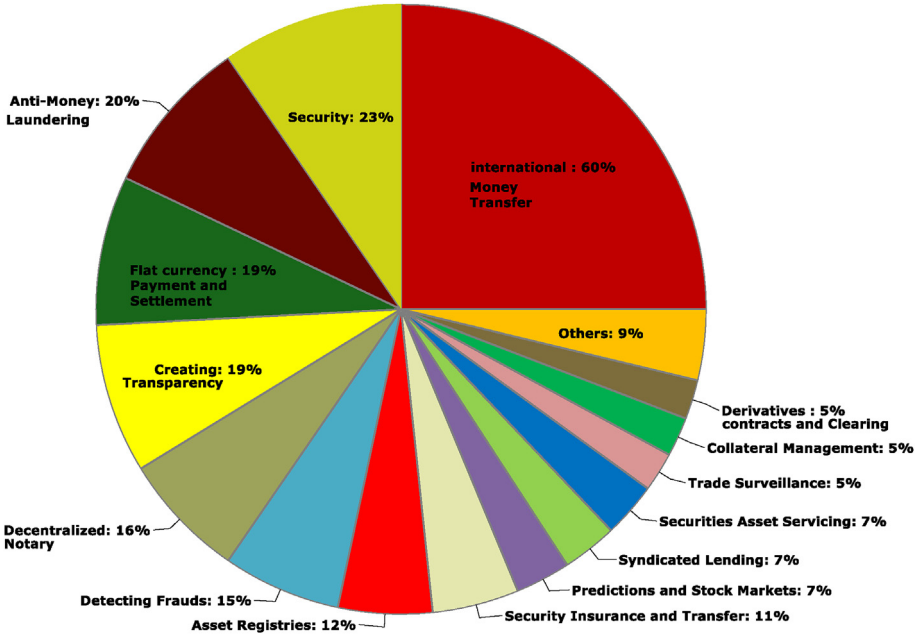

Fig. 1. Various financial applications of blockchain across world in 2016 (financial-institutions).

of blockchain 2.0, a number of new cryptocurrencies came into existence, which had smart contracts implemented in them. Some of them are Ethereum, Ethereum classic, Hyperledger Burrow, etc. Currently, Ethereum is regarded as the most extensively used blockchain which supports smart contracts. Till now, we already have 317,506 as smart contracts number and transactions greater than 75,000 happened on daily basis (Lee Kuo Chuen, 2015). The core technology used to develop various cryptocurrencies is Blockchain because they make use of its decentralized nature. As already mentioned, blockchain has distributed consensus mechanism, thus there is no need for a trusted third party to exchange information or perform transactions. Thus, the distrusted users involved can accomplish their task without any central authority.

1.5. Our contribution

From the above explanation, the authors realized that in order to dig deep into the cryptocurrencies and comprehend their operations and vulnerabilities, they have to focus on their foundation, i.e., blockchain. Hence, this article not only discusses the architecture and mechanisms involved in blockchain, however, also focusses on the cryptocurrencies, their vulnerabilities and exploitations of those vulnerabilities. Further, it elaborates on the enhancements made in the field of blockchain and regions in which improvements can be made.

1.6. Outline

The remaining paper is structured in the following format. In section 2, the authors are focussing on brief overview of the blockchain technology. Section 3 focusses on the emerging blockchain technologies like consensus methodologies, smart contract, forks, etc. Section 4 focuses on expansions of blockchain, categories and applications of blockchain. Section 5 emphasises on the platforms in blockchain like

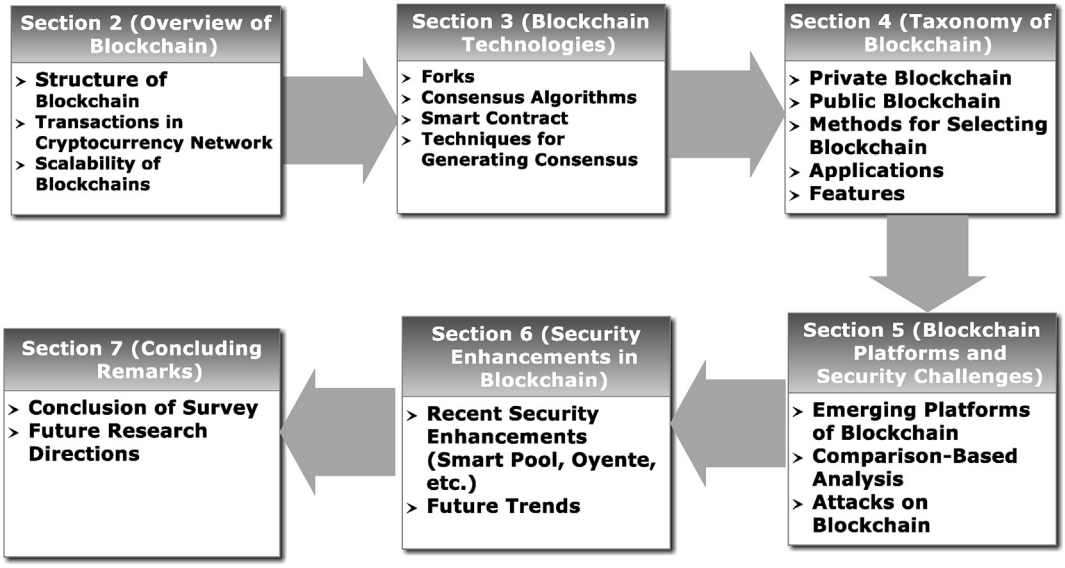

Fig. 3. Outline of the paper.

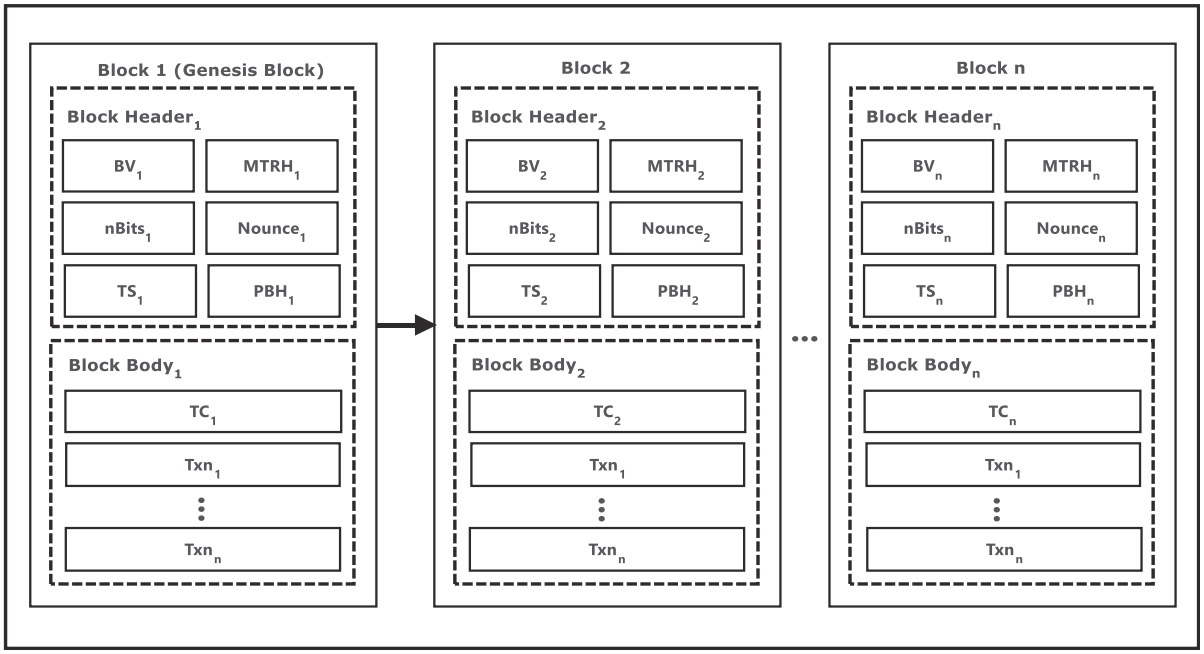

Fig. 4. Structure of Blockchain and its constituents.

cryptocurrencies, Hyperledger, multichain etc. This section also highlights the challenges and vulnerabilities of blockchain and also includes the attacks on the two major cryptocurrencies - Bitcoin and Ethereum. Existing security enhancements in blockchain is discussed in section 6. Finally, in section 7, the authors conclude their survey by reviewing the enhancements and future trends in the field of blockchain. Fig. 3 displays schematic representation of the organisation of this survey article.

2. Overview of blockchain

Since, blockchain works in the absence of a central authority (i.e., in a distributed environment) and they consist of public digital ledgers, which are immutable in nature. If any user in blockchain network wants to perform a transaction, his request is stored in a ledger in a node whose copy is available to the all the users in the network. These users perform verification of the transaction in the node and if the users reach a consensus, the node is found to be authentic and only in that case it is added to the blockchain as new block. After this, the transaction cannot be altered. Now, if a user wants to perform any malicious activity, then he will have to take control over the entire network of blockchain because the copy of transaction is available to everyone. Practically, modification of the transaction in a malicious way is an almost impossible process however, in theory it is possible to corrupt it.

Blockchain technology may seems to be a very simple process however, there are many complexities involved in this technology. Several mechanisms which are present in computer science community like distributed network, cryptography, data structures are involved in blockchain. These are amalgamated with some of the concepts of finance like ledgers. The next subsection enlightens the readers in briefly understanding the architecture and mechanism of blockchain.

2.1. Structure of blockchain

Blockchain name clearly signifies that it is a series of connected blocks. These connections are possible since, each block has a *parent block* (previous block), whose hash is recorded in the related header of the block. In case of Ethereum blockchain, the hashes of block's ancestors (*uncle blocks*) are also stored (Buterin, 2014). The *genesis* block (first block) does not have any parent block. Each block comprises of two parts:

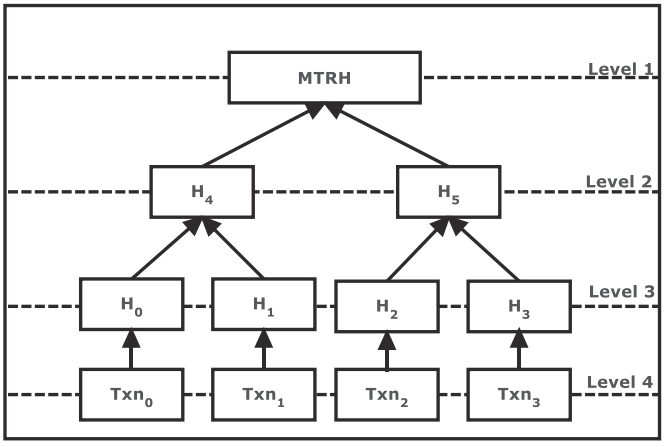

Fig. 5. Structure of merkle tree root hash.

- ***Timestamp:*** It represents the current time (in seconds) since 1st January 1970.
- ***nBits:*** It is aimed threshold of the hash value of an authentic block.
- ***Nonce:*** Nonce usually begins with a 0 and is incremented for each hash value computation. Its size is 4-byte. This is further explained while explaining PoW mechanism.
- ***Parent block hash:*** This is a hash value of size 256-bit, which indicates at the preceding block.

(i) *The Block header* (ii) *The Block body*.

Fig. 4 clearly illustrates the block header, which consists of Block Version (BV), Merkle Tree Root Hash (MTRH), nBits, Nounce, Time Stamp(TS), Parent Block Hash(PBH) and the Block Body, which holds Transaction Counter(TC) and entire records of transaction such as conventional public ledger (Lee Kuo Chuen, 2015).

2.1.1. Block header

The header of a block in the blockchain comprises of six attributes. All of them are explained as under.

- ***Block version:*** A blockchain network consists of few authentication rules that needs to be followed, therefore, block version denotes the set of protocols to obey.
- ***Merkle tree root hash:***It is defined as the hash value for the entire block. Instead of saving the hash value of all the transaction, a single hash value is created using the Merkle tree. This tree merges hash values of all the transaction together (taking two at a time) till one hash value is achieved. This is called a Merkle tree root hash. This is an effective method to encapsulate and authenticate all the transactions in a block. It supports in delivering immutability since, block hash value is stored in the child block header also, and any alteration to transaction will result in mismatch of Merkle root hash. Fig. 5 displays the working mechanism of a Merkle tree:

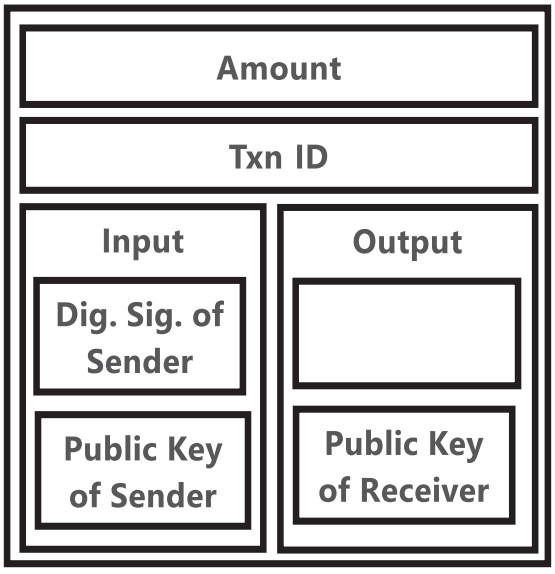

Fig. 6. Structure of a transaction that a block contains.

- ***Amount*** – The sum of all the digital values that needs to be transferred.
- ***Inputs***– The input includes a log of the values of the digital asset that needs to be transmitted (the entire value must equal to the amount). Here, all the digital assets must be exclusively recognised and could include values that are distinct to other assets. Though, credentials could not be incorporated or eliminated from the recent digital credentials. A replacement for this is that the electronic possessions may be divided into numerous new digital possessions (having less values) or merged for creating some new digital asset (with higher value).
- ***Outputs***– This stores the details of the accounts that act as recipients of the value. It consists of the digital asset that will be transmitted to the account of the recipient, the unique identity related to the recipient, and certain rules the recipient should not violate for receiving the related value. In case digital value offered extra assets, it is refunded. ("*make change mechanism*").
- ***Transaction ID or Hash***– Every transaction has an exclusive value for its identification. It can either be a transaction ID or transaction hash value. It is essential to authenticate a transaction for the digital signature which is based on public key cryptography (NRI, 2015).

➢ The leaves of the tree (level 4) represent the transactions records from Txn_o toTxn_3 be encapsulated.
➢ Level 3 displays the hashed value of the transaction records.
➢ The hashed value in level 3 is then merged and we get new hashed value in Level 2
➢ Lastly, level 1 displays the Merkle Tree Root Hash, which merged hashes H_4 and H_5.

2.1.2. Block Body

The body comprises of transaction counter as well as transactions. The size of transaction and block determines the largest number of transactions which can be present in a block.

- ***Transaction Counter***:It stores the number, if transactions are in the block.
- ***Transaction***: It refers to a log of transmission of assets between two entities. In blockchain, several transactions are present in one block. A typical transaction is displayed in Fig. 6 and usually involves the following attributes:

2.2. Lifecycle of a transaction in a cryptocurrency network

This section illustrates the transaction steps involved in bitcoin network between several users. Fig. 7 shows the processing of transaction in a cryptocurrency.

If sender *A* wish to send few bitcoins to receiver *B*, he/she must have a Bitcoin full installed in his/her device. An alternate option to Bitcoin full is lightweight client-side software. Along with this, he/she requires sender's private key information and receiver's Bitcoin address. All the entities in the blockchain network have the transfer digital asset to the sender's Bitcoin address. Nevertheless, only an exclusive signature which is created with the help of private key has the ability to allow the transaction of bitcoins from account. In order to prove that the amount being sent belongs to the sender, he/she utilizes a cryptographic key for implementing digital signature on transaction. As soon as, the sender publicized his transaction in the bitcoin network, a signal is broadcasted to every miner present in the network. This is done in order to inform the miners about the arrival of the new transaction. Subsequently, the miners authenticate digital signatures, and also verifies if sender is transferring amount within its specified limits.

Furthermore, miners compete with one another to gather all awaiting transactions in the network (including the senders) and mine the block (by fluctuating nonce). An elaboration to this is that, miners generate hash of the entire block, subsequently the hash value should commence with a definite number of zeros, if it does not, a new nonce is selected and function to generate hash value is run again. Initially, miners arbitrarily opt for any value of nonce and if the hash function is run again, nonce is incremented and the new value is selected. This process take place until the miner or another miner solves the problem. Once, the aimed value found, the message is broadcasted in the network. The sender and receiver also receive an acknowledgment stating the success of transaction. Other miners in the network accept the new block, and then commence to determine the succeeding block in the cryptographic network. Although, a transaction which is successful, can be rejected later in case it is incapable of staying in the blockchain network if there exists numerous forks or many of the miners do not approve to accept the block which contains this transaction, detection of double spending attack, etc. According to the rules of the Bitcoin, the miner who mines the block gets some bitcoins as reward, moreover the block is incorporated in the public ledger. When the sender's transaction is incorporated in the blockchain, the sender and the receiver receive the acknowledgment that bitcoins are transferred to the receiver. The time taken by one transaction is dependent not only on the load in the network but also on the transaction payment incorporated by the sender. Minimum time required is

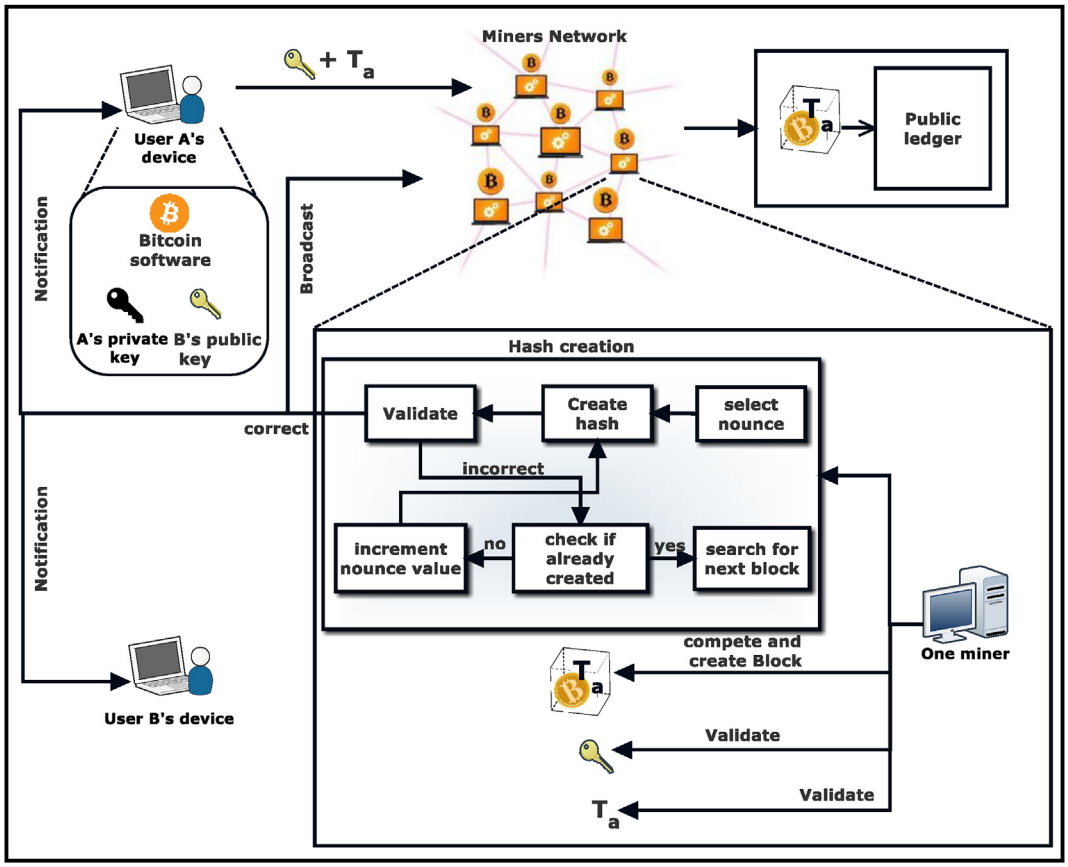

Fig. 7. Lifecycle of transaction in bitcoin network.

approximately 10 min. Nevertheless, if first acknowledgment is received, this does not signify successful processing of transaction. The transaction can be considered as illegitimate any time. For considering a transaction as legitimate, the Bitcoin society suggested that after mining the block, it must get successive acknowledgments for block (presently six).

2.3. *Expansion of blockchain through addition of new blocks*

A new block mainly indicates a list of transactions. Here, we will assume a permissionless (or public) blockchain that uses the capabilities of the Proof of Work (PoW) consensus mechanism (Miglani et al., 2020; Karl, 2016a, 2016b). It is a well-known methodology on which bitcoin is operating. The blockchain network consists of miners who have a blockchain software installed in their devices. The consensus of these miners is required to maintain the blockchain in the network. Since, the system is decentralized, hence, central control has not the authority to control who decides which entity should publish the succeeding block in the blockchain. Every entity should keep record of blockchain and might recommend some new block to other miners. It is computationally feasible to authenticate a block as compared to computing a block, therefore illegitimate blocks are easily sensed and discarded. According to the application of blockchain, the process of mining in blockchain requires either memory or processing power or both. The consensus mechanism takes the decision of the new block that will be incorporated in the blockchain. Details of the mechanism are mentioned in the later sections. Any device running the software of the blockchain is regarded as a *node*. There exist two categories of nodes: (i) full nodes (ii) light-weight nodes.

2.3.1. *Full node*

A *full node* records the blockchain information, forward the data to the rest of the nodes, and guarantee that new blocks are legitimate. Authentication certifies that the block format is valid, hashes present in the recent block are accurate, preceding block hash is present in the recent block, and every transaction that a block contains is authentic and digitally signed by entities involved. A full node could also behave as miners.

2.3.2. *Lightweight nodes*

These do not record entire duplicates of blockchain, instead they may forward their information to the full nodes for processing. These are generally those devices which have less computational power or memory, e.g. smartphones 588, IoT devices, etc. Any of the entity or node in the network can recommend some recent transactions. These new transactions are broadcasted to nodes till, they are incorporated in a block.

Projected transactions in blockchain are recorded in the miners in unspent transaction pool, in anticipation of getting incorporated in a block. When a new block in created, the miners incorporate a group of unspent transactions in it. This group may consist of an amalgamation of some delayed transactions and some recent transactions that present a greater payment (transaction fee). If invalid transactions are present, the miners discard the entire block. In order to avoid this situation. the miner itself checks the validity of every transaction. Now, the miner will fill all the data, except nonce, which is essential for the block structure.

Few of the blockchain systems may necessitate a type of sacrifice for generating the following block. This can be spending time, energy, risking for the benefit, etc. If the endeavour and time requirement of the

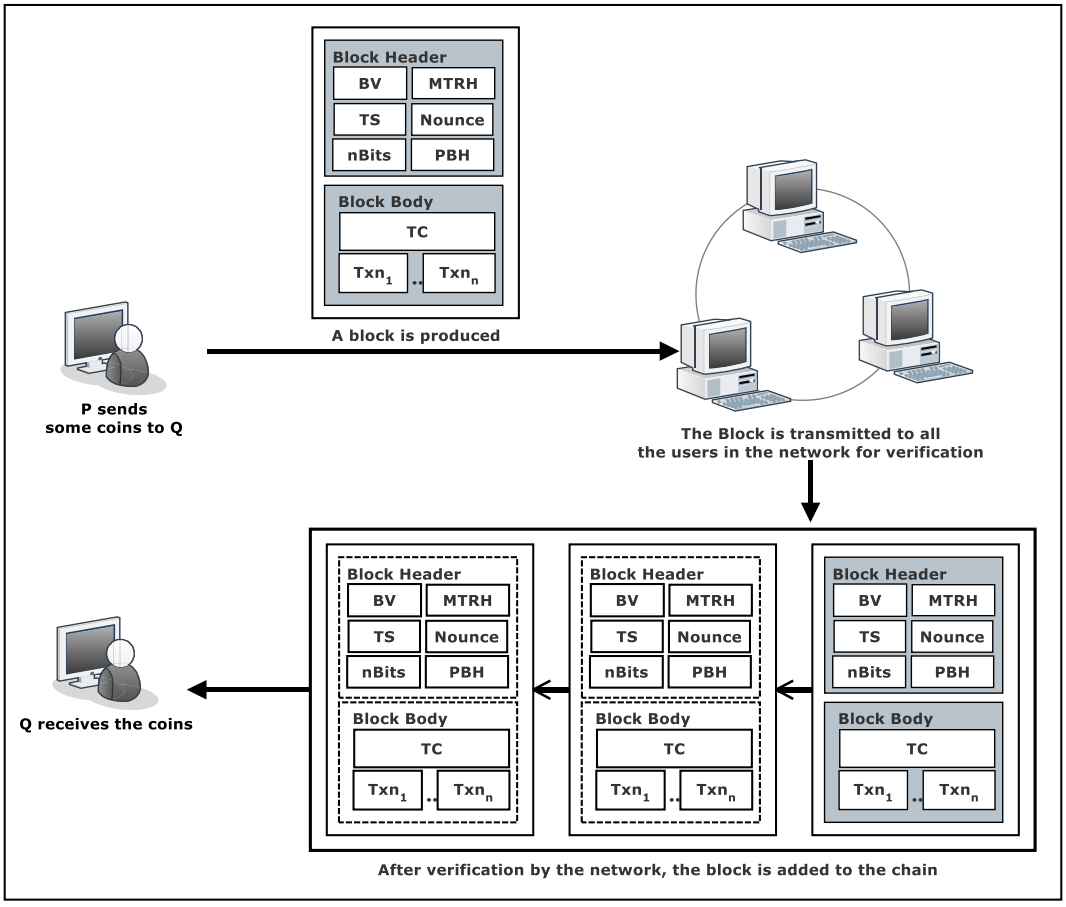

Fig. 8. The process for expansion of blockchain by addition of a new block. BV_n - Block Version of nth block, $\mathbf{BV_n}$ - Block Version of nth block, $\mathbf{MTRH_n}$ - Merkle Tree Root Hash of nth Block, $\mathbf{TS_n}$ - Time Stamp of nth Block, $\mathbf{PBH_n}$- Parent Block Header of nth Block, $\mathbf{TC_n}$- Transaction Counter of nth Block, $\mathbf{Txn_n}$ - nth Transaction.

system is high, the miner will have to calculate numerous arbitrary nonce values for trying to resolve a computationally hard problem. The winner entity secures the privilege to issue the succeeding block. Generally, miners test several nonce values before resolving the puzzle. After the puzzle is resolved for some nonce value, the entity will create hash of the data of the block and record it in the block. Fig. 8 displays the architecture of the created block. This block is broadcasted in the blockchain network for authentication. After the block being authenticated, the nodes admit it as new block and forwards it.

3. Blockchain technologies: forks and consensus methodologies

The above section clearly describes the structure of blockchain in detail. In addition to this, it also enlightens the procedure which is used to append new blocks to the existing blockchain. In this section, the authors illustrates the various mechanisms involved in the blockchain (forks and consensus methodologies), which will further help the readers to gain a better understanding of the previously discussed concepts.

3.1. Forks

Since, the blockchain technology is distributed in nature and the consent of entities are required, updating rules in the network may become almost impossible. Variations in software of blockchain as well as implementation is known as fork. Fig. 9 illustrates the overview of soft and hard fork.

3.1.1. Soft forks

This occurs when device of the entity encounters with new rules and is incompatible with the preceding version, the new entities in the network will not acknowledge mining of early entities. Since, computation power of recent entities are more robust than early entities, block mined by early entities will not be sanctioned by recent entities. Nonetheless, the recent entities and early entities will mine on the common chain. Once soft fork occurs, upgradation to new rules may not happen simultaneously, it permits gradual upgradation. Soft fork has one chain, moreover, after upgradation, it does not affect constancy and efficiency of system. Though, in soft fork, the early nodes are insensitive to the fact that the agreement rules have been changed, which is contrary to a protocol, which states that each entity can authenticate appropriately to some range.

A soft fork is a modification in the protocols of the blockchain technology that would not entirely prohibit the entities, who refuse to accept the alteration from operating on the altered system. An example of alteration can be upgrade to contemporary version. As, obsolete entities would identify recent blocks as authentic, a soft fork is capable of backward compatibility, which necessitate only majority of entity upgrade in order to implement new rules of soft fork.

When a latest consensus protocol was incorporated to strengthen "*escrow*"as well as time-locked repayments, soft fork happened on Bitcoin network. In the year 2014, a proposition repurposing an operation code was made which implemented OP_NOP2 (no operation) to CHECK_LOCK_TIME_VERIFY, that permitted yield of a transaction for being not spendable in the near future (blockchain.info, 2017; Todd, 2014). Hence, for the users who tend to deploy this modification, the interpreter of blockchain would implement this latest operation, nonetheless for clients who do not have provision for the modification. The script remains authentic, moreover execution would pursue as if *"**NOP**"*is executed.

3.1.2. Hard forks

It is a modification in the protocols of the blockchain technology that would entirely prohibit the entities who refuse to accept the alteration from operating on the altered system. In hard fork, protocols would be modified in such a way that necessitates entities to update to remain with *"main fork"* or continue the primary chain. Entities present on distinct hard forks can never communicate. If there is modification in the structure of the block, e.g. selecting hashing algorithm, it would need hard fork.

In the year 2016, DAO (Decentralized Autonomous Organization), a smart contract was implemented in Ethereum. There were few faults in the construction of the smart contract because of which a malicious user pulled out Ether, which subsequently led to burglary of $50 million (Wong and Kar, 2016a). All the ether holders voted for a hard fork proposition, which was approved by 89 percent and thus, created a new variety of blockchain, returning of the robbed assets. The old chain was renamed as Ethereum 842 Classic, which was supported by few original users. In cryptocurrencies, in case, a hard fork is existing and blockchain is divided, coins that an entity possess at that time would be copied to each fork. In case majority of the activities transfers to recent blockchain, the old chain would not be used. Table 1 illustrates about few of the differences that are prevalent in hard and soft fork in the blockchain technology.

3.2. Consensus methodologies

Since, the blockchain systems are decentralized in nature, they do not require a trusted centralized authority. Decentralized consensus methodologies are implemented by blockchain in order to provide dependability and uniformity of data as well as secure transactions. Currently four major consensus mechanisms are used in blockchain technology: Proof of Work (PoW), Practical Byzantine Fault Tolerance (PBFT), Proof of Stake(PoS) and Delegated Proof of Stake(DPoS) (Zheng et al., 2016). Some other consensus methodologies which have been implemented in few of the blockchain technologies are Proof of Bandwidth (PoB) (Ghosh et al., 2014), Proof of Authority(PoA) (P. technologies, 2017a), Proof of Elapsed Time (PoET) (Intel, 2017), etc. Among all these, PoW is deployed

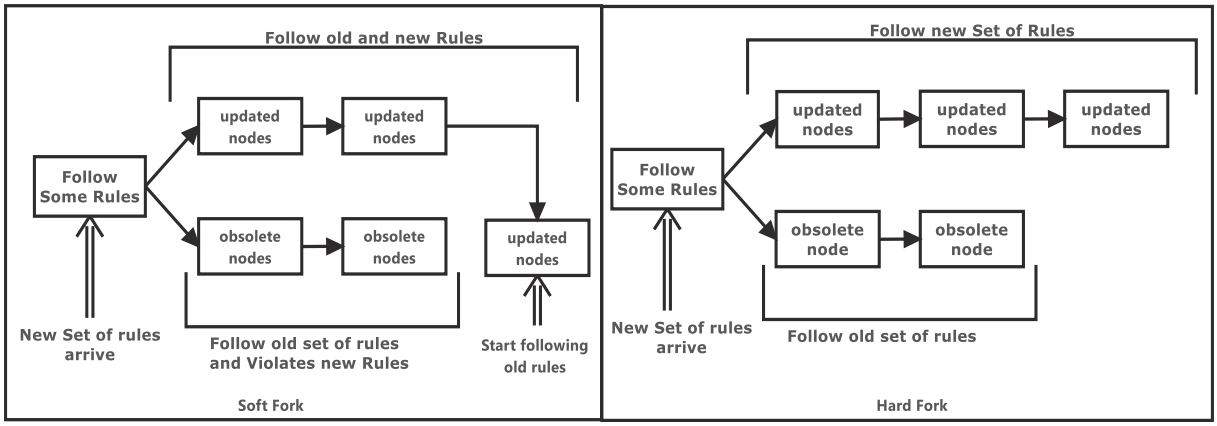

Fig. 9. Overview of soft fork and hard fork.

Table 1
Comparisons–based analysis of variations of forks.

Categories	Hard Fork	Soft Fork
Divergence type	Permanent divergence in the block chain	Temporary deviation in blockchain
Cause	The entities which are not upgraded are unable to validate the blocks constructed by the upgraded entities (obeying newer consensus protocols).	When non-upgraded nodes not following new consensus rules
Backward compatibility	Backward compatible.	Not backward compatible.
Parallelism of chains	The new as well as old blockchain execute parallelly, however both follow distinct set of protocols.	There are no parallel chains.
Funds	Brings up the issue of duel funds	No concept of duel funds
Implementation type	The new protocols which give rise to compatibility should be deployed in hard fork.	Most of the new features like check sequence verify or CSV or segregated witness are deployed by a soft fork because it is secure and more trivial.

in Bitcoin and Ethereum, which are the two prevalent blockchain systems (cryptocurrency). In addition to PoW, Ethereum also integrates PoA methodology (that is, Kovan public test chain (E. community, 2017a)), and few cryptocurrencies, like PeerCoin, Shadow Cash, etc, deploy PoS methodology. Table 2 shows the categories of consensus algorithms. Recently, IoT has acquired tremendous attention from various domains. It has wide spectrum of applications ranging from finance to social services and has greatly influenced the emerging business world. Since, IoT technology is getting embedded in the e-commerce services, the consensus algorithms are gaining huge prevalence. PoW and PoS are few such consensus algorithms, which have utilized the platforms of IoT. IoT can be easily integrated in the distributed database system containing immutable ledgers using several consensus algorithms, which are prone to attack by malicious users. Although, from the initial digital currency to the present smart contract, the utilities of consensus algorithms have been harnessed, the innovative technology has to rely on cryptography for its security.

3.2.1. *Proof of work (PoW) algorithm*

Bitcoin cryptocurrency implements the PoW consensus methodology (State of blockchain q1, 2016). In a distributed system, an entity is elected for storing the transactions. The simplest method is to select randomly. Nevertheless, this type of selection has high risk of attacks. Therefore, for publishing a block, an entity has to perform many operations for proving that the entity involved is not malicious. Fig. 10 shows the working of the proof-of-work mechanism.

This technique utilizes the answer of problems for validating the authenticity of data. The problem is generally computationally hard but

Table 2
Types of consensus algorithm and their comparisons.

Consensus Algorithms Parameters	Tendermint	Delagated Proof of Stake	Ripple	Proof of Stake	Proof of Work	Practical Byzantine Fault Tolerance	Proof of Burn	Proof of Capacity	Proof of Elapsed Time
Example	Tendermint	Bitshares	Ripple	Peercoin	Bitcoin	Hyperledger Fabric	Slimcoin	Burst Coin	Sawtooth
Threshold for attack	33.33% malicious Nodes	33.33% Malicious Nodes	20% Malicious Nodes	51% Hash power	25% hash power	33.33% Malicious nodes	23% Hash Power	27% Malicious Nodes	25% hash power
Knowledge of Node Identity	Validators	None	None	None	None	Miners	Miners	None	Validators
Energy Consumption	Low	Moderate	Low	Moderate	High	Low	Moderate	High	High

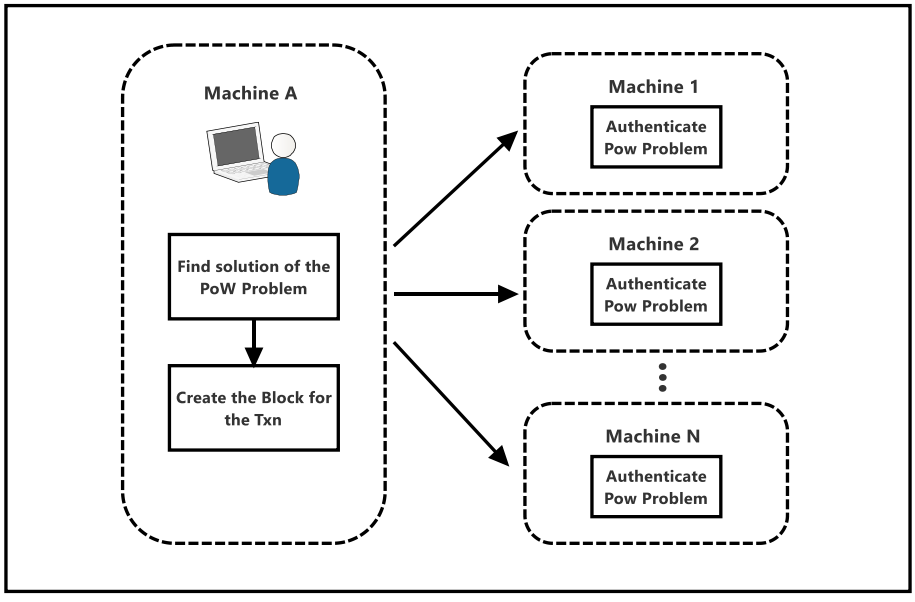

Fig. 10. The overview of proof-of-work consensus methodology.

verifiable. Subsequently, it would broadcast the block to other entities in the network to attain consensus, as displayed in Fig. 10. The structure of a block in a blockchain varies from system to system. In a cryptocurrency such as Bitcoin, a block is generally composed of PBH, nonce, and Txn (Luu et al., 2017; Buterin, 2013a; Aggarwal et al., 2019). Here, nonce value is derived by resolving the PoW problem. A valid nonce must assure that hash displayed in the Equation-I, is smaller than the aimed value, that can be altered for changing the complexity of Proof of Work problem.

$$SHA_{256}(PBH \parallel Txn1 \parallel Txn2 \parallel \ldots \parallel nonce) < \text{Aimed value} \quad (I)$$

When block is authenticated, rest of the miners will join this block to their blockchain. Entities which compute hash are known as miners. In Bitcoin, the PoW methodology is known as mining. In distributed network, legitimate blocks may be produced as soon as two or more entities find appropriate nonce. If nonce is found by these entities simultaneously then branches might be produced as displayed in Fig. 11. Nevertheless, it is doubtful that the two rival forks will produce succeeding block also concurrently. In PoW methodology, a branch that turns out to be longer later is estimated to be valid. Let us assume two forks generated by simultaneously authenticated the blocks. The miners will continue to mine the blocks till a lengthy chain is established and later, the miners may shift to the lengthy chain.

In PoW methodology, miners perform numerous computations, therefore these operations waste the available resources. To avoid this, few PoW methodologies, where works can have some supplementary-applications are constructed. An example of this is Primecoin (Luu et al., 2017) which explores to find some special chains of prime numbers which might be productive in mathematical research.

3.2.2. Proof of stake (PoS)

The PoS methodology utilizes proof of proprietorship of the corresponding cryptocurrency for proving the authenticity of data. The blockchain system in which the PoS is implemented, while designing either a block or a transaction, the entities involved are needed to give some assets. When the designed block or the designed transaction are authenticated, the asset (that was paid) would be refunded to the initial entity as bonus. If this is not the case then, it will be penalized. In PoW methodology, plenty of computations are required, which results in wastage of computing power. However, in PoS methodology the computation can be decreased to a large extent, thus the efficiency of the blockchain system is increased.

3.2.3. Practical Byzantine Fault Tolerance (PBFT)

The PBFT consensus methodology is a duplication algorithm to endure byzantine faults (Buterin, 2013a; Aggarwal et al., 2019). PBFT can process approximately one-third of the illegitimate byzantine duplicates, thus, it is used as consensus methodology in Hyperledger Fabric (NRI, 2015). In one round, only one block is determined. In one cycle, a key entity is elected concurring with few protocols who orders transaction. The entire process can be partitioned in three stages: (i) *pre--repaired*, (ii) *prepared* (iii) *commit*. At every stage, an entity will progress to the succeeding stage only if it is elected by more than two-third of all entities. Therefore, PBFT has a prerequisite that each node should be recognised to the entire blockchain network. SCP (Stellar Consensus Protocol) (Aggarwal et al., 2019; Eyal and Sirer, 2014a) is a Byzantine agreement protocol and has similarity with PBFT. Difference between PBFT and SCP is that in case of PBFT, every entity is required to interrogate other entities whereas SCP grants entities the authority to select set of the entities who are supposed to be trusted. A modified version of PBFT called Delegated Byzantine Fault Tolerance (DBFT) is implemented by Antshares (Luu et al., 2015a). In DBFT, few specialized entities are elected for storing the transactions.

3.2.4. DPoS (Delegated Proof of Stake)

In case of PoS, it behaves as direct democratic whereas DPOS behaves as a representative democratic. Participants select the representatives for generating and authenticating the blocks. If there are lesser entities to authenticate the block, less time will be taken for validation. This will lead to fast approval of the transactions. In the meantime, the specifications of network like *block size* as well as *block intervals* can be changed by the representatives. Moreover, clients do not have to be concerned about the malicious representatives since, they will be recognised easily and voted out. This consensus mechanism is used in Bitshares (P. technologies, 2017a).

3.2.5. Ripple

Another consensus methodology, Ripple (Intel, 2017), makes use of collaborative subnetworks (which are fully trusted) inside a bigger network. In such type of blockchain networks, the entities are classified into two categories: (i) *Server (ii) Client*. The server takes part in the consensus procedure while, the clients transfer assets. A server contains a Unique Node List (UNL), which is essential for server. To decide if a transaction has to be incorporated in a ledger, server interrogates the entities in UNL. In case, more than 80% consensus are received, transaction will be added in the ledger. An entity considers a ledger to be

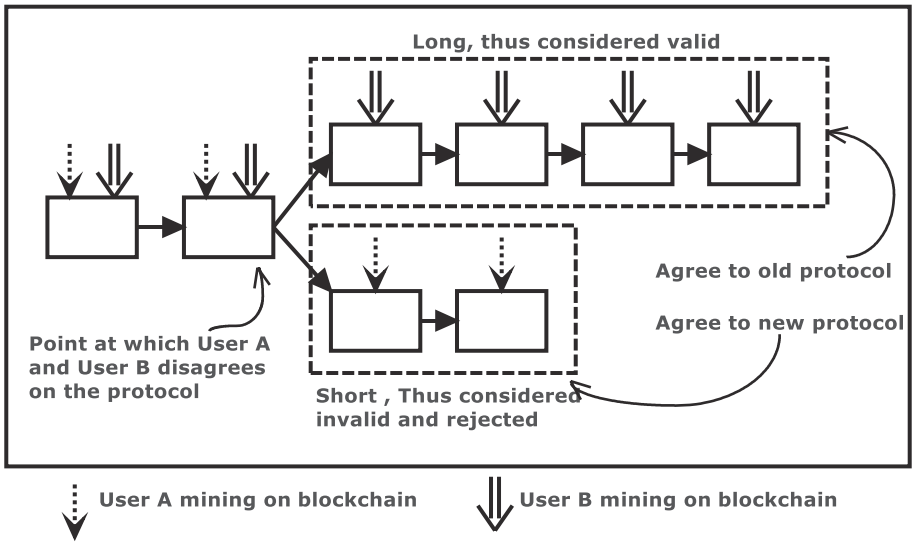

Fig. 11. Process of forking in PoW consensus methodology.

legitimate if malicious entity percentage in UNL is lesser than 20%.

3.2.6. Tendermint

A byzantine consensus methodology, in which one new block is found in one cycle, is used by tendermint (E. community, 2017a). In a round, an entity, called *proposer,* will be chosen for broadcasting an unauthenticated block. This procedure is classified in three stages: (i) *Prevote stage*: In this step the authenticators determine if they should transmit prevote intended for the recommended block. (ii) *Precommit stage*: In this, if the entity collects prevotes greater than two-third for recommended block, it will broadcast precommit for the recommended block. In case, the entity collected precommits greater than two-third, it will move to the commit stage. (iii) *Commit stage*: In this stage, the entity will authenticate the recommended block and will broadcast commit in the network. In case, the entity collected two-third of commit messages, it will admit the block. However, in PBFT, entities must lock their assets for becoming an authenticator. If an authenticator is discovered to be malicious, it will be penalized.

3.3. Qualities of a good consensus algorithm

The attributes of an upright consensus methodology are effectiveness, security and ease of use. Lately, much work is done for the advancement of the consensus mechanisms. Few latest consensus methodologies have come forward which targets to resolve few issues in blockchain. One of them is PeerCensus (Decker et al., 2016), which aims to separate block generation from transaction authorization for increasing the speed of the consensus process. Another consensus mechanism is Kraft (2016), which proposes a consensus algorithm for ensuring that block generation speed is stable. Since, if block generation speed is high, the security of Bitcoin is negatively affected, and for solving this issue, GHOST (Greedy Heaviest Observed Sub Tree) chain selection rule (Sompolinsky and Zohar, 2013) is projected.

In this consensus methodology, the lengthiest chain of blocks in not selected. GHOST provides weights to the chains and the miners can select the one, which they find better. Chepurnoy et al. (2016) has projected another type of consensus methodology, in which any entity who is providing non-interactive evidence of retrievability for the past state snapshots is allowed to create block. In this type of methodology, miners will be required to save the past block headers, rather than storing entire blocks.

3.4. Smart contract

A smart contract is a treaty among disbelieving members, which is implemented by the blockchain's consensus methodologies. The computer code and data in smart contracts are often called as methods and states. The expected transactions received by the blockchain may call the contract's public methods using its data for performing a service. Since, the code is on the blockchain, it is immutable and may be treated as third party to perform complex financial transactions. Smart contracts may carry out calculations and record data for financial transmissions.

While mining blocks, the miners also execute smart contract programs. Therefore, execution of the smart contracts has higher cost as compared to transfer of assets in other blockchain-based cryptocurrencies. Apart from paying for normal transaction fees, the client, who is requesting for a transaction to a smart contract, must also pay the charges for program execution. Limited execution time is allotted for a smart contract call. In case, it is exceeded, the program execution will be terminated, and transaction is rejected. For the execution of this code, the miners are rewarded and an adversary is prohibited from deploying and subsequently fetching the smart contracts that consume all resources and execute DoS attack (denial of service) on miners.

Fig. 12 displays the procedure related to development phase, deployment phase and interaction phase of smart contract. Every installed smart contract refers to an IPaddress, using which the consumers may communicate with the smart contract using transactions via various clients (for instance, Parity, Geth, etc.). A smart contract is capable of calling other smart contracts via messages and hence, programmers may develop more dynamic dAPPs.

3.4.14.1. Smart contract in ethereum

Ethereum (Buterin, 2013b) is the most famous framework in which smart contract is deployed. It contains smart contracts as computer codes, which is implemented in EVM bytecode (Buterin, 2013b) which is Turing-complete. Smart contracts in Ethereum can also be used for transferring ether, a digital currency, to or from different consumers and to some other contracts.

The target of Ethereum's consensus methodology is to guarantee accurate execution of contract. For appending a block to blockchain, the entity should take part in a lottery, in which the winning probability increases with the increase in computing power of an entity. A reward methodology makes sure that, in case an adversary (after winning lottery) attempts to append a block with illegitimate execution of the contact, later the block will be eliminated from the blockchain. Although, there are many criticisms concerning the efficiency of consensus methodologies (Eyal and Sirer, 2014b; Luu et al., 2015b), some studies established that if most of the calculative power lies with the legitimate users, the consensus methodologies is secure (Garay et al., 2015a; Sompolinsky and Zohar, 2015; Li et al., 2019).

To guarantee effectiveness, the execution of Ethereum smart contracts should be appropriate. If this is not so, a malicious user may interfere with the execution. Many security risks in smart contract of Ethereum are found by implementation (Li et al., 2019; Delmolino et al.,

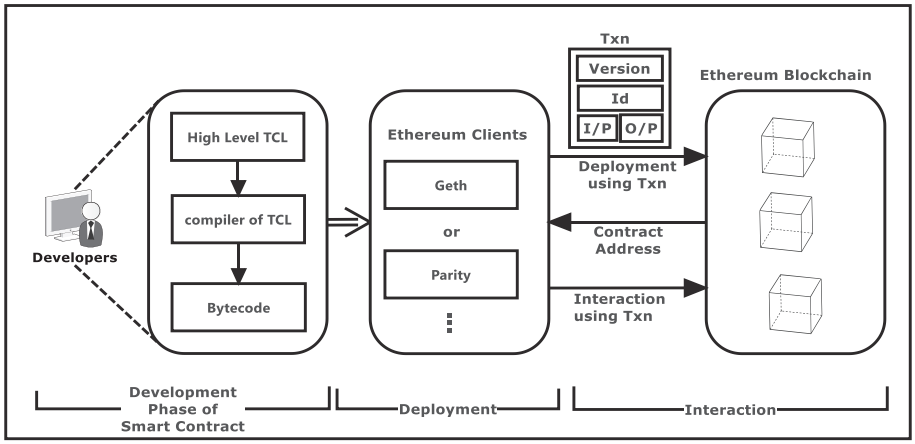

Fig. 12. Smart contract process.

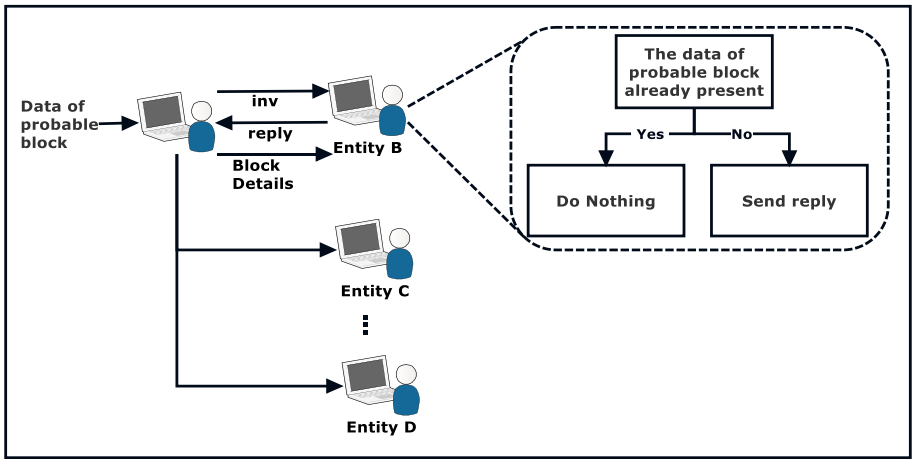

Fig. 13. An overview of advertisement-based propagation.

2016), and examination of Ethereum blockchain contracts (Luu et al., 2016a). Some of the vulnerabilities of these smart contracts have also been exploited. Many causes exist that make Ethereum's smart contract development and most of them are pertaining to solidity language. The issue with solidity is that it does not present constructs for handling domain-specific concepts, for instance the calculation stages are stored on public blockchain, thus, reordering or delay can be achieved easily. One other reason for security aspect is that all documents of well-known vulnerabilities are distributed across research papers (Li et al., 2019; Delmolino et al., 2016; Luu et al., 2016a; Anderson et al., 2016), official documents (Ethereum Wiki; Solidity), and Internet (Ethereum).

3.5. Techniques for generating the consensus

A complete entity in a blockchain, records the data of all blocks. Block propagation methodology, which is the groundwork for the construction of consensus in blockchain, can be classified into the following categories (Karl, 2016a, 2016b; Gervais et al., 2016a; Mistry et al., 2020):

3.5.1. Advertisement-based propagation

In this mechanism, once an entity A received the data of a probable block, it would broadcast *inv* message (used in cryptocurrency, like Bitcoin) to entities associated to it. Once this message is received by entity B, it will check if entity B has the data of this block beforehand, in which case nothing will be done, otherwise, a reply will be sent to entity A. On receiving the reply by entity B, entity A sends the entire data corresponding to this block to entity B. Fig. 13 shows the detailed explanation of advertisement-based propagation.

3.5.2. Sendheaders propagation

This is an enhanced version of advertisement-based propagation. In such type of propagation methodology, entity B sends sendheaders message (used in cryptocurrency, like Bitcoin) to entity A. Once entity A has received the data corresponding to a block, it transmits the block header data to entity B. Here, entity A is not required to transmit inv messages, therefore, the speed of block propagation is incremented.

3.5.3. Unsolicited push propagation

In this, when the mining of a block is accomplished, the entire block is transmitted to all the entities in the network. In the absence of sending inv message or sendheaders message, the propagation speed of block is further increased.

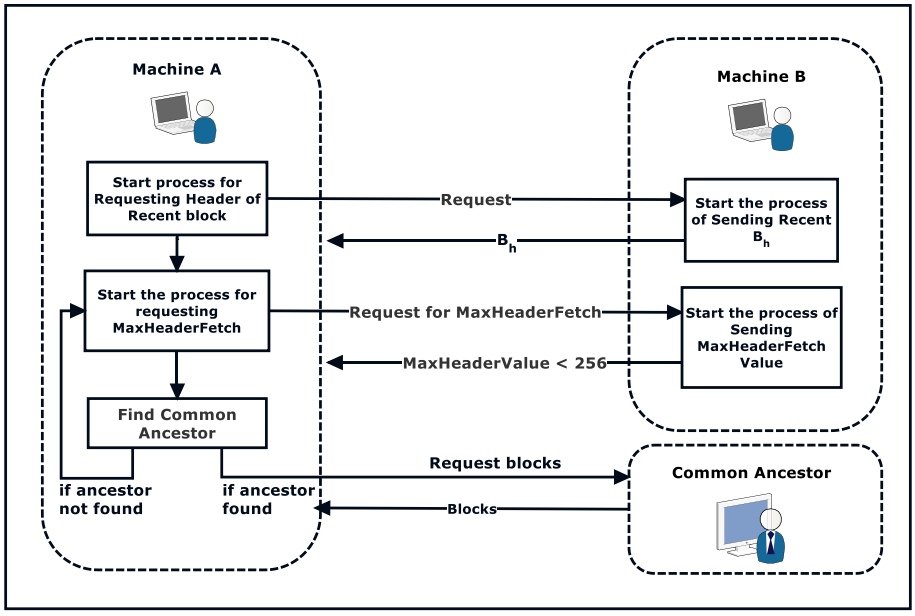

Fig. 14. The procedure for block synchronization.

3.5.4. Relay network propagation

It is an enhanced version of unsolicited push methodology. Here, a shared transaction pool is present which is accessible to all mining entities. Instead of transaction, there is a global ID, which results in reduction in the block size. This helps in the reduction of load on the network, thus supporting the increment in the speed of propagation of block.

3.5.5. Push/advertisement hybrid propagation

In this case, there is an assumption that entity A possesses n contemporaries. Here, the block is propagated to $\sqrt{n}$ contemporaries by entity A. For rest of the $n - \sqrt{n}$ associated contemporaries, entity A broadcasts hash value of the block. This methodology is implemented in Ethereum blockchain.

According to the blockchain systems, the methodologies used for synchronization of the block may differ. In case of Ethereum blockchain, entity A may appeal to entity B for block synchronization with a greater complexity (Miglani et al., 2020; Karl, 2016a, 2016b; Gervais et al., 2016a). Fig. 14 displays the process involved for block synchronization.

1. Entity A makes an appeal related to the header of recent block to entity B by transmitting GetBlockHeaders message. BlockHeaders message, which also includes appealed block header is sent as a response by entity B to entity A.
2. Entity A appeals for MaxHeaderFetch blocks for finding shared parent from entity B. By default, MaxHeaderFetch has value 256, however, the number of headers of the block that entity B sends to A might be lesser as compared to this value.
3. If A is unable to find a shared ancestor after completion of step (1) and step (2), entity A will again send GetBlockHeaders, in order to request header of one block at a time. At the same time, entity A will perform a binary search for finding the shared ancestor within its local blockchain.
4. Once entity A finds a shared ancestor, entity A requests shared ancestor for block synchronization. In this procedure, entity A demands MaxHeaderFetch blocks for each request, however, the real number of nodes which are sent from entity B to entity A might be lesser as compared to this value.

3.6. Cross-chain communication

Blockchain technology is in a very nascent stage, much like the Internet in the early 1990s. Continuous and rigorous development is going on to ensure that the use cases of Blockchain are numerous and re-useable. Until the advent of Emails-only a few people who hosted LAN connections were able to use the internet and communicate with each other. The same case is with the current scenario of Blockchain ledger usage. Today, there are many types of different blockchain networks (both public and private) to cater to the needs of people all over the world. Private chains are analogous to Intranets of 2 decades ago. When it comes to managing their information while maintaining authority, some, like IBM and JP Morgan, choose to develop on private blockchains. These chains aren't truly making use of the full potential of the technology. Instead, they are creating intranet-like solutions, which are essentially extremely inefficient databases.

But what if two different instances (networks) of Blockchain could communicate with each other? This would be advantageous in many ways. Blockchain interoperability ensures a user-friendly operation and increases adoption. Some benefits-

1. Multi-token transactions
2. Cross-chain exchange of information/receipts/databases
3. Users can work with multiple currencies at once
4. Smart contracts can be executed effortlessly

This would also solve the Atomic-Swap problem. In layman terms, the problem of an atomic swap is one where two parties can exchange data/currency (both own different coins or forms of currency) without having to trust a third party. Today, if one wants to convert INR to USD (to be used in a foreign country), one needs to trust a centralized third party (their banks) to provide them with suitable cards (linked to the Indian bank accounts) which can provide them with USD.

The problem with Cross-chain communication –

Two main principles followed by all blockchain networks are –

1. Classical Atomicity - a transaction's effects take place everywhere or nowhere.
2. Classical Isolation – guarantees that concurrent transactions cannot interfere in destructive ways

Both these properties are poorly suited to work in cross-chain communication where mutually un-trusting parties may require multiple cautious interactions to set up and execute a deal. There are multiple approaches suggested (as research as well as working methods) which aim at Cross-chains. Cross-chain deals are not atomic transactions. They solve different problems: transactions perform complex distributed state changes, while deals, by contrast, simply exchange assets among parties. While a transaction's effects must be "all-or-nothing" to preserve global invariants, each autonomous party in a deal can decide independently whether it finds an outcome satisfactory for itself. Transactions and deals make different failure assumptions: transactions usually assume parties can fail only by crashing, while deals necessarily assume parties may deviate arbitrarily from the common protocol.

For e.g. Ethereum Blockchain supports smart contracts written in solidity whereas Hyperledger composer supports smart contracts (called chaincode) which are actually written in NodeJS or Go. A complex use-case of Blockchain networks may require different types of these networks. If the project has a lot of business logic, which needs to be executed before actually making changes to state variables on a blockchain network through a transaction, Hyperledger would be the right choice as all of the NodeJS backend code could be used within the chaincode. Thus, interchange b/w these two types of networks would be very difficult because "what to include in a smart contract and how to execute transactions" is different at the core. (Experienced while writing smart contracts for Agri-Chain project).

4. Taxonomy of blockchain

Blockchain is based on distributed ledger technology, which provides a consensus authentication technique via a computer network which works in the absence of a centralized control for facilitating transactions and store the information which is produced by them. The classification of Blockchain is categorized into two classes: Permission-based and Participation-based.

4.1. Permission-based

These types of blockchains are not same as the primary concept i.e. all the members in the blockchain community can access and modify the blockchain, and that the ledger involved in the process is transparent. These are built by organisations for the purpose of confidential usage.

4.1.1. Permissioned blockchains

Companies may either build a private blockchain network or modify a primitive blockchain network. Occasionally, few organisations might join forces for construction and sharing of a patent network for simplifying the transaction process amongst them. An example of this situation is "R3 blockchain consortium", that presents a blockchain system which is used by economic institutes. Therefore, permissioned blockchain networks are proprietary in nature i.e., only certain trusted entities are permitted to audit their transactions on the distributed ledger although, everyone have the authority to read the transactions. Based on

Table 3
Comparison-based analysis between permissionless blockchain, permissioned blockchain and centralised system.

Parameters	Permissioned Blockchain	Centralised System	Permissionless Blockchain
Consensus Technique	Byzantine Fault Tolerance(BFT)	N	PoW, PoS, etc.
No. of Untrusted Writers	L	N	H
Central Control	Yes	Yes	No
No. of Readers	H	H	H
No. of Writers	L	H	H
Jitter	M	H	L
Efficiency	H	E	L
Scalability	M	H	M
Throughput	H	H	L
Verification Speed	H	L	M

L – Low, H – High, M – Moderate, E – Extreme, N – None.

confidence the entities involved have for each other, they can determine which consensus mechanism should be used by them. It is also possible to set up the permissioned blockchains in such a way so that any entity can log its transaction onto the blockchain, but only few members have the permission to read it. Few of its characteristics are analogous to permissionless blockchain like distributed storage, immutability, traceability, and redundancy of data. Example: Banking, Supply chain, Insurance, healthcare,etc.

4.1.2. Permissionless or public blockchains

These types of blockchains do not have a central control and are distributed in nature. They are unrestricted for users for participation and there are rewards involved for the process. An example for this is bitcoin network where users can perform transaction using bitcoin. These are often found to make use of consensus mechanism for avoiding malicious users from sabotaging the system.

The comparison-based analysis between permissioned blockchain, centralised system and permissionless blockchain is shown in Table 3. The jitter and efficiency in centralized system is better when compared to blockchain because in blockchains there is supplementary complexity due to the presence of consensus techniques. For instance, in Bitcoin, in 1 s, there can be only seven transactions. Without negotiating with the security measures (Mistry et al., 2020; Gervais et al., 2016b), this value can be increased to around 66. Whereas in a centralized system, for instance, transactions greater than fifty thousand can be handled. There always exists a compromise amid decentralization, i.e. performance of the scaled system due to increase in number of untrusted writers, and efficiency, i.e. performance of the system during peak time. It is essential to consider this compromise while determining whether it will be feasible to use blockchain or not.

4.2. Participation-based

As the demand of blockchain technology is rising, different variations of the systems are coming into existence. This is due to the fact that the need for the blockchain system differs from one field to another. Sometimes, the required might be of all the entities participating in the consensus process, whereas sometimes only few are needed.

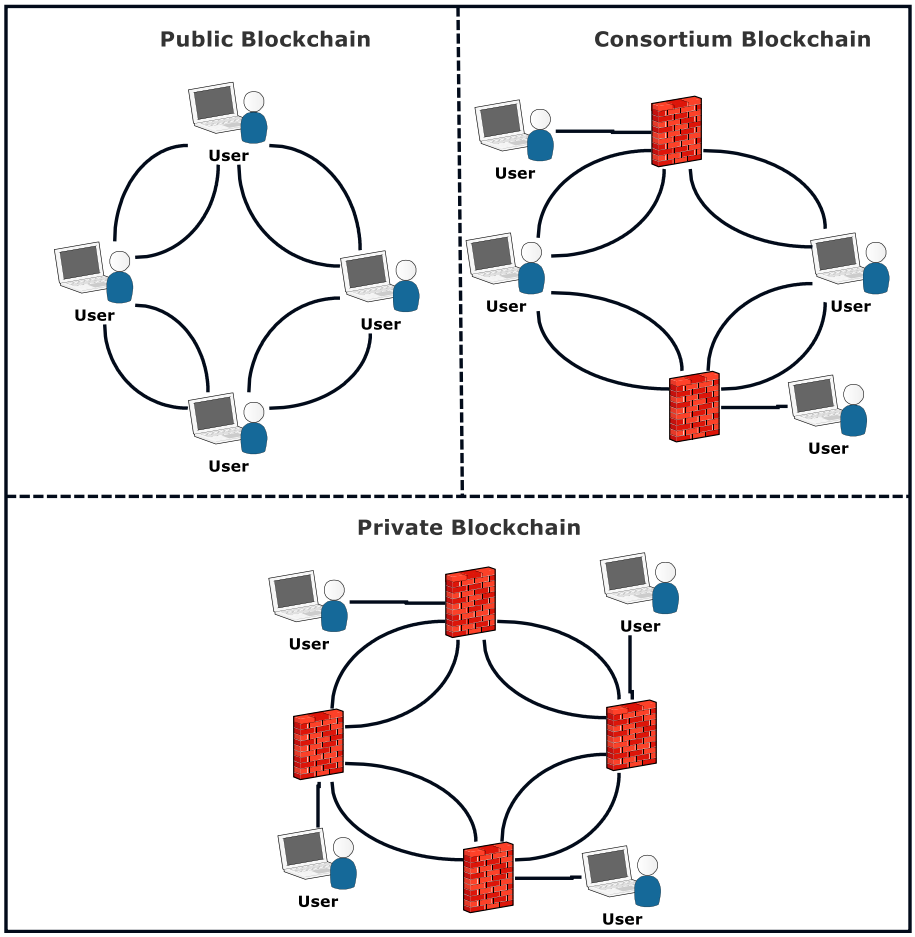

Fig. 15. Overview of the categories of the blockchain.

Table 4
Types of blockchains and their comparisons.

Parameters	Private blockchain	Consortium blockchain	Public blockchain
Throughput	High	High	Less
Participation in Consensus Process	Authentication required	Authentication required	Authentication not required
Central Authority	Complete	Partial	Decentralized
Transaction Mutability	Alteration is possible	Can be altered	Cannot be tampered
Read Access	Decided by organisation	Decided by organisation	Public
Block Authentication	Specific organisation	Selected nodes	All
Asset	Any Asset	Native Asset	Native Asset
Security	Pre-approved participants	Proof of Work	Proof of Stack
Identity	Known Identities	Pseudonymous	Anonymous
Speed	Faster	Slower	Slower
Applications	Multichain, Blockstack	Ripple, R3	Bitcoin, Ethereum, Factom

4.2.1. Public blockchains

This category of blockchain is regarded as "completely distributed". In this, any entity in the network can perform reading operation, sending transactions and viewing them being incorporated in case they are legitimate and consensus process (i.e. the procedure to determine which block gets appended to the blockchain and the contemporary state) is open for participation. Blockchain is used by crypto-economics, that is the amalgamation of economical provisions and authentication via techniques like proof of work, proof of stake. They follow the conventional notion that the extent to which a user might influence the consensus procedure is proportionate to the amount of commercial assets they could bring to operation.

4.2.2. Consortium blockchains

This category of blockchain is regarded as "moderately distributed". In this, the consensus procedure is operated by a group of nodes that are initially selected. For instance, if there are fifteen entities involved in a consortium network and ten entities out of these fifteen should authenticate each block so that the block can be validated. The read operation on the blockchain might be unrestricted or confined to the members of the network.

4.2.3. Private blockchains

In this, write operation is restricted to one central organization and read operation either unrestricted or confined to an arbitrary range. Applications of these kind of blockchains may involve database management, review, etc which are intramural (for a company). Therefore, open readability might not essentially be required in several cases. Although in some cases public assessment may be necessary desired.

Given that, public blockchain is opensource, it has the capability to draw the attention of several users. Gradually a number of public blockchains are coming into existence. Considering consortium blockchain, it can be applicable in the field of business. Presently, Hyperledger is being used to develop commercial consortium blockchain model and Ethereum had bestowed tools, which help to develop consortium blockchains. With respect to private blockchain, their capabilities are generally utilized by various companies who implement them because they find them to be efficient. The categories of blockchain are displayed in Fig. 15. According to the requirement, the blockchain systems are divided into the following three categories and their comparison and analysis are presented in Table 4.

- ***Throughput***: Considering a public blockchain network, the number of nodes involved is high, therefore it takes sufficiently large amount of time for propagation of not only transactions but also the blocks. Considering the issue of network security, limitations on public blockchain is high and this results in decline in transaction throughput, increase in latency. In case of consortium and private blockchain, since there are few authenticators they are regarded as more efficient.
- ***Participation in Consensus Process:*** In public blockchain, any entity in the network can collaborate in its consensus process. Whereas in consortium and private blockchain, an entity requires authentication in order to participate in the consensus process, i.e., they are permissioned.
- ***Central Authority***: This is considered to be the primary distinction existing amongst the three categories of blockchain. The public blockchain works in the absence of a central authority whereas consortium is partly centralised in nature. In case of private blockchain, it is completely controlled by a central authority since it is governed by one entity.
- ***Transaction Mutability***: Blockchain is a decentralized network, hence the transaction involved are saved in varying entities in the computer network. Therefore, it becomes almost unfeasible to modify the public blockchain. Nevertheless, if there are some the influential entities who wish to modify the blockchain, the consortium or private blockchain can be altered.
- ***Read access***: In case of a public blockchain, it is open and thus transactions are accessible to everyone. Whereas when the blockchain is private or consortium the approval for performing read operation is dependent on the network. In these, the organisation determines if the information available is unrestricted or confidential.
- ***Block Authentication:*** For authentication of the block, all entities, in the public chain, participate. As for consortium blockchain, few designated entities perform validation. In case of private blockchain, this process is completely managed by single authority who can decide the ultimate consensus.

4.3. Methods for selecting blockchain

In case, we have multiple entities who neither have trust on each other nor want a central authority although, they wish to communicate and perform some transactions, one may use permissionless or permissioned blockchain. Table 5 helps in determining the blockchain to be

Table 5
Feasibility of Permission and permissionless blockchain.

Blockchain Type	Requirement to Save State	Numerous Writers	Availability of Trusted Third party	Knowledge of all writers	Confidence on all writers	Need for pubic verification
Permissionless Blockchain			Not always online	×	-	-
Public Permissioned Blockchain			Not always online		×	
Private Permissioned Blockchain			Not always online		×	×
	×	-	–	-	-	-
Don't Use Blockchain		×	–	-	-	-
			Always Online	-	-	-
			Not always online			-

chosen. If we do not have to save the data, we do not require a database. Since, blockchain sometimes also behaves as a database, it is not required in this case. There may be one or more than one entity involved who are responsible for writing the state of the system. This implies that writer refers to a node that has access for write operation in a database or for consent for contestants in blockchain. In case, data is not required to be recorded, database is not required, thus, blockchain is not needed. Likewise, when one write is present, supplementary guarantee is not provided by blockchain and a conventional database is preferable, since it gives superior high efficiency and low latency.

When a trusted centralized authority (TCA) is present, two things can happen. Firstly, writing operation is entrusted to it and it may work as an authenticator for the evolution of states, provided the TCA is always available. Secondly, it may work as an authenticator for permissioned blockchain, in which each writer of a node must be approved, provided TCA is generally unavailable. In case, mutual trust exists among the writers, i.e. no illegitimate writer is present, the prime solution might be a database having communal write permission. In the absence of mutual trust among writers, permissioned blockchain should be used. Subject to requirement of public verification, either an entity is permitted for reading the state that occurs in public permissioned blockchain or group of entities who want to perform the read operation that occurs in private permissioned blockchain may also be limited (In case, the group of the writers is dynamic and is known to members, e.g. in Bitcoin, permissionless blockchain should be used).

4.4. Primary features of blockchain

Until now, we have focussed on the architecture of the blockchain, followed by procedure of expansion of the existing blockchain. Further, we reviewed on the mechanisms involved in this remarkable technology. In our prior discussion, we focussed on the distributed nature of blockchain. Fig. 16 shows the features of blockchain and let us further elaborate on them.

- ***Decentralization:***In the field of centralized systems, every transaction requires authentication by a trusted third party. This resulted in restriction in the price and the execution of the servers. However, in blockchain, the central authority is not required and consensus methodologies in blockchain can sustain the consistency of information in distributed environment.
- ***Persistency:***In blockchain, authentication of transactions is very fast, and illegitimate transactions will not be incorporated by legitimate miners. Omission or rollback of transactions is not plausible if they are incorporated in the chain. If blocks contain illegitimate transactions, they will be recognised instantly.
- ***Anonymity:***In blockchain, every entity has a generated address, using which it can communicate with each other. These addresses do not disclose the original identity of entities involved. Blockchain does not assure the flawless privacy protection because of some inherent constraints.
- ***Auditability:***Cryptocurrency which uses blockchain (in this case Bitcoin) recorded data of an entity's assets according to UTXO model, that is Unspent Transaction Output model (Nakamoto, 2008c). A transaction must indicate to a preceding unspent transaction. As soon as, the present transaction is stored in blockchain, state of the indicated unspent transactions is substituted to spent. Thus, the transactions can easily be easily authenticated and traced.
- ***Real-time records:*** Decentralized ledgers must be updated as soon as transactions happen, or other proceedings take place, with the help of some software which automate the process. This certify that every network entity holds its own real time record of its transactions, that in turn decreases the possibilities for malicious activity. The computerized method and distributed record storage increases productivities and causes reduction in cost.
- ***Immutability:*** In Blockchain, immutable records are created which offers profit, however, it may cause authoritarian peril for few entities. Authorities may be provided with authorization for the access of all transaction histories if any investigation takes place which involve transactions stored in a blockchain. This makes it problematic for the entities that claim shortage of transactions. Moreover, to maintain permanent log of some transactions as well as entities, a blockchain may involve data confidentiality protocols, mainly as authenticators progressively emphasizing on safeguarding customer's confidentiality.
- ***Vulnerabilities:*** Blockchain networks is considered to be the primary focus of the malicious users. Although, blockchains have not yet been hacked or modified efficiently, the organisations and technologies related to it are reported to be affected. The spectrum of attacks ranges from service interruptions to thievery of confidential information and valued assets. Nevertheless, the distributed architecture of blockchain technology makes the network more robust attacks or modifications.
- ***Tax implications:*** Blockchain transactions which involve valuable assets, which can generate unforeseen tax penalties that depends the way in which the concerned tax experts deal with digital currency. For example, IRS (US Internal Revenue Service), considers cryptocurrency as assets, that signifies that a transaction might develop the necessity to identify profit or loss when a cryptocurrency is transferred.

4.5. Applications of blockchain

As mentioned earlier, there are several fields in which the blockchain technology is being applied. In this section, we will see in detail the various domains in which blockchain technique is implemented as well as discuss about the work done by various authors in their corresponding field using blockchain. Fig. 17 shows application of blockchain.

4.5.1. Finance

Economic Services: The advent of various blockchain systems like Bitcoin and Hyperledger has resulted in huge influence on respective traditional system. Blockchain may change the entire banking system.

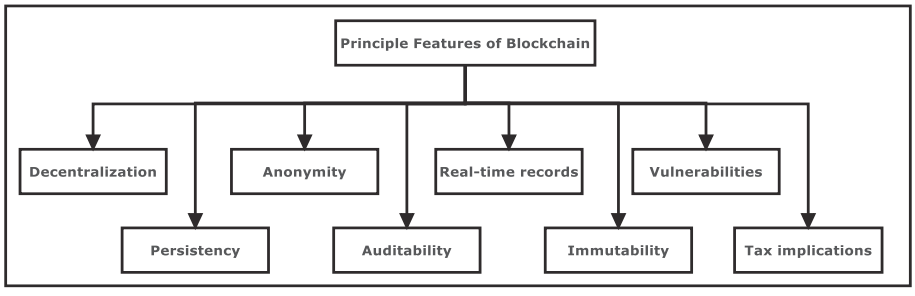

Fig. 16. Primary features of blockchain.

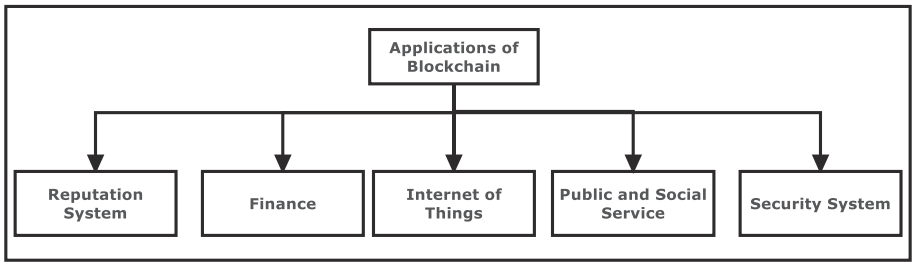

Fig. 17. Applications of blockchain.

Blockchain can be harnessed in several fields like settlement and clearance of economic assets etc. Some real business cases such as collateralisation of various financial results can use blockchain to decrease costs as well as risks. Microsoft Azure (2016) and IBM (2016) have begun to present Blockchain-*as*-a-Service.

Organisation Transformation: Blockchain may support traditional enterprises to accomplish the organisation transformation. For instance, customary POs (Postal Operators) is a bridge between traders and clients, digital currency and blockchain may support POs to expand their functions which may include economic and non-economic services. Battista et al. (Di Battista et al., 2015; Bose et al., 2019), (Bodkhe et al., 2019)said that every PO can release postcoin (a type of coloured coin of Bitcoin). As POs are considered trustworthy by people, postcoin can exist fast with their retail network which is quite dense. Moreover, they also tell that blockchain technique proposes business openings for POs in supply chain management, identity services as well as device management.

Peer-to-Peer (P2P) Economic Market: Blockchain may support to construct a peer-to-peer economic market with security and reliability. Noyes surveyed various methods to combine P2P methodologies and multiparty calculation rules to construct a Peer-to-Peer financial Multiparty Computation (MPC) market. These MPC market permit offloading computational jobs on anonymous peer-processors network.

Hazard Management: It has an important role in economic technology (FinTech) and after the advent of blockchain, their combination can give better result. Pilkington presented a hazard-management framework, where blockchain technique helps in analysing investment hazards in Luxembourgish situation. Stockholders, holding securities via custodians, also face failure hazards. Blockchain also help to decide investments and securities fast and avoid looking at longstanding consideration. In Micheler and von der Heyde (2016), it is stated that combination of blockchain with some new system and may decrease risk and attain transactional safety. Moreover, smart contract, based on blockchain, enable the DAO (decentralized autonomous organisations) to get involved in business work associations. Norta et al. proposed a DAO-GaaS conflict model to defend consistency rules.

4.5.2. IoT (internet of things)

It is projected to assimilate the smart objects into internet and delivers several services to clients. Blockchain technology has the potential to enhance the IoT sector.

E-business: In the year 2015, Zhang and Wen projected a novel IoT digital business prototype and realized that the smart property transactions have blockchain and smart contract as their base. In this innovative prototype, DAC (distributed autonomous corporations) is accepted as distributed transaction element. The clients transact using DACs for acquiring coins and exchanging sensor information in the absence of a third party.

Privacy and Security: This is another issue for IoT industry and can be improved using blockchain. Hardjono and Smith projected a methodology for safeguarding the privacy for appointing an IoT device inside a cloud system. It supported the design to certify its production provenance in the absence of verification by a third party as well as permitted to enrole anonymously. In the year 2015, IBM revealed the evidence for ADEPT (Autonomous Decentralized P2P Telemetry), which builds a decentralized system using blockchain technology. In ADEPT, home appliances will have the potential to recognize operational issues and will independently fetch the software updates.

4.5.3. Social services

There are various methods in social services which utilizes the capabilities of blockchain. Few of them are mentioned as follows:

Land Registration:In this the data related to the land like related rights like physical status may be logged and broadcasted on the blockchain network. If any alterations are performed on the corresponding land, for instance transfer or mortgage establishment, it may also be logged and operated on the blockchain. This will in turn improve the effectiveness of social services.

Use of Non-Conventional Energy Sources. Gogerty and Zitoliprojected the '*solarcoin*' for inspiring the use of non-conventional energies. Solarcoin is a type of electronic currency which give prize to the manufacturers of solar energy. As long as the miner generate solar energy, they will receive solarcoins, as incentive, by solarcoin foundation.

Teaching and learning. If teaching and learning process is considered, blockchain technique has wide scope in online education market also. Devine projected the idea of learning with the help of blockchain. In such kind of learning, blocks can be packaged and located in the blockchain network by the instructors and the learning accomplishments may be considered as coins.

Free-speech right: The blockchain technology may be utilized for securing internet structure like identities and DNS. For instance, Namecoin is a novel technology which is open-source and enhances decentralization, privacy and security, speed of DNS and identities, as well as censorship resistance. Since it makes the internet more censorship resistant, therefore it safeguards free-speech right.

Some other social services of blockchain might be registration of marriage, income taxation, and patent management. Blockchain may also help to reduce paper work because with the advent of the recent social services (blockchains embedded), digital signatures may substitute seals that have to be attached on official documents.

4.5.4. Reputation system

A user's reputation might be based on his past transactions and communications with the public. Recently, a several instances have come up which states falsification of individual's reputation information. For instance, in e-commerce, a number of service-providers register large number of false customers for achieving a greater reputation. Blockchain has the potential to resolve this issue.

Academics:Domingue and Sharples projected a decentralized system based on blockchain for educational record and reputation. Initially, every institute and staff were awarded with educational reputation currency as prize. An institute can honour a worker by giving them some reputation records. The changes in reputation can easily be sensed as transactions are recorded on the blockchain.

Web Society: Carboni projected a blockchain-based model for reputation. In this, a voucher is signed only if the client is content with the services provided and is willing to bestow a positive feedback.

Subsequently, the service-provider will need to acquire additional 3% of payment as voting fee to its network for discouraging Sybil attack. This voting fees is used for calculating the reputation of the service provider. Dennis and Owen projected a novel reputation system which could be applied to several networks. They constructed a new blockchain for recording one-dimension reputation value (that is, either 0 or 1) from the accomplished transactions. For instance, in file sharing, Node Atransmits a file to node B. On getting the delivery of the file, Node Btransmits a transaction which comprises of score, file has as well as private key node B for authenticating the identity. Subsequently, the mining nodes contact node *A* and node *B for* confirming that the transaction takes place without any malicious activity. As the transactions are recorded on the blockchain network, the probability of reputation records being altered is almost negligible.

4.5.5. Security and privacy

Security Improvement: Blockchain has the potential to provide assistance for enhancing the security of decentralized networks. Charles projected a new *anti*-malware environment known as BitAV, where the clients could distribute various virus patterns onto the blockchain network. Therefore, fault tolerance of the system is improved. Noyes discusses that BitAV improved the speed of scanning as well as enhanced the reliability for faults. Blockchain technique may also enhance security infrastructure reliability. For instance, PKIs (public key infrastructures) are generally prone to single point of failure either because of software and hardware issues or attacks. In Axon, besides enhancing conventional PKIs reliability, blockchain may be utilized to design a privacy-conscious PKI.

Privacy Protection: Our personal data is susceptible to malware as well as service providers, who collect the data and record it on a central repository, which is vulnerable to malicious users. The decentralized nature of blockchain can solve this problem and enhance the security of information of the customers. Zyskind et al. projected a blockchain-based distributed personal data management system which guarantees ownership of the users for their data. The following three privacy issues can be resolved: (i)data possession (ii)data clarity and auditability and (iii) fine-grained access control.

4.5.6. Security and privacy

- ***Security improvement:*** Blockchain has the potential to provide assistance for enhancing the security of decentralized networks. Charles (in Noyes, 2016) projected a new *anti*-malware environment known as BitAV, where the clients could distribute various virus patterns onto the blockchain network. Therefore, fault tolerance of the system is improved. Noyes (in 2016) discusses that BitAV improved the speed of scanning as well as enhanced the reliability for faults. Blockchain technique may also enhance security infrastructure reliability. For instance, PKIs (public key infrastructures) are generally prone to single point of failure either because of software and hardware issues or attacks. In Axon (2015), besides enhancing conventional PKIs reliability, blockchain may be utilized to design a privacy-conscious PKI.
- ***Privacy protection:*** Our personal data is susceptible to malware as well as service providers, who collect the data and record it on a central repository, which is vulnerable to malicious users. The decentralized nature of blockchain can solve this problem and enhance the security of information of the customers. Zyskind et al. (in 2015) projected a blockchain-based distributed personal data management system which guarantees ownership of the users for their data. The following three privacy issues can be resolved: (i)data possession (ii)data clarity and auditability and (iii) fine-grained access control.

4.5.7. Secure blockchain solution in cloud computing

Privacy leakage in cloud computing environment may have negative impacts. Blockchain technology provides the clients with anonymity. Amalgamation of blockchain and cloud computing may result in enhanced security measures. Fig. 18 highlights the working of secure bitcoin protocol.

Installation of e-wallet is performed for using blockchain technology, which if improperly removed, deduces the clients' information. For solving this issue, a solution is proposed in (Blockchain) which ensures secure installation as well as deletion of e-wallet. A client will have to install e-wallet software on his system for secure use of bitcoin. After completion of the installation, the public key to the corresponding platform is transmitted to the e-wallet. E-wallet further transmits a certificate which was dispensed in the development phase of the platform. Diffie–Hellman methodology is utilized for exchanging keys between e-wallet and the platform. On the arrival of request for a transaction (involving bitcoins) by the user, a ledger data which contains time stamp data between e-wallet and platform are encrypted with the shared key and sent. On the arrival of the request for disposal, the certificate of the user is obtained and deleted from e-wallet. Finally, acknowledgment is transmitted for confirming secure disposal. Moreover, the related files are also removed in order to remove the remaining information securely.

5. Blockchain platforms and ITS security challenges

There are several applications, of blockchain but it is seen that it is primary used for digital currencies. In this section, we will be primarily focussing on discussing few blockchain platforms for highlighting

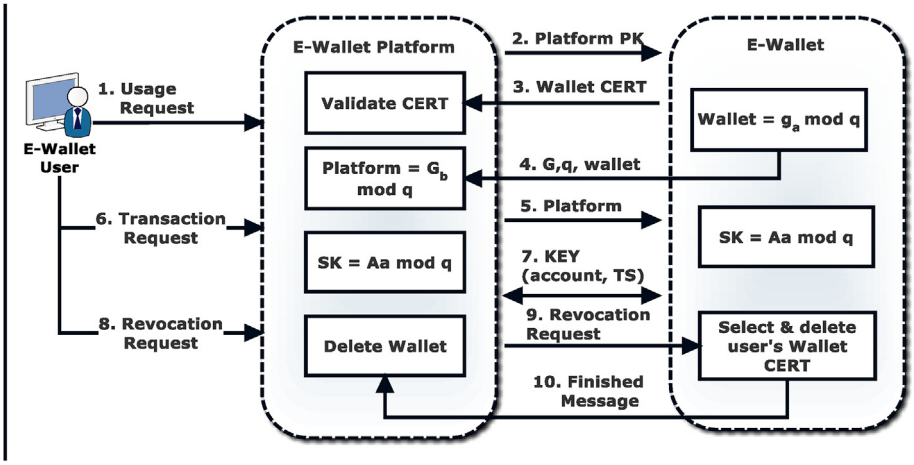

Fig. 18. Secure Bitcoin protocol.

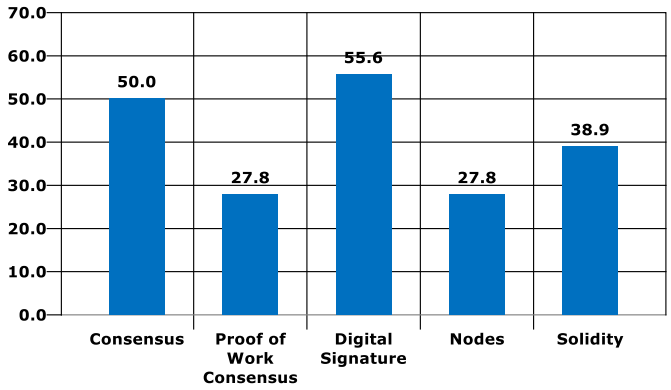

Fig. 19. Various platforms in which blockchain are being used.

technical dissimilarity and tactics which are being applied. Fig. 19 highlights the taxonomy of blockchain platforms. It is to be noted that we are not endorsing any of the mentioned platforms, moreover it should be not be interpreted as catalogue of the most prevalent platforms.

5. 1Cryptocurrencies

Several blockchain applications are focused on transfer of digital currency from one user's account to the other. In this, we will see a number of instances of this type of blockchain applications.

5.1.1. Bitcoin (BTC)

Bitcoin refers to a cryptocurrency system that we have reviewed as the developer of blockchain. In this, latest blocks are produced in every gap of 10 min with the help of SHA-256 hashing for connecting them with one another. Here, we use a PoW methodology, in which miners should discover a nonce to incorporate in their own block so that block hash is lesser than previously computed complex value. The complex value is increased or decreased in order to accomplish the 10-min target to create a block. In the past, separate computer system worked as miner and published blocks; presently Bitcoin needs large data centres, dedicated hardware, or several entities who work collectively in mining pool for winning the challenge for publishing blocks. When Bitcoin is used, transaction fee payment optional because miners get large portion of their assets by block publication. Therefore, this payment is planned to be less for every transaction, however over the years this payment has increased because of a considerable amount of backlogged transactions.

If high transaction fee is paid, the transaction may be given higher priority in order to get appended in the blockchain. In the beginning, miners received fifty Bitcoin for every block, after a few blocks they had to pay half of this value. For instance, in July 2016, 12.5 Bitcoins was the reward to mine a block. According to Bitcoin rules, this value was halved every 210,000 blocks, moreover, the value reduced to zero after the production of 21 million Bitcoins (Gupta et al., 2019; Herrera-Joancomart et al., 2015; Valenta et al., 2015). At this stage, the mining of Bitcoin will be continued, however reward for an entity who mines was entirely drawn from the transaction fee. Every Bitcoin transaction has program written in Script language. This program states the transaction and is not comprised of loops, moreover, it is extremely limited with respect to functionality, that is, the programs are not Turing complete. Contemporary transactions of Bitcoin utilize a minor part of Script's characteristics. Realistically, a large number of transactions of Bitcoin makes use of any one of the few patterns of program for the transfer of assets among entities.

5.1.2. Bitcoin Cash (BCC)

In July 2017, around 80%–90% of Bitcoin computing authority voted for including SegWit, that is, Segregated Witness, in which transactions are divided in two parts: (i) transactional data (ii) signature data. This helped to decrease the quantity of data, that has to be authenticated in every block. The activation of SegWit led to production of a hard fork. The miners as well as the users who were unwilling to go through transformation began to call the primary blockchain of Bitcoin as BCC, that is Bitcoin Cash. Therefore, Bitcoin Cash is initial blockchain whereas Bitcoin is just a fork. After the hard fork was caused, entities had approach to the equal quantity of assets on the Bitcoin chain as well as Bitcoin Cash chain.

5.1.3. Litecoin (LTC)

Litecoin (LTC) has similarity with Bitcoin, however, it targets to decrease the confirmation time. LTC was the one who constructed Seg-Wit, which splits the transactions in two parts and hides the block size which was increased (Hertig, 2017). Here, “witness” signature is detached from Merkle tree. LTC makes use of Scrypt algorithm to hash whereas Bitcoin uses SHA-256. Since Scrypt algorithm has high memory consumption, it is hard to solve as compared to SHA-256. This results in increasing the creation difficulty level of custom ASICs, that is, application-specific integrated circuits. The peak amount of assets that can be mined here is high, that is, 84 million. LTC is similar to Bitcoin, and has greater number of transactions, however it is not constructed to substitute Bitcoin (Litecoin Project).

5.1.4. Ethereum (ETH)

This is the blockchain platform that aims to provide smart contracts, which are codes present on blockchain and can be approached by the users of Ethereum. These are capable of receiving and transferring of assets, at the same time perform random calculation. If designed appropriately, smart contracts may behave as trusted intermediary in case of financial transactions as its program is public as well as immutable. Ethereum uses a Turing complete language for transaction programming. Here, the miners get assets by mining as well as transaction fees. There is a theory in Ethereum known as “gas”, which is utilized to fuel the transactional calculations and is usually about 1/100,000th of an Ether. Each transaction uses gas while executing, and the designer of a specific transaction should give adequate gas, otherwise transaction execution is terminated. Here we have a limited amount of gas for each smart contract (at present it is three million) to avoid computationally costly programs to be proposed to Ethereum miners. This is done since all the miners should execute transactions parallelly (Wood). If a transaction is submitted to Ethereum contract, it will cause a program to execute parallelly on a miners’ system. Thereafter, the user who publishes the subsequent block also records the resulted state of contract on the blockchain.

5.1.5. Ethereum Classic (ETC)

Ethereum experienced a DAO hack (Wong and Kar, 2016b), where a malicious user withdrew about $50 million. Subsequently, a hard fork was produced by Ethereum Foundation which was called as Ethereum Classic. This was done to move the thieved assets to the state prior to the attack. Entities who possessed Ethereum earlier the DAO hack now owned equal quantity of assets in Ethereum Classic. The cause of its existence is that many Ethereum users did not accept the fork because of philosophical reasons (Pearson), which included a protocol that blockchain should not be altered and were stubborn to use Ethereum blockchain, which was unforked. The mining and the software in Ethereum Classic is almost same as Ethereum. The only difference is that Ethereum is more popular, even though it is a fork.

5.1.6. Dash (DASH)

This is a cryptocurrency which aimed to provide quicker transactions. It utilizes a network known as “masternode” and is capable of making transactions in 4 s (What Is Dash?). With the help of hash and PoW for each block, itutilizes deterministic ordering for masternodes. In order to become a masternode, one necessitates 1000 Dash collateral. This makes it extremely costly and almost infeasible to govern 50% or greater part of the blockchain network (Duffield, Diaz). The collateral needed for

masternodes increases the issue of untrustworthy entities in a decentralized network. Unlike most of the blockchain platforms, Dash utilizes x11 as hashing algorithm. This comprises of utilizing eleven SHA-3 contestant algorithms, and every hash is put forward to the succeeding algorithm which is existing in the chain (Duffield, Diaz). Thus, it becomes very difficult to generate an ASIC, which aims to resolve these hashes in the hardware.

5.1.7. Ripple (XRP)

Ripple is a cryptocurrency and the same name is used for the related payment network where this currency is being transmitted. It aims to construct on the methodology of Bitcoin as well as link various payment systems to one another. It has an unchanging supply of 0.1 trillion Ripple, from which half is selected for transmission (Introduction, 2013; Brown, 2013; Sharma et al., 2018). It is effortless for the clients to connect to the network because they do not haveto download complete blockchain. Moreover, since cost of every transaction is a small quantity of Ripple, no mining payment exist to run the server. Thus, there does not exist any mining entity or pools; instead, near 1/1000th of a cent from every transaction is demolished (Brown, 2013; Sharma et al., 2018). XRP is not constructed for providing anonymity, however it has properties which provide privacy, for example, utilizing proxied gateway are used for payments.

5. 2Hyperledger

This is a set of projects whose objective is to generate open-source, enterprise-grade, decentralized ledgers (Linux Foundation). Linux Foundation hosted and supported the Hyperledger Projects. Though, Linux Foundation hosted Hyperledger projects, varied sources developed and contributed to every project. Hyperledger project consist of many projects and each project provides blockchain platform for solving a particular problem.

5.2.1. Hyperledger Fabric

It is a permissioned and modular blockchain, which can execute smart contracts (known as Chaincode) (Cachin, 2016). Initially, Digital asset and IBM contributed the Hyperledger Fabric to Hyperledger Project.

5.2.2. Hyperledger Sawtooth

Hyperledger Sawtooth uses PoET as the consensus methodology and is modular decentralized ledger. In PoET, each participating entity demands a hardware enclave for “wait time”. A hardware enclave is a protected and trusted feature existing on some hardware and it will allocate wait times arbitrarily. An entity who gets the least time is responsible for creating the succeeding block in the sequence. A hardware enclave supporting hardware has tightly coupled the use of Hyperledger Sawtooth. Originally Intel contributed to Hyperledger Sawtooth.

5.2.3. Hyperledger Iroha

The Hyperledger Iroha uses blockchain technology for knowing its clients. It permits organisations to share information and handle individuality. Originally Colu, Soramitsu, Hitachi and NTT Data contributed to Hyperledger Iroha.

5.2.4. Hyperledger Burrow

This blockchain platform is permissioned and smart contract-active and this accepts the smart contract code which are based on Ethereum. Initially, Monax and Intel contributed to Hyperledger Burrow.

5.2.5. Hyperledger Indy

Hyperledger Indy is one of the independent platforms which provides trusted transactions and reliability. It provides provisions for user-controlled swaping of certifiable rights about recognizing the data, and revocation models. Hyperledger Indy provides three security properties: (i) DIDs (Decentralized Identifiers) (ii) pointers to off-ledger sources – to avoid writing personal data on the ledger, (iii) zero-knowledge-proofs. Sovrin Foundation is sponsoring Hyperledger Indy. Table 6 clearly displays the comparisons among Hyperledger, Ethereum and Bitcoin. These cyptocurrencies can be utilized in different network environments for mining Bitcoin where large amount of resources are required because of the PoW methodology. However, some substitutions are present like PoS. With PoW, the possibility to mine a block is dependent on the miners and the amount of work done by them. Though, Bitcoin API is utilized in different network services to develop services, it is very challenging for the users to utilize its capabilities.

Table 6
Comparison and contrast between some cryptocurrencies and hyperledger.

Parameter	Hyperledger	Ethereum	Bitcoin
Language	Java, Golang	Python, Golang	C++
Cryptocurrency Used	None, but can be implemented when required	Ether	Bitcoin
Consensus Methodology	PBFT	PoW (Ethash)	PoW (SHA 256)
Smart Contract & Language	Yes (chaincode)	Yes (Solidity)	None
Network Type	Permissioned	Public	Public
Confidentiality	Confidential Transactions	Transparent Transactions	Transparent Transactions
Business Platforms	Preferred platform for B2B businesses	Platform for B2C businesses and generalized applications	Preferred platform for B2B businesses
Mode of Peer Participation	Private and Permissioned Network	Public/Private and Permissionless Network	Public/Private and Permissionless Network

5.3. MultiChain

Itis a blockchain platform that allows everyone to setup, configure, as well as execute a blockchain. The blockchain can be a private, consortium, or public blockchain. Thus, it is open source. This blockchain platform is actually a fork of the Bitcoin cryptocurrency, however it has several alterations. The clients can decide if they wish to have related cryptocurrency, and consensus method. By default, MultiChain is a private-permissioned blockchain which makes use of round-robin consensus. This says that any entity who sets up the blockchain will acts as an manager and primary entity; other entities involved should guide their corresponding MultiChain blockchain users to the primary entity, and the manager should permit them. MultiChain Stream (Greenspan, 2016) an exclusive feature; these are defined as “shared immutable key value time series databases” and are recorded on blockchain.

5.4. Security issues and challenges in blockchain systems

In the previous sections we have focussed on thefoundations on blockchain. We learnt about the Blockchain technology and various platforms in which it is applied in the current era. Although blockchain is an innovative and ground-breaking technology which has the potential to change several applications, it is accompanied with a number of issues. Few of the corresponding issues will be discussed in this section.

5.4.1. Challenges

With the growing useblockchain technology, various technical challenges and drawbacks have come forward. Swan (Di Battista et al., 2015; Bose et al., 2019) came up with the following technical challenges and drawbacks for the acceptance of this technology: (i) usability (ii) Versioning, hard forks, multiple chains (iii) Size and Bandwidth (iv) Privacy (v) Security (vi) Wasted Resources (vii) Latency (viii) Throughput. Table 7 shows various challenges in the blockchain system.

Table 7
Blockchain challenges and its state of art.

(ii)Versioning, Multiple Chains and Hard Forks: If the chain, in blockchain network, comprises of a smaller number of entities, then the probability of 51% attack is high. In addition to this, when the chains are divided for managerial or versioning objective another problem appears.

(iii) Size and Bandwidth:The Bitcoin blockchain size has been increasing from the time when it was created, i.e., in 2009, and is expected to reach about 197 gigabytes by January 2019. When throughput grows to height of VISA, the blockchain size can increase 214 PB every year. The community Bitcoin believes that size of a block is approximately 1 MB, and one block is constructed in 10 min (Koshy et al., 2014; Beikverdi and Song, 2015; Bodkhe et al., 2019). Thus, there exists is a constraint for number of transactions that can be managed (approximately 500 transactions in a block) (Kaur et al., 2018; Barkatullah and Hanke, 2015; Zhang et al., 2015). In case Blockchain is required to manage greater number of transactions, the size as well as bandwidth challenges should be resolved.

(iv) Privacy and Security: Currently, blockchain does have a probability of 51% attack, in which one entity will have complete control over major portion of mining hash-rate of the network. Moreover, it will have the capability to alter the blockchain. In order to overpower this challenge, more research is required in the field of security.

(v)Wasted Resources: For mining Bitcoin, large amount of resources are required because of the PoW methodology. However, some substitutions are present like PoS. With PoW, the possibility to mine a block is dependent on the miners and the amount of work done by them (Shojafar et al., 2016). Whereas, in PoS, the resource that is compared is the amount of Bitcoin a miner holds (Shojafar et al., 2016). The challenge with wasted resources has to be resolved for having more productive mining in the Blockchain.

(vi) Latency: For providing security for a block in Bitcoin transaction, approximately 10 min is required to accomplish one transaction. For achieving efficiency in security, greater amount of time is spent on one block, since it needs to overshadow the price of double spending attack that is successful expenditure of coins more than one time (Decker and Wattenhofer, 2013). Double spending is avoided by Bitcoin by authenticating every transaction which is appended to the blockchain, in order to guarantee that inputs involved in a particular transaction is not spent before (Decker and Wattenhofer, 2013; Andrychowicz et al., 2015; Bansal et al., 2019), as a result increasing the latency. In VISA, a transaction processing networks, only few seconds are taken to accomplish a transaction that is a greater lead as compared to Blockchain.

(vii) Throughput: Currently, the throughput of Bitcoin network is increased to 7tps (i.e. transactions per second). However, throughput of VISA and twitter is 2,000tps and 5,000tps correspondingly. If the frequency of blockchain transactions grows to the levels of VISA and Twitter then, blockchain's throughput will have to be upgraded.

Challenges	Types	Subdivisions	State of Art
Usability	End User support	Blockchain network analysis	Visualization of (1) bitcoin flow (Di Battista et al., 2015) (2) bitcoin user group (Spagnuolo et al., 2014)
		Transaction Validity check	(1) Audit software for exchange participants (Decker et al., 2015) (2) Reputation Rating system (Vandervort et al., 2014)
	Developer Support	Nil	Solution Not addressed
Versioning, multiple chains and Hard Forks	Nil	Nil	Solution Not addressed
Size and Bandwidth	Nil	Nil	Solution Not addressed
Privacy	Definition of anonymity in digital currency	Nil	Definition framework for anonymity (Meiklejohn et al., 2015)
	Deanonymization by linking transactions	Nil	(1) Composite Signature (Saxena et al., 2014) (2) Transaction Mixing protocols (Valenta et al., 2015; Ruffing et al., 2014; Androulaki et al., 2014; Ziegeldorf et al., 2015)
	Deanonymization by linking Bitcoin address and IP	Nil	(1) Transaction Mixing protocols (Herrera-Joancomart et al., 2015)
	Analysis of anonymity	Nil	(1) Reverse Engineering method (Moser et al., 2013) (2) P2P network analysis Framework (Feld et al., 2014; Koshy et al., 2014)
Security	51% attack	Market-based centralization on mining power	Solution Not addressed but talked about in paper (Beikverdi and Song, 2015)
		It is not safe to have 51% computation power	The protocol which is used to limit the computation power by one third (Garay et al., 2015b)
		Selfish mine attack	The protocol which is used to limit the computation power by one fourth (Eyal et al., 2014)
		Verifier's dilemma	Protocol for regulating the total quantity of work done on authentication (Luu et al., 2015c)
		Blockchain Forks	Protocol for decreasing the propagation delay (Decker and Wattenhofer, 2013)
	Data Malleability	Greater likelihood of alteration of Bitcoin transactions and illegitimate conduct of current wallets	Protocol for malleability-resilient refund transaction (Andrychowicz et al., 2015)
	Security Incidents	Currency exchange and huge mining pools are the main marks of DDoS attack	Mentioned in (Vasek et al., 2014) but solution not addressed
		Various categories of security breaching (e.g. DDoS, private account hacking)	Security counter-measures (e.g. Bitcoin H/W wallet etc.) (Lim et al., 2014)
		Various categories of bitcoin financial scams, mining scams, scam wallet etc.)	Mentioned in (Vasek et al., 2015) but solution not addressed
	Authentication	Identical key production of elliptic curve cryptography (ECC)	Mentioned in (Bos et al., 2014) but Solution Not addressed
		Absence of governance in Bitcoin address production	Authorized clients addresses form trustworthy parties (Ateniese et al., 2014),
		Private key protection	(1) BlueWallet, machine for Bitcoin hardware token (Bamert et al., 2014; Aujla et al., 2018) (2) Two-factor verification by using private key between wallet as well as another machine (Mann et al., 2015)
Wasted Resources	Speed of Bitcoin mining	Nil	(1) Combined usage of CPUs and GPUs for non-custom hardware-based mining (Kumari et al., 2018; Anish Dev, 2014) (2) Modified ASIC processor for more energy-friendly Bitcoin mining (Barkatullah and Hanke, 2015)

(*continued on next page*)

Table 7 (*continued*)

Challenges	Types	Subdivisions	State of Art
	Computation race game between bitcoin miners	Nil	(1) Computation power-free Proof-of-work scheme (Paul et al., 2014; He et al., 2017) (2) Economic model for miners (Wang et al., 2015)
Latency	Nil	Nil	Solution Not addressed
Throughput	Nil	Nil	Solution Not addressed

(i) ***Usability:*** Though Bitcoin API is present to develop services, it is very challenging for the users to utilize its capabilities. Therefore, there is a requirement for developing a better and user-friendly API to exploit the potentials of Blockchain. They may be similar to the REST or RESTful API design (Representational State Transfer)

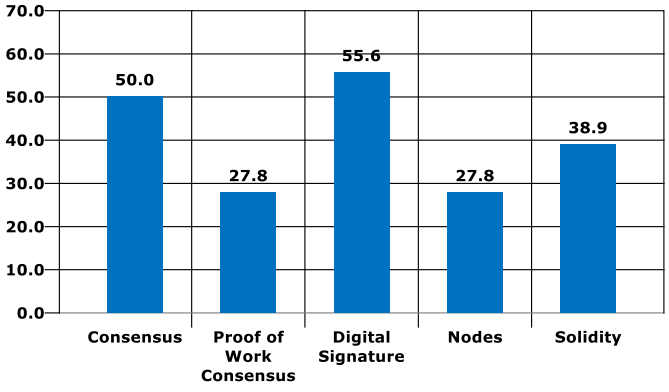

Fig. 20. Factors hampering the performance of blockchain.

5.5. Attacks on blockchain systems

Till now, we have focussed on the foundation of the two main cryptocurrencies- Bitcoin and Ethereum. In this section, we first talk about the factors which hamper the working of blockchain and thereafter, we will focus on the attacks on bitcoin, followed by the vulnerabilities and attacks on Ethereum.

There exist few key limitations of blockchain which hamper its performance. Fig. 20 clearly displays the impact of current challenges of blockchains on smart contract in public as well as private networks. It further elaborates on the fact that few problems which influence only the public networks. On the other hand, several challenges influence both private and public blockchains. There exists not a single challenge which have impact on only private blockchains. The problem of unsustainable consensus methodology displayed in Proof of Work has no effect on private blockchains networks, as majority of the times the problems have consensus methodology on the basis of voting for authenticated transactions (Neto and Hirata, 2013). Since, the authority of authenticating the member lies with the permissioned blockchain, the trustworthy third party's problems related to requirement is solved as most of the nodes involved are trustworthy and known.

5.5.1. Attacks on bitcoin

In the previous section, we have looked at the blockchain's architecture, working and understood the core concept and working of the cryptocurrencies. In this section, we will particularly focus on the attacks

Table 8
Examples of attacks on Bitcoin.

Attack	Explanation	Victims	Negative Impacts	Forecasted Defensive Measures
Double spending	Identical bitcoins are used for more than one transactions Conflicting transactions are sent one after the other in the bitcoin network	Trader	Forks of blockchain are generated Legitimate clients are denied service Products of the merchants are lost	Install monitors in the bitcoin network. Send alert message of the attack to all hosts. Clients near the merchant must inform him about the attack immediately. The primary incoming connection of the merchant must be dismissed
Finney Attack	Attackers secretly mine a blockchain fork (say B) and when they receive the product they purchased they send B over the network	Merchants	Forks of blockchain are generated Legitimate clients are denied service Products of the merchants are lost	Before sending the asset to the host, the merchant should wait for large number of authentications
Brute force attack	Attackers secretly mine a blockchain fork (say B).	Merchants	Large Forks of blockchain are generated Legitimate clients are denied service Products of the merchants are lost	Install monitors in the bitcoin network Send alert message of the attack to all hosts Clients near the merchant must inform him about the attack immediately. The primary incoming connection of the merchant must be dismissed
One confirmation attack or vector 76 attack	Create deposit transaction T_d followed by a new fork(F) and then a withdrawal transaction T_w. IfT_d is rejected, attack is successful.	E-commerce dealing with digital currency	Forks of blockchain are generated Legitimate clients are denied service Huge amount of bitcoin is lost	Before sending the asset to the host, the merchant should wait for large number of authentications
Goldfinger	One miner has >50% computing resources	E-commerce dealing with digital currency and hosts	Denial of service, legitimate users avoid using the network, makes the consensus protocol fragile	Install monitors in the bitcoin network Clients near the merchant must inform him about the attack immediately. TwinsCoin, PieceWork
Selfish mining	Forks in blockchain are generated and longest block chain is considered, rest discarded	Legitimate miners	Facilitate Goldfish attack, because of forking we have race conditions, legitimate miners unnecessarily waste their resources	Various methods can be used -ZeroBlock, Timestamp, DÉCOR + protocol
Block withholding	Partial Proof of work submitted. Two types –	Legitimate miners	Drop the capital of the network, depletion of resources of peers	Network consist of legitimate miners, cease the network if the capital is less than a threshold
FAW attack	Sabotage and Lie in wait enhances on the negative impacts of attacks like selfish mining and block withholding	Legitimate miners	Drop the capital of the network, depletion of resources of peers	None

which occur in the Bitcoin network. But, first we will study about the double spending concept because of which the cause of many attacks in the bitcoin network. Table 8 shows the various examples of attack on Bitcoin.

5.5.1.1. Double spending concept. A user in the Bitcoin network accomplishes a double spending only if he can concurrently expend same bitcoin collection for two distinct transactions (Donnelly, 2016). Example, a malicious user(U_m) generates some transaction(T_{Um_M}) at t time with the help of a collection of bitcoins which has merchant's address(M) for purchasing product from the merchant. U_m broadcasts T_{Um_M} in the bitcoin network. At t_0 instant, U_m generates as well as broadcasts some other transaction T_{Um_Um} with the help of same collection of bitcoins (i.e., B) which has receiver's address as U_m or address of an entity which is works under the user U_m. In this situation, double spending attack is successful, if U_m is able to deceive M to admit T_{Um_M} (i.e., M delivers the product that is purchased to U_m), however M is unable to redeem.

5.5.1.2. Precaution in blockchain. In Bitcoin network, a group of miners validate and execute all transactions as well as they guarantee that for the subsequent transactions only unspent coins which were stated in the preceding transaction results are utilized as input. This protocol is inflicted at run-time to provide protection against the probable double spending. In blockchain network, in order to store the transactions methodically, PoW consensus mechanism and decentralized time stamps are used. For instance, as soon as some miner gets T_{Um_M} and T_{Um_Um} transactions, it can recognize that both transactions (T_{Um_M} and T_{Um_Um}) are using same bitcoins. Therefore, it will authenticate only one of the transactions and discard the other one.

5.5.1.3. Double spending in blockchain. Although in blockchain, ordering of transactions, decentralized time-stamp (Hertig, 2017), PoW mechanism, and other consensus methodologies (Litecoin Project; Wood), is performed, double spending attack is still plausible in Bitcoin. However, there are few necessities which should be accomplished for performing successful double spending: (i) portion of the miners in bitcoin network validate the transaction T_{Um_M} and the merchant (M) receives the validations from miners, and therefore dispatches the product to the malicious client (U_m), (ii) simultaneously, other portion of the miners in the bitcoin network validates transaction T_{Um_Um}, which results in forks in blockchain infrastructure, (iii) the merchant receives the validation of transaction T_{Um_Um} after accomplishing transaction T_{Um_M}, and therefore losses its product (iv) a major part of the miners work on the chain which includes T_{Um_Um} as a legal transaction. In case the above-mentioned steps are have occurred in order then the malicious user will be able to achieve a double spend successfully. Following are some of the variants of double spending attack:

5.5.1.4. Finney attack (Wong and Kar, 2016b). In this attack, malicious users (U_m) privately mines a block(B_p) that includes transaction T_{Um_Um}, and subsequently generates a transaction T_{Um_M} with the help of the same set of bitcoins for the merchant (M). B_p is not notified to the bitcoin network, until transaction T_{Um_M} is admitted by M. M admits T_{Um_M} only if it gets validations from miners that T_{Um_M} is legal and incorporated in the blockchain. Only when U_m receives the purchased product from merchant M, the malicious user broadcasts B_p in the network. This results in creation of a blockchain fork (F′) of same length to the prevailing fork (F). In case the subsequent mined block extends F′ fork in place of F, then all miners in network will have to mine on F′(according to bitcoin protocol). When F' becomes lengthiest blockchain, all miners ignore F, hence thus first block in F which has the transaction T_{Um_M} becomes illegal. As a result, making transaction T_{Um_M} invalid and M will lose its product. Transaction T_{Um_Um} will be executed, and the malicious client will receive its coins. In Finney attack, malicious user double spends only if one-confirmation vendors are present. Fig. 21 shows this attack in detail.

5.5.1.5. Precaution. For circumventing Finney attack, merchants must anticipate for many validations before dispatching the product to purchaser. This anticipation for several validations will result in making double spend more difficult, however the plausibility for double spending will still be prevalent.

5.5.1.6. Brute-force attack (pearson). This is an improvement on Finney attack. In this, a resourceful adversary has governance on some nodes(N) in the bitcoin network, and these n nodes make communal effort to mine block privately with an intension to double spend. An adversary

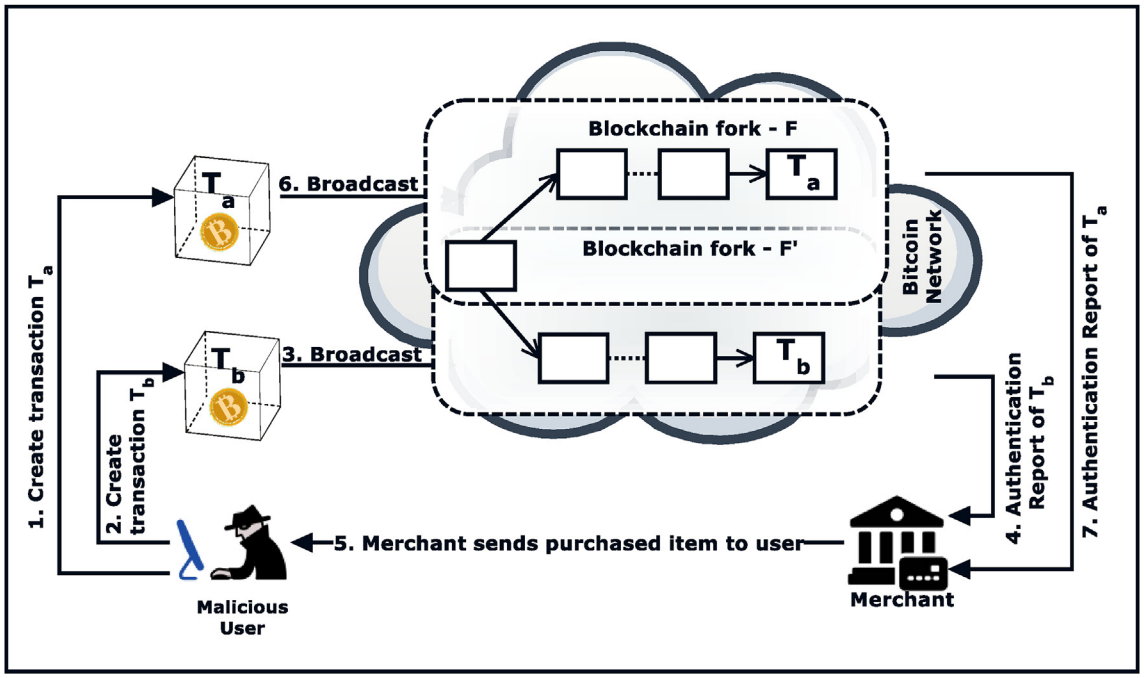

Fig. 21. Finney attack on a Bitcoin network.

incorporates a double spending transaction in some block, simultaneously working on the expansion of private chain (i.e., F′). Assuming a merchant anticipates for 'x' validations before admitting a transaction, and it will deliver the product after it receives 'x' validations. Later, the adversary may mine the 'x' blocks privately and broadcast these blocks in bitcoin network. Since, this will result in longer F′ as compared to F, the fork F′ will be expanded by all miners in the bitcoin network resulting in successful double spend.

5.5.1.7. Vector 76 attack (What Is Dash?). This is another type of attack that makes use of privately mined block for performing double spending attack in Bitcoin Exchange (BE) Networks. A BE is a digital market in which merchants can purchase, exchange or sell bitcoins for some assets. In this attack, a malicious user (U_m) contains a previously mined block which contains a transaction implementing some deposit. The malicious user (U_m) anticipates subsequent block broadcast and sends the previously mined block and newly mined block to the BE or to its neighbouring peers. It expects that some of the miners will mine on the blockchain which contains previously mined block (F') as prime chain. U_m quickly transmits another transaction which requests for withdrawal from the trade of same set of bitcoins which was submitted by the malicious user in its preceding transaction. Now, if the other fork (F) which do not include the transaction which the adversary utilized to credit bitcoins lasts, the credit will be cancelled, however by now U_m has already accomplished the withdrawal. Therefore, the exchange results in loss of bitcoins.

5.5.1.8. Balance attack (Natoli and Gramoli, 2016). In this type of attack, procrastination of network communications among many subdivisions of miners who have balanced mining power occurs. There exists a trade-off between communication latency in bitcoin network and hash-power of adversary which is required to double spend with higher chances in the network of Ethereum (Vasek et al., 2015).

5.5.1.9. Goldfinger attack (Kroll et al., 2013). When computation resources for mining block increases, there is an increase in the possibility of the accomplishment of a double spending which leads to Goldfinger attack (Bos et al., 2014). In this attack, majority of computation resources in the network (more than 50%) are under the influence of only one miner or mining pool. If any action is introduced (transaction rejection/inclusion), this attack can abolish the steadiness of the entire network. This instability in the bitcoin network leads to strengthening the adversary's place when legitimate miners begin to quit the network. This attack is also called as > 50% attack. Table 9 shows the various attacks on the Bitcoin System.

5.6. Attacks on ethereum

In the previous section, we discussed about the vulnerabilities and attacks on the bitcoin network and the concepts involved for the same. In this section, we will focus on the vulnerabilities in the Ethereum cryptocurrency and also discuss the various attacks on the Ethereum network. Before beginning with the attacks, let us first focus on the nomenclatures and vulnerabilities involved in the smart contract, which is an important part of Ethereum. Table 10 shows some of the vulnerabilities and the corresponding attacks on the Ethereum network.

5.6.1. Call to the unknown

Few of the primitives utilized for calling procedures and transferring the ether in solidity might get the poor consequence of referencing callee/recipient fallback function. The fallback function is a unique procedure that can be coded randomly without a function name as well as without arguments. This procedure also runs in case an empty signature

Table 9
Some other attacks on the Bitcoin system.

Attack	Explanation	Victims	Negative Impacts	Forecasted Defensive Measures
Bribery attacks	Malicious users bribe the mining nodes in order to mine for them	The mining nodes as well as the traders	Increases the possibility of withholding the block or a double spending attack	Increment the incentive for honest mining nodes, alerting the mining nodes of the disadvantages of bribery
Refund attacks	Malicious user uses the refund protocols of payment	Traders as well as users	Loss of assets by traders, loss of reputation of honest mining nodes.	Evidence which can be publicly authenticated
Feather and Punitive forking	Malicious miners blacklist transactions of specific address	Users	Freeze the bitcoins of user for forever	No solution yet
Transaction malleability	Malicious user does not validate the transaction and modifies the transaction-id	Centres where Bitcoin exchange take place	Exchange losses assets because of the increment in double credit or double debit	Many metrics for authenticating the transaction
Wallet theft	The malicious user steals or damages the user's private Key	Business or clients	Loss of bitcoin assets in wallet	Secret sharing which is protected by password, two-factor security which has threshold signature, TrustZone-backed Bitcoin wallet, hardware wallets
Time jacking	The malicious user speeds up the clock of many mining nodes.	Mining nodes	A miner is separated and all its resources are wasted and has an impact on mining.	Put constraints on the range of tolerance, time sampling or NTP should be performed on values that are received from the peers
DDoS	Performed to exhaust the resources available in the network	Mining nodes, businesses Bitcoin network, and clients	The facilities of the honest miners are denied, the mining nodes are separated or driven away	signature-based authentication, the Proof-of-Activity protocol
Sybil	The malicious user is responsible for creating many virtual Identities	Clients, mining nodes, Bitcoin network,	It enables time jacking, the privacy of user is threatened, double spend and DDoS	A protocol known as Xim may be used in which two parties are mixed
Eclipse or netsplit	Adversary monopolizes all incoming and outgoing connections of victim	Mining nodes, clients	The network and blockchain's view is not consistent, the double spending concept is enabled with multiple authentications	Whitelists should be utilized, inactivate incoming connection
Tampering	Procrastinate the broadcast of the blocks as well as the transactions to nodes	Mining nodes, clients	Increases the attacks due to DoS, mining advantage is incorrectly mounted, probability if double spending attack	Enhancement of the management system of block requests
Routing attacks	Few nodes are segregated from the network of the Bitcoin, procrastination of the block propagation occur	Mining nodes, clients	DoS attack, mounts probability of the double spend without authentication, mounting fork rate, the mining power of pools is wasted	Entity connections diversity is mounted, supervise the time required for round-trip, utilize the gateways in varying ASes
Deanonymization	The addresses of the clients are attached with the Bitcoin wallet	Users	Privacy breaching of the clients	CoinShuffle, CoinJoin,

Table 10
Few of the vulnerabilities in ethereum and the corresponding attack.

Stage	Vulnerability	Attack
Blockchain	Time Constraint	GovernMental
	Unpredictable State	GovernMental, Dynamic libraries
EVM	Immutable bugs	Rubixi, GovernMental
	Stack size limit	GovernMental
Solidity	Exception disorders	King of the Ether Throne, GovernMental
	Keeping secrets	Multi-player games
	Gasless send	King of the Ether Throne
	Reentrancy	The DAO attack
	Call to the unknown	The DAO attack

is sent on the contract: this situation may occur in case ether is sent to contract.

5.6.2. Gasless send

While utilizing the send procedure for transferring ether to some contract, one may come across an "Out_Of_Gas" exception. The developers cannot anticipate such situation because running the program is not related to transmission of the ether.

5.6.3. Exception disorder

The various circumstances in which exception may occur like, out-of-gas-exception, call stack reaching its threshold, execution of throw command, etc. The safety of the contracts is affected by the variability in management of the exceptions.

5.6.4. Immutable bugs

After the publication of the contract, it becomes immutable. Therefore, the clients may have confidence that in case the contract is executing its respective operation, then its conduct during runtime will be as anticipated because the consensus methodology guarantees it. The disadvantage is that, there exists no straightforward way to redesign it if a contract incorporates a bug. So, in implementing it, designers must predict methods to change or abort a contract even though the coherence of this with Ethereum's protocols can be debated.

5.6.5. Reentrancy

The atomic, as well as the sequential nature of the transaction, might lead developers to assume that non-recursive procedure cannot be re-entered before its cancellation when it is called. This is not always true, though, because the process of fallback can permit an intruder to re-enter the procedure caller. This can lead to disastrous behaviors and probably to loops of invocations that ultimately use all the gas.

5.6.6. Keeping secrets

Contract fields may be public, i.e. accessible directly by all, or private, i.e. not accessible directly by several other clients or contracts. Nevertheless, private declaration of the fields does not ensure its confidentiality. This is because customers need to deliver a proper transaction to the mining nodes to set the value of a field, after which the miners will publish it on blockchain. Because of the public nature of the blockchain, anyone can examine the transaction's contents and thus deduce the field's new value. We will be illustrating some attacks on the Ethereum network, many of these attacks are inspired to real world instances which exploit vulnerabilities as mentioned in the prior section.

5.6.7. The DAO attack (understanding the DAO attack)

DAO was actually a smart contract which implemented crowd-funding platform, which increased approximately $150 M before 18th June 2016, when it was attacked (DAO, 1463). An adversary held approximately $60 M under his influence till blockchain's fork invalidated the transactions engaged in the malicious activity. The shortened version of DAO is shown in Fig. 22.

DAO permits participators to contribute ether for funding the smart

```
Contract DAO {
mapping (address => uint) public creditValue;
function contribute (address destination)
   {creditValue[destination] += msg.value;}
function askCredit (address destination) returns (uint) {
return credit[destination];
}
function debit (uint asset) {
if (credit[msg.sender]>= asset) {
msg.sender.call.value(asset)();
credit[msg.sender]-=asset;
}}}
```

Fig. 22. The shortened version of the DAO attack.

contracts according to will using the function "contribute". The contracts are later allowed to withdraw their assets using the function "debit". Attacks on the above-mentioned smart contract are-

Attack 1: It permits the malicious user to loot the entire ether from DAO. The initial phase of this attack is publicizing contract attack1.

Fig. 23 shows the code for the attack1 on the DAO smart contract. In this, the attacker gives small amount of ether to attack1 as well as calls its fallback which in turn calls withdraw. which transmits ether to attack1. Attack1's fallback is again invoked because of the used function call. This will further call withdraw. It should be noted that withdraw is intermittent before updating credit attribute. Subsequently, DAO again transmits credit to attack1, calls fallback in a loop till exhaustion of gas or stack is overflowed or DAO balance is finished. Using the attack ether can be stolen from DAO.

Attack 2: In this attack an attacker is permitted to loot the entire ether from DAO, however it requires two calls to the fallback function. The initial phase is to publicize Attack2, delivering it with little ether(say 1wei). Subsequently, the attacker summons attack for donating 1wei to itself, and then withdraws it. The responsibility of withdraw function is to examine that user's credit is sufficient, and if this condition is satisfied it transmits ether to attack2.

Fig. 24 displays the code for attack2 on the DAO smart contract. In this, like previous attack, Attack's fallback, followed by "debit", is

```
contract Attack1 {
DAO public dao_attack = DAO(0x354...);
address Sender;
function Attacker1 (){Sender = msg.sender; }
function() { dao_attack.withdraw(dao.askCredit(this)); }
function getJackpot(){Sender.send(this.value); }
}
```

Fig. 23. The code for Attack1 on the DAO smart contract.

```
accomplishAttack = false;
dao.debit (1);
}}
function getJackpot(){
dao.debit (dao.value);
sender.send(this.value);
}}
```

Fig. 24. The code for attack2 on the DAO smart contract.

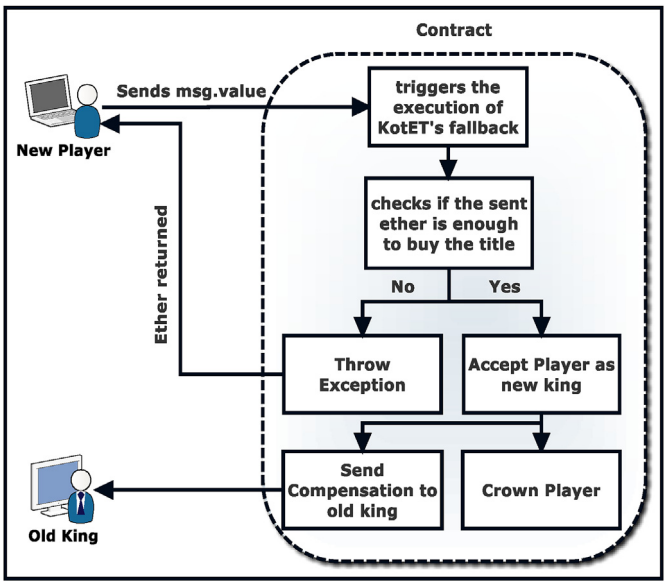

Fig. 25. The procedure of selection of new king.

invoked. Before updation of credit, "debit" is interjected. Thus, 1wei is again sent to Attack2 by DAO invoking fallback again. However, the nested calls will close because nothing is done by fallback. Because of this, credit of attack2 is upgrades two times. For ending the attack getJackpot is called so that all ether from DAO is stolen and transferred to attacker's owner. From these, we can say that the attack 1 is more efficient for greater investment, and attack 2 rewards even for 1wei investment.

5.6.8. King of ether throne ([119], king of the ether thronea; king of the ether throneb)

Is an attack in which contestants compete to acquire "King of the Ether" title. In case any player aspires for becoming king, he has to give the current king some amount of ether and some fees to smart contract. The reward for becoming the king increases at a constant rate. Fig. 25 displays this process clearly.

Here we are discussing about a simple version of the King of the ether game which has similar vulnerabilities and is deployed as mentioned in Fig. 26. In Fig. 26, we can see that on sending msg. value to the smart contract, the contestant triggers KoET's fallback, which checks if the ether is sufficient to purchase king's title. If it is insufficient an exception is raised, and ether is returned otherwise contestant is crowned as new king and *compensation* is given to old king. Contract keeps the value which is the difference of compensation given to old king and msg. value.

```
}
}
function() {
if (msg.value < claimAsset) throw;
uint reimbursement = calculateReimbursement ();
etherKing.send(reimbursement);
etherKing = msg.sender;
claimAsset = calculateNewAsset();
}
/* rest of the procedures*/
}
```

Fig. 26. Code for the king of the ether smart contract.

This value can be collected by KoET's owner by *sweepCommission*.

This contract is dishonest because if send's return code is not examined properly, ether can be stolen. As *send* is exposed with gasless send vulnerability, sending of compensation will be unsuccessful if the address of the former king has contract with costly fallback. In such situation, the contract keeps the compensation because of exception disorder.

5.6.9. Multi-player games

In this, a contract performs "odd & even" game which involves two players, one of which selects a number. First player loses if sum is odd, likewise second player loses if sum is even.

Fig. 27 displays the code which can be implemented for the multiplayer game. In this, the bets of the contestants are stored in "participants". Other contracts cannot access this bet because the field is "private". For joining this game, every contestant should send 1 ether while calling "participate" function. In case different amount is sent, an exception is raised, and the amount is returned. After the second player joins the game, the smart contract runs "Winner" to rewards 1.8 ether to winning player. Rest of the 0.2 ether remains with the contract, which can be accumulated by the possessor using "getIncome".

For performing attack on such contract, an attacker may behave as a second contestant and wait for first contestant's bet. Even though the first contestant's bet cannot be accessed, his bet can be determined by examining the transaction in the blockchain where he got associated with the game. Now, the attacker may become the winner by calling "participate"with appropriate bet.

5.6.10. Rubixi (Bitcointalk, 1400; Etherscan, 2719)

It deploys a Ponzi scheme, which is a deceitful investment system where new users' investments are exploited by the members to acquire money. Moreover, the contract proprietor may accumulate some charges, paid to the contract upon investments. This attack permits the attacker to thieve some amount of contract's ether, taking advantage of "immutable bugs" vulnerability.

In Fig. 28, we can see that the contract name is Rubixi. However, the constructor's name is Rubix by mistake. The contructor should be

```
contract oddsNevens{
struct Participant { address IPadd; uint value;}
Participant[2] private participants;
uint8 total = 0; address sender;

function oddsNevens () {sender = msg.sender;}

function participate(uint value) {
if (msg.value != 1 etherValue) throw;
participants [total] = Participants(msg.sender, value);
total++;
if (total==2) Winner();
}

function Winner() private {
uint n = participants[0].value+ participants[1].value;
participants[n%2].IPadd.send(1800 finney);
delete participants;
total=0;
}
function getIncome() { sender.send(this.asset); }
}
```

Fig. 27. Code for Multi-player game.

```
contract RubixiContract {
address private sender;
function pyramid() { sender = msg.sender; }
function collectIncome() { sender.send(accumulatedInvestment);
}
/* rest of the code*/
```

Fig. 28. Code for the Rubixi Smart contract.

executed only when the contract runs for the first time. However, due to this bug(change in constructor's name), the constructor became public, and could be called by anyone. Rubix function initializes the proprietor's address and the proprietor may use "collectIncome" to acquire his profit.

Because of this bug, users began to call Rubix for becoming the proprietor, and hence gain profit. Fig. 29 highlights the exploitation of governmental attack. Here, Governmental contract collects assets of the contestants in cycles, and the contract rewards only one winner in one cycle. For participating in the scheme, a contestant should pay minimum half of "jackpotValue", whose expense increases after every investment. On invocation of "resetInvestmentValue", winner receives the jackpot and the rest of the ether is transmitted to the contract proprietary. In this, the contract makes an assumption that the contestants are either contracts containing void fallback (to avoid the exception of 'out-of-gas') or the clients. Following are the attacks on such contract.

***Attack 1*:** In this attack, "stack size limit" and "exception disorder" is implemented by the proprietor of the contract. The main aim is to avoid

```
if (msg.value<1 ether) throw;
}
function invest() {
if (msg.value<jackpotValue/2) throw;
finalInvestor = msg.sender;
jackpotValue += msg.value/2;
finalInvestmentTime = block.time;
}
function resetInvestmentValue() {
if (block.time < finalInvestmentTime+
oneMinute)
throw;
finalInvestor.send(jackpotValue);
sender.send(this.balance-1 ether);
finalInvestor = 0;
jackpotValue = 1 ether;
finalInvestmentTime = 0;
}
}
```

Fig. 29. Code for the governmental attack.

paying the winner in order to keep ether with the contract and the proprietor can claim it later. For this, the proprietor attempts to fail the execution of "finalInvestor.send(jackpotValue); ". For this he has to publish the contract Attack1 as mentioned in Fig. 30.

In Fig. 30, we can see that this contract invokes Attack1's "accomplishAttack" function, which will result in recursive calling of "accomplishAttack" function. The stack will now start growing and when the size 1022 is reached, "resetInvestmentValue" function of Governmental is invoked. Now, this is executed at stack size 1023 and thus sending of jackpot to winner will fail because of the call stack limit. In Governmental, the return code of sending functions are not examined and the code resets the contract status and begins another cycle. The contract's cost grows in each cycle since the legitimate winner is not given the amount. For accumulating ether, the proprietor has to wait for next cycle to end successfully.

***Attack 2*:** In this attack, miner (who in reality is the adversary) impersonates as a contestant. Since he is a miner, he has the authority of not selecting the blocks which contain transactions to Governmental. He may select only the block containing his transaction for becoming the last contestant in a cycle. Moreover, the adversary may change the ordering of the transactions, in order to keep his transaction first. If he plays first, he may choose appropriate ether amount for investment and prevent rest of the contestant from joining this scheme and thus, becoming the last contestant in the cycle. This attack makes use of a vulnerability called "unpredictable state". This is because while publishing the transaction for joining the scheme, the contestants are uncertain if the investment is sufficient for the success of the operation.

***Attack 3*:** In this attack, also miner (who in reality is the adversary) impersonates to be a contestant. Assuming the adversary joins the scheme, in order to become the last contestant in one cycle, which executes for a minute, he can manipulate the block's timestamp. For this he has to set the new block's timestamp minimum 1 min later the present block's timestamp. Thus this attack exploits "time constraints" vulnerability. In case the adversary, publishes a new block which has deferred timestamp, he may end up being the last contestant in a particular cycle and may win the prize.

5.6.11. Dynamic libraries

In this dynamic updation of library of tasks is performed. Thus, in case a bug is removed or a better implementation for these tasks is deployed, the contract may utilize the newer library version. In Fig. 31, we can see that the contract's proprietor "Set_Provider" may utilize the function "update_Library" for replacing the address of the library with the address of the new library. Library's address can be fetched by every user using "getLibAdd". Some elementary set tasks are implemented by the "library Set". A library is a specific contract, which for instance contain immutable fields. If a client state an interface as library, *delegatecall* can be used to making straight invocation its functions. The attributes, called"storage", are passed by reference. Assuming that "User" a Set_Provider's client who is legitimate. In Fig. 32, user requests for the library version via "getLibraryVersion" function.

Assuming the proprietary of Set_Provider is a malicious user, he may attack User for thieving his ether. For this, we can see in Fig. 33 that first "AttackingSet", a new library is set by the attacker and subsequently,

```
contract Attack1 {
function accomplishAttack(address targetIP, uint countValue) {
if (0<=countValue && countValue<1023) this.accomplishAttack.gas(msg.gas-
2000)(targetIP, countValue+1);
else GovernMentalAttack(targetIP).reset();
}
}
```

Fig. 30. Code for the attack on the GovernMental Contract.

```
Contract Set_Provider {
address LibAdd;
Address sender;
function Set_Provider() {
sender = msg.sender;
}
}
function update_Library(address argument) {
if (msg.sender==sender)
setLibAdd = argument;
}
function getLibAdd () returns (address) {
return LibAdd;
}
}
library Set {
struct Info { mapping(uint => bool) flag; }
function insertion(Info storage selfInfo, uint asset)
returns (bool) {
selfInfo.flag[asset] = true;
return true;
}
function remove(Info storage selfInfo, uint asset)
returns (bool) {
selfInfo.flags[asset] = false;
return true;
}
function contains(Info storage selfInfo, uint asset)
returns (bool) {
return selfInfo.flag[asset];
}
function versionNumber() returns(uint) { return 1; }
}
```

Fig. 31. Code for the dynamic libraries.

```
library Set { function version() returns (uint); }
contract User {
Set_Provider public suppplier;
function User(address arguments) { supplier =
Set_Provider(address); }
function getLibraryVersion() returns (uint) {
address setAddress = supplier.getLibAdd ();
return Set(setAddress).versionNumber();
}}
```

Fig. 32. Code for legitimate User requesting for library version.

"update_Library" of "Set_Provider" is invoked for directing it to "AttackingSet".

"AttackingSet" sends ether to the malicious user. As "User" has stated the library as Set interface, straight invocation of the version is deployed as *delegatecall*, and therefore run in the environment of "User". Thus, "this.asset" is user's balance and therefore his entire ether is transferred to the malicious user. After this, accurate version is returned by the

```
library AttackingSet {
address constant attackerAdd = 0x42;
function versionNumber () returns(uint)
{
attackerAdd.send(this.asset);
return 1;
}}
```

Fig. 33. Code for the attack on the dynamic library.

function. The function selfdestruct may be used for creating a malicious library. It deactivates the executing contract and sends its entire ether to a specific address. Thus, "attackerAdd.send(this.value); " can be replaced with "selfdestruct(attackerAdd); "In this, "unpredictable state" vulnerability is being exploited, as User is unknown to the library version that executes on using "Set_Provider". The primary problem with libraries is the existence of portions in code that can be updated only publishing the contract, thus permitting the malicious user to change these parts according to them.

6. Security enhancements in blockchain

Till now, we have discussed about the working of the blockchain, which forms the base of the bitcoin and ethereum network which are popular network of cryptocurrency. Then, we studied about various vulnerabilities and attacks performed on these networks. Now, this section summarizes the security improvements in the field of blockchain, that may be utilized for the deployment of blockchain and hence improve the bitcoin and Ethereum networks. Table 11 shows few of the security enhancements in the area of blockchain.

6.1. SmartPool

A mining pool having computational power greater than 40% exists which threatens the distributed nature of blockchain. This in turn makes blockchain vulnerable to various threats and attacks. Fig. 34 illustrates the working of a novel approach called smart contract which was proposed by Loi et al. (Luu et al., 2017). Various Ethereum users like parity (P. technologies, 2017b), geth (E. community, 2017b) etc. send transactions to the SmartPool. These transactions comprise of information related to the mining job. Subsequently, a miner performs hash computation on the basis of jobs and thereafter sends back the accomplished shares to the smartpool user. After the quantity of the accomplished shares becomes equal to some specific amount, the shares will be dedicated to the smartpool contract. This smartpool contract is implemented in Ethereum and will also authenticate the shares as well as provide incentives to the user. The process is shown in Fig. 34.

When we assess traditional peer-to-peer pool with respect to the SmartPool system, we come across following advantages:

1) Distributed: Blockchain has smart contract implemented in it and SmartPool is deployed in terms of this smart contract. At first the miners involved associate with Ethereum for mining through the user and the mining pool may depend on the consensus methodology in Ethereum for execution. Thus, it guarantees distributed behaviour of the pool miners. In addition to this, pool operator is not required as Ethereum supervises the state of the mining pool.

2) Effectiveness: The miners involved may transmit to smartpool contract the accomplished shares in batches. Moreover, the miners are required to transmit only a portion of the shares which have to be

Table 11
Security Enhancement in the field of blockchain

Technique	Primary Focus	Key Elements used	Problem on which work is done	Deployment	Contributions	Shortcomings or challenges	Advantage
SmartPool (Loi et al. Luu et al., 2017**)**	Smart Contract	Introduce new data structure called augmented Merkle tree	In case one pool operator governs more than 50% of the mining power of the network, a 51% attack starts threatening the Nakamotoconsensus protocol's security	It is implemented on core network using a community Project which is crowd-funded	(1) A solution for distributed pool mining is introduced (2) Distributed pool miningprotocol is implemented as smart contract. (3) Ithas scalability and efficiency.	(1) A pool may contain many shares, therefore the contract many receive many messages. (2) In case feesfor submitingone share outweighs the incentive received, StrawmanPool cangive negative income to the mining nodes. (3) Any malicious user may witness transaction of other miners. (4) No guarantee is provided by the smart contract executing in the Bitcoin mining pool for Bitcoin payment	Decentralized, Efficiency, Secure
Quantitative Framework (Arthur et al. Gervais et al., 2016c**)**	PoW	(1) a blockchain instance (2) a blockchain security model.	The security is the affected in case the performance of the blockchains (based on PoW) is improved	(1) Consensus Layer (2) Network Layer	(1) The greater the block incentive in blockchain, greater is the resilience against double spend. (2) gives information to the traders for determining the number of authentications to avoid double spend attack. (3) Ethereum requires minimum of 37 validations for matching the security of the Bitcoinwith six block validations against a malicious user having 30% mining power. In the same way, Litecoin needs 28 and Dogecoin needs 47 block validations. (4) Examine the effect of alteration of size of the block.		Performance, Security
Oyente (Loi et al. Luu et al., 2016b**)**	Smart Contract	This requires two inputs-current global state of Ethereum and the bytecode of the contract.	Miners in Ethereum must follow few rules while taking part in the network, however the exists high probability of alterations of risk of not witnessing novice implementation		(1) It records many security bug classespresent in smart contracts of Ethereum. (2) It gives some solutions for the recorded bugs. (3) It gives Oyente, an virtual execution tool that helps smart contact of Ethereum for identifying bugs. (4) implements Oyente on Ethereum smart contracts as well as assures the attacks which is possible on real network of Ethereum.	5411 contracts have mishandledexceptions	(1)Bugs are Removed (2)open source
Hawk (Ahmed et al. Kosba et al., 2016**)**	Smart Contract		The completeseries of activities which occur in smart contract in Ethereum are broadcasted in the		(1) **On-chain privacy** -secures the privacy of the parties involved in		Privacy-preserving smart contracts

(*continued on next page*)

Table 11 (*continued*)

Technique	Primary Focus	Key Elements used	Problem on which work is done	Deployment	Contributions	Shortcomings or challenges	Advantage
			network and stored in blockchain, thus they can be read publicly.		the contractfrom the public (2) **Contractual security**- safeguards the parties involved in the same contractual agreement from one another		
Town Crier Zhang et al. Zhang et al. (2016)	Smart contract	The TC Contract, the Enclave and the Relay	Internet connection with a trustworthy website can be established using HTTPS and appears to have given a solution, however the Ethereum's smart contract cannot access the network. Moreover, the digital signature in not present for the out-of-band authentication in HTTP Protocol.	Blockchain, Town Crier Serevr	(1) gives a peer-to-peer deployment of TC (2) examineTC's security in the structure of Universal Composability (3) gives a hybridized TCB that spans the blockchain as well as an SGX enclave. (4) investigates the three TC applications which displays TC's ability for supporting wide variety of services.		Smart Contracts interact with external data sources

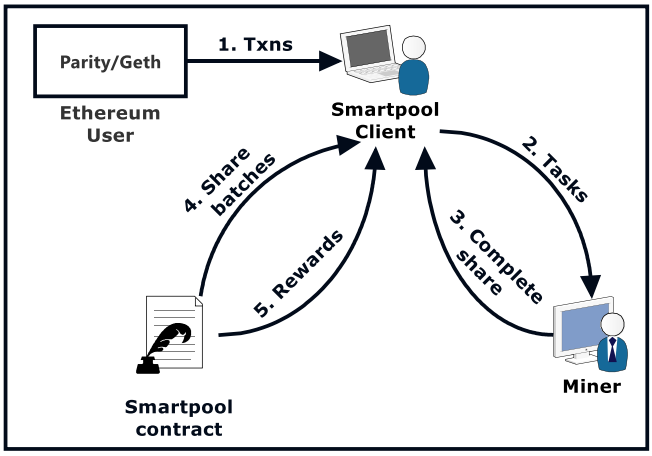

Fig. 34. Smartpool's Execution process.

authenticated. Therefore, the effectiveness and efficiency of SmartPool is higher when compared to peer-to-peer pool.

3) Security: SmartPool makes use of a new data structure, that has the ability to avoid the adversary from submitting the shares in various batches. Moreover, the authentication methodology of SmartPool assures that legitimate miners will get anticipated incentives even if dishonest miners are present in the pool.

6.2. Quantitative framework

It is used for analysis of performance, execution and security aspects of blockchains which is based on PoW consensus methodology. As represented in Fig. 35, this framework consists of two main constituents- 1) blockchain stimulator 2) security model. Simulator has consensus protocol's attribute and network's attribute as input and performs execution of blockchain. By analysing the simulator, theblockchain's performance statistics can be gained. This also includes throughput, network delays, block propagation times, stale block rate, block sizes, etc.

6.3. Oyente

Loi et al. (Luu et al., 2016b) propositioned Oyente for detecting faults in the smart contract of Ethereum. Oyente(open source (Luu et al., 2016c)) makes use of simulated execution for analysing smart contracts' bytecode. As Ethereum incorporates smart contracts' bytecode in blockchain, detection of faults in the implemented smart contracts may be done by Oyente.

Fig. 36represents structural design and execution procedure of Oyente. There exist two inputs-1) bytecode of smart contract 2) Global state of Ethereum. Initially, smart contract's bytecode is used by CFGC (Control Flow Graph Constructor) for constructing CFG (Control Flow Graph) for smart contract. This CFG, along with Ethereum state, is leveraged by EXPLORER for execution. This will improve the CFG since few jump targets are variable and are calculated during this execution. Subsequently, the output is supplied to CORE ANALYSIS which leverages analysis algorithms for detecting four vulnerabilities, which is

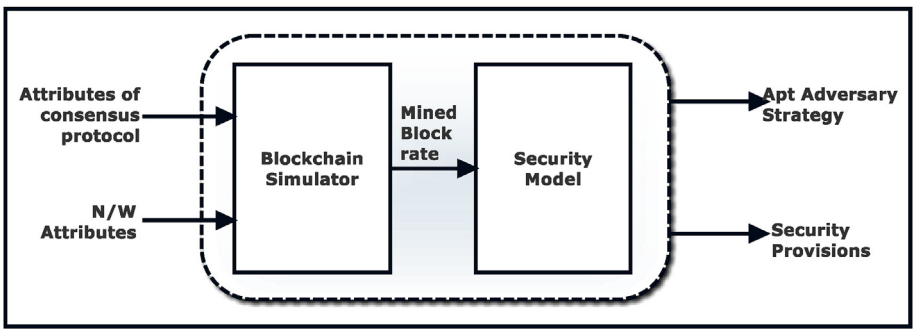

Fig. 35. Overview of quantitative framework.

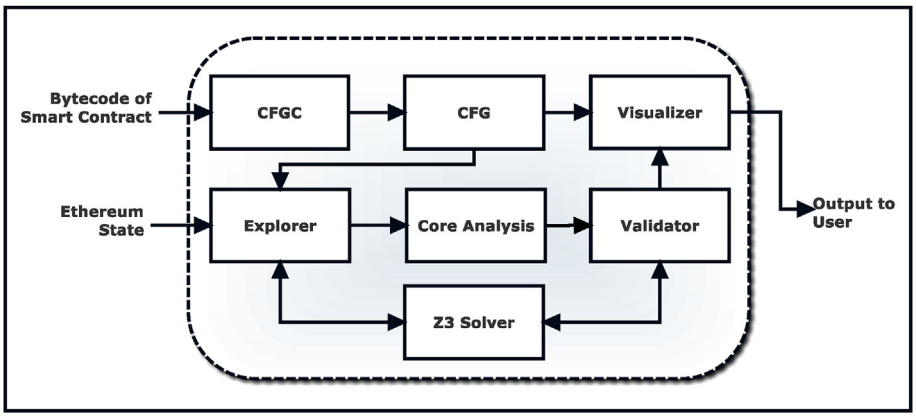

Fig. 36. Structural design and execution of Oyente.

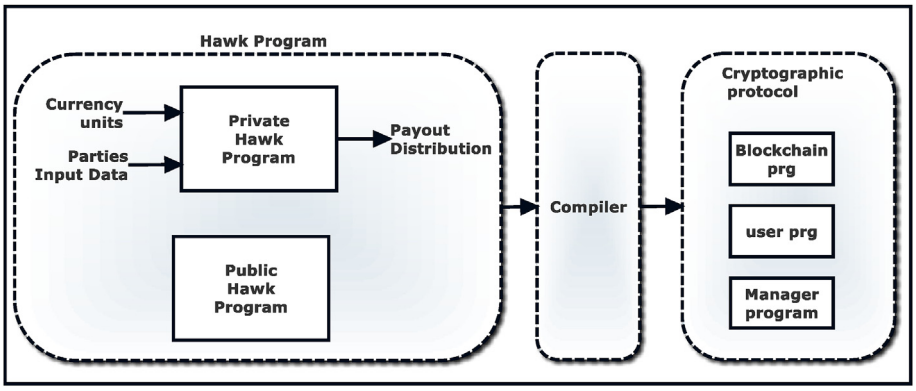

Fig. 37. Structural design and Execution of Hawk.

authenticated by VALIDATOR. Authenticated vulnerability as well as CFG will become the output for the VISUALIZER, which may further be used by the clients for debugging as well as for program analysis.

6.4. Hawk

Ahmed et al. (Kosba et al., 2016) made a proposition Hawk, which is a framework to develop private smart contracts. With the help of Hawk, the developers may develop smart contracts which are privacy-preserving, and there is no necessity to leverage code encryption technique. Moreover, the information of economic transaction is not explicitly recorded in the blockchain.

The smart contract in Hawk consist of private and public sections. The private section includes private data as well as codes related to economic function and public section includes that information which do not contain private data. The process is shown in Fig. 37.

A smart contract in Hawk can be compiled in three phases:

(1) Code to be executed in entities' machines.
(2) Code to be executed by smart contract's users.
(3) Code to be executed by a trusted entity in Hawk called manager, who can read the private data of the smart contract but not reveal it.

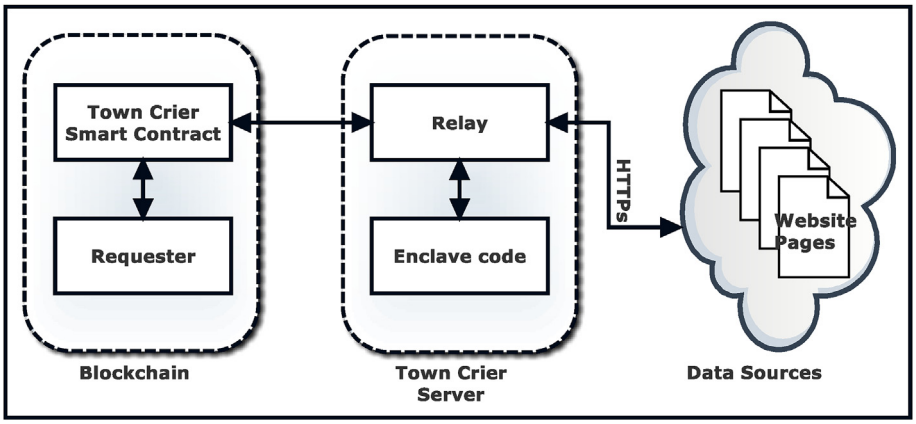

Fig. 38. Overview and working of Town Crier.

Table 12
Artificial intelligence techniques based solutions for DDoS attack detection.

State-of-art	Approaches used	Summary
Berral (García et al., 2008)	Machine learning, Naive Bayes	- The paper extends a framework proposed by zhang in 2006 to detect/prevent DDoS flood attacks based on machine learning - nodes in an intermediate network share information about their local traffic observations, improving their global traffic perspective
Kiruthika (Devi et al., 2014)	Machine learning, SVM	- The Spoofed traffic detection module incorporates hop count inspection algorithm (HCF) to check the authenticity of incoming packet - OMS (online monitoring system) provides DDoS impact measurements in real time by monitoring the degradation in host and network performance metrics - HCF is coupled with SVM - accuracy ~ 98.99%
Zhao (Zhao et al., 2015)	neural network, Hadoop	- develop a DDoS detection system with learning capability to adapt to new types of DDoS attacks - ability to store and analyze a huge unstructured dataset collected from network logs - a list of training samples is developed to train the neural network
Ndibwile (Ndibwile et al., 2015)	ML	- makes use of real web server, Bait server, and Decoy web servers to distinguish DDoS traffic from normal traffic. - A Custom Intrusion Prevention System (IPS) is used which uses rules generated by a random tree machine learning algorithm using supervised learning
Robinson (Robinson and Thomas, 2015)	ML	- Aim - to capture DDoS attacks using ML Algorithms - Provides Evaluation/Ranking of some supervised ML algorithms with the aim of reducing type I/type II errors, increasing precision and recall while maintaining detection accuracy - Performance evaluation is done using Multi Criteria Decision Aid software called Visual PROMETHEE
Heish (Hsieh and Chan, 2016)	Neural Network, Hadoop	- Proposes DDoS detection method based on Neural Networks, implemented in the Apache Spark cluster - Use of 2000 DARPA LLDOS 1.0 dataset to train and perform experiments to the detection system in a real network environment - Avg detection rate- over 94%
Meitei (Meitei et al., 2016)	ANN	- Detection using Decision Tree (TREE), Multi Layer Perceptron (MLP), Naïve Bayes (NB) and Support Vector Machine (SVM) to classify the DNS traffics into normal and abnormal - Attribute selection algorithms such as Information Gain, Gain Ratio and Chi Square are used to achieve optimal feature subset - 99.3% accuracy
Fouladi (Fouladi et al., 2016)	ML, Naive Bayes	- Uses a Naive Bayes classifier with two frequency based methods of discrete Fourier transform and discrete wavelet transform in order to separate between attack and normal traffics
Ramadhan (Ramadhan et al., 2016)	Artificial immune system	- designs a TCP flood DDoS detection system which uses Artificial Immune System(AIS) - Uses dendritic cell algorithm (DCA) - The DCA is also designed to solve the problem in network intrusion detection
PerakoviÉ (Peraković et al., 2016)	ANN	- artificial neural network(ANN) architecture to detect DDoS attack. - Traffic are classified as four kinds – class-DNS DDoS attack traffic, chargen DDoS attack traffic, UDP DDoS attack traffic and normal traffic.
Xuan (Yuan et al., 2017)	Deep Learning, CNN, RNN	- Deep Learning based detection algorithm - DeepDefense - A recurrent deep neural network to learn patterns from sequences of network traffic and trace network attack activities

Apart from providing privacy from public, hawk also provides confidentiality among several Hawk contracts. In case, Hawk protocol is aborted by manager, it is economically fined, and the clients get reimbursement.

6.5. *Town crier*

Frequently, Smart contract requires communication with off-chain information source. Zhang et al. (2016) made a proposition of Town Crier, that is a validated information feed system for information communication procedure. Smart contract cannot acquire information via HTTPS as they are not in direct contact with the network. Town Crier is a connection between external information source, which is HTTPS-enabled and smart contract. The structural design of town crier is displayed in Fig. 38. Town Crier smart contract is in fact front end of Town Crier structure, which behaves as an API in between contract of the clients and Town Crier server. Main code of Town Crier is executing in the Intel SGX enclave.

The primary focus of the Town Crier server is to acquire information request from contracts of the client as well as acquire information from the targeted websites. Subsequently, the Town Crier server returns blockchain messages, which contain digital signature as datagram to client's contract. Town crier can secure the process which is demanding information. The primary modules of the Town Crier are run on distributed Ethereum, enclave which is SGX-enabled, and websites which are HTTPS-enabled. Moreover, the enclave deactivates internet connectivity for maximizing security. The relay module has been constructed as a internet communication hub which is utilized by the information source websites, environment of SGX enclave and smart contracts. Thus, it acquires isolation between the execution of Town Crier's main code and internet communication. The function of Town Crier is unaffected by modification of internet communication packets or some attack on Relay module. Town crier is inaugurated online for public service (Zhang et al., 2017). Table 12, summarize the various artificial intelligence techniques based solutions proposed by the researchers for DDoS attack detection.

6.6. *Future trends*

According to the above-mentioned methodical survey on blockchain and its challenges, we concluded with the following findings of future areas in which efforts can be put in research directions.

- Currently, PoW is one of the most extensively utilized consensus methodology which is being implemented in blockchain. However, a lot of computing resources are being wasted in PoW. For finding solution of this issue, a hybrid consensus methodology of Proof-of-Work and Proof-of-Stake mechanisms is being developed by Ethereum. Performing research and coming forward with more effective consensus methodology may lead to significant contribution for the advancement of the technology of blockchain.
- With the increase in the amount of decentralized applications which are rich in features, there is also an increase in risk of privacy leakage. A decentralized application and communication process that exists between the decentralized application and the network, both face the issue of privacy leakage. Some solutions to these issues are:

application hardening, code complication, execution trusted computing (e.g., Intel SGX), etc.

- A lot of data is produced by the blockchain but not all data that is recorded in the blockchain is authentic. Example, SUICIDE and SELFDESTRUCT may be used by smart contract for erasing its code, however the smart contract's address is not deleted. Moreover, many contracts either do not contain code or the code is exactly same as in Ethereum, moreover some contracts may have not been executed even once after it was deployed. An effective data recognition and cleaning methodology is required for improving the efficiency of execution of blockchain.

7. Concluding remarks

Recently, blockchain is extremely valued and recommended due to its peer-to-peer nature and decentralized structure. Nevertheless, numerousstudiesrelated to blockchain were only restricted to Bitcoin. However, blockchain could be realized in numerous areas, which fall outside the boundary of Bitcoin. Many times, blockchain has revealed its capabilities for converting conventional IT sector area with its several features: decentralization, persistency, privacy and auditability. In this survey article, the authors have tried to provide a systematic and comprehensive survey of blockchain initially explicitly highlighting the structure of network of blockchain and the lifecycle of transactions involved in a cryptocurrency network. The authors also includednumerous technologies involved in blockchain like consensus methodologies, forks and also facilitates with a detailed discussion on smart contract which acts as a treaty among disbelieving members and implemented by the blockchain's consensus methodologies. A detailed taxonomy of blockchain (comprising public blockchain, private blockchain, etc.) clearly highlighting their features and real-world applications is also presented along with their detailed comparison-based analysis.

The authors also explain numerous key platforms of blockchain (like bitcoin, litecoin, ethereum, hyperledger, etc.) along with their comparison-based analysis based on some useful parameters (like consensus algorithms involved, blockchain type, etc.). Existing security issues and challenges of blockchain systems is also investigated in this article along with the key factors hampering the performance of existing blockchain systems. Several emerging vulnerabilities of bitcoin and ethereum (for e.g., double spending attack, finney attack, vector 76 attack, etc.) is also discussed in this article. Finally, the authors summarizes the security improvements in the field of blockchain, that may be utilized for the deployment of blockchain and hence, improves the bitcoin and ethereum networks. The authors would like to carry forward their research on the smart contract languages as a part of future work, since several real-world applications is somewhat infeasible to implement precisely using such emerging platforms of smart contract languages.

Authors statement

Authors declare that the manuscript is submitted only to this journal and is not being considered for submission simultaneously at other venues.

Declaration of competing interest

Authors declare no comfily of interest of any kind during the submission.

Acknowledgement

The survey presented in this article was supported by Research Initiation Grant (RIG) and financially supported by Birla Institute of Technology and Science, Pilani, India. The authors would also like to thank all their group members and related co-authors who were actively involved in providing the valuable feedback and comments related to this article.

References

And Adinolfi, J., 2016. 2016's Best-Performing Commodity Is ... Bitcoin?. http://www.marketwatch.com/story/and-2016s-best-performing-commodit y-is -bitcoin-2016-12-22.

Aggarwal, Shubhani, Chaudhary, Rajat, Singh Aujla, Gagangeet, Kumar, Neeraj, Raymond Choo, Kim-Kwang, Zomaya, Albert Y., 2019. Blockchain for smart communities: applications, challenges and opportunities. J. Netw. Comput. Appl.

Anderson, L., Holz, R., Ponomarev, A., Rimba, P., Weber, I., 2016. New kids on the block: an analysis of modern blockchains. CoRR abs/1606.06530.

Androulaki, E., Karame, G., 2014. Hiding transaction amounts and balances in bitcoin. of Lecture Notes in Computer Science. In: Holz, T., Ioannidis, S. (Eds.), Trust and Trustworthy Computing, vol. 8564. Springer International Publishing, pp. 161–178. https://doi.org/10.1007/978-3-319- 08593-7_11. Available from:

Andrychowicz, M., Dziembowski, S., Malinowski, D., Mazurek, 2015. On the malleability of bitcoin transactions. of Lecture Notes in Computer Science. In: Brenner, M., Christin, N., Johnson, B., Rohloff, K. (Eds.), Financial Cryptography and Data Security, vol. 8976. Springer Berlin Heidelberg, pp. 1–18. https://doi.org/10.1007/978-3-662-48051-9_1. Available from:

Anish Dev, J., 2014. Bitcoin mining acceleration and performance quantification. In: Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on, pp. 1–6.

Ateniese, G., Faonio, A., Magri, B., de Medeiros, B., 2014. Certified bitcoins. Available from:. In: Boureanu, I., Owesarski, P., Vaudenay, S. (Eds.), Applied Cryptography and Network Security. Vol. 8479 of Lecture Notes in Computer Science. Springer International Publishing, pp. 80–96. https://doi.org/10.1007/978-3-319-07536-5_6.

Aujla, G.S., Chaudhary, R., Kumar, N., Das, A.K., Rodrigues, J.J.P.C., 2018. SecSVA: secure storage, verification, and auditing of big data in the cloud environment. IEEE Commun. Mag. 56 (1), 78–85.

Azaria, A., Ekblaw, A., Vieira, T., Lippman, A., 2016. Medrec: using blockchain for medical data access and permission management. In: International Conference on Open and Big Data. OBD, pp. 25–30.

Bamert, T., Decker, C., Wattenhofer, R., Welten, S., 2014. BlueWallet: the secure bitcoin wallet. of Lecture Notes in Computer Science. In: Mauw, S., Jensen, C. (Eds.), Security and Trust Management, vol. 8743. Springer International Publishing, pp. 65–80. https://doi.org/10.1007/978-3- 319-11851-2_5. Available from:

Bansal, Gaurang, Amit, Dua, Singh Aujla, Gagangeet, Singh, Maninderpal, Kumar, Neeraj, 2019. SmartChain: a smart and scalable blockchain consortium for smart grid systems. In: 2019 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, pp. 1–6.

Barkatullah, J., Hanke, T., 2015. Goldstrike 1: CoinTerra's first-generation cryptocurrency mining processor for bitcoin. Micro, IEEE 35 (2), 68–76. https://doi.org/10.1109/MM.2015.13.

Beikverdi, A., Song, J., 2015. Trend of centralization in Bitcoin's distributed network. In: Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS International Conference on, pp. 1–6.

Bitcointalk: Hi!My name is Rubixi, https://bitcointalk.org/index.php?topic=1400536.60.

Blockchain Blockchain security in cloud computing: use Cases,Challenges, and solutions .

blockchaininfo, 2017. Confirmed Transactions Per Day. https://blockchain.info/charts/n-transactions?timespan=all/#.

Bodkhe, Umesh, Bhattacharya, Pronaya, Tanwar, Sudeep, Tyagi, Sudhanshu, Kumar, Neeraj, Obaidat, M.S., 2019. Blohost: blockchain enabled smart tourism and hospitality management. In: 2019 International Conference on Computer, Information and Telecommunication Systems (CITS). IEEE, pp. 1–5.

Bos, J.W., Halderman, J.A., Heninger, N., Moore, J., Naehrig, M., Wustrow, E., 2014. Elliptic curve cryptography in practice. March 3–7, 2014, Revised Selected Papers. In: Financial Cryptography and Data Security—18th International Conference, FC 2014, Christ Church, Barbados, pp. 157–175. https://doi.org/10.1007/978-3-662-45472-5_11. Available from:

Bose, Arnab, Singh Aujla, Gagangeet, Singh, Maninderpal, Kumar, Neeraj, Cao, Haotong, 2019. Blockchain as a service for software defined networks: a denial of service attack perspective. In: 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech). IEEE, pp. 901–906.

Brown, A., 2013. 10 Things You Need to Know about Ripple. CoinDesk. May 17. https://www.coindesk.com/10-things-you-need-to-know-about-ripple/.

Buterin, V., 2013. Ethereum: a Next Generation Smart Contract and Decentralized Application Platform. https://github.com/ethereum/wiki/wiki/White-Paper.

Buterin, V., 2013. Ethereum: a Next Generation Smart Contract and Decentralized Application Platform. https://github.com/ethereum/wiki/wiki/White-Paper.

Buterin, V., 2014. A Next-Generation Smart Contract and Decentralized Application Platform. *white paper*.

Bylica, P., Gleń, Ł., Janiuk, P., Skrzypczak, A., Zawłocki, A., 2015. A Probabilistic Nanopayment Scheme for Golem. http://golemproject.net/doc/GolemNanopayments.pdf.

Cachin, C., 2016. Architecture of the Hyperledger blockchain fabric. In: Workshop on Distributed Cryptocurrencies and Consensus Ledgers. July.

Chaudhary, Rajat, Jindal, Anish, Singh Aujla, Gagangeet, Aggarwal, Shubhani, Kumar, Neeraj, Raymond Choo, Kim-Kwang, 2019. BEST: blockchain-based secure

energy trading in SDN-enabled intelligent transportation system. Comput. Secur. 85, 288–299.
Chepurnoy, A., Larangeira, M., Ojiganov, A., 2016. A Prunable Blockchain Consensus Protocol Based on Non-interactive Proofs of Past States Retrievability arXiv preprint arXiv:1603.07926.
The DAO raises more than $117 million in world's largest crowdfunding to date. https://bitcoinmagazine.com/articles/the-dao-raises-more-than- million-in-world-s-largest-crowdfunding-to-date-1463422191.
Decker, C., Wattenhofer, R., 2013. Information propagation in the Bitcoin network. In: Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on, pp. 1–10.
Decker, C., Guthrie, J., Seidel, J., Wattenhofer, R., 2015. Making bitcoin exchanges transparent. of Lecture Notes in Computer Science. In: Pernul, G., Ryan P, Y.A., Weippl, E. (Eds.), Computer Security—ESORICS 2015, vol. 9327. Springer International Publishing, pp. 561–576. https://doi.org/10.1007/978-3-319-24177-7_28. Available from:
Decker, C., Seidel, J., Wattenhofer, R., 2016. Bitcoin meets strong consistency. In: Proceedings of the 17th International Conference on Distributed Computing and Networking (ICDCN). ACM, Singapore, Singapore, p. 13.
Delmolino, K., Arnett, M., Kosba, A.M.A., Shi, E., 2016. Step by Step towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab.
Desjardins, J., 2016. It's Official: Bitcoin Was the Top Performing Currency of 2015. http://money.visualcapitalist.com/its-official-bitcoin-wasthe-top-performing-currency-of-2015/.
Devi, BS Kiruthika, Preetha, G., Selvaram, G., Mercy Shalinie, S., 2014. An impact analysis: real time DDoS attack detection and mitigation using machine learning. In: 2014 International Conference on Recent Trends in Information Technology. IEEE, pp. 1–7.
Di Battista, G., Di Donato, V., Patrignani, M., Pizzonia, M., Roselli, V., Tamassia, R., 2015. Bitconeview: visualization of flows in the bitcoin transaction graph. In: Visualization for Cyber Security (VizSec), 2015 IEEE Symposium on, pp. 1–8.
Donnelly, J., 2016. What Is the 'Halving'? A Primer to Bitcoin's Big Mining Change. CoinDesk. June 12. https://www.coindesk.com/making-sense-bitcoinshalving/.
Dorri, A., Kanhere, S.S., Jurdak, R., Gauravaram, P., 2017. Blockchain for iot security and privacy: the case study of a smart home. In: IEEE Percom Workshop on Security Privacy and Trust in the Internet of Thing.
Duffield, E. and Diaz, D., "Dash: A Privacy-Centric Crypto-Currency." https://github.com/dashpay/dash/wiki/Whitepaper .
E. community, 2017. Kovan - Stable Ethereum Public Testnet. https://github.com/kovan-testnet/proposal.
E. community, 2017. Official Go Implementation of the Ethereum Protocol. https://github.com/ethereum/go-ethereum.
Ekblaw, A., Azaria, A., Halamka, J.D., Lippman, A., 2016. A Case Study for Blockchain in Healthcare: Medrec Prototype for Electronic Health Records and Medicalresearch Data. https://www.media.mit.edu/publications/medrecwhitepaper/.
Ethereum reddit page, https://www.reddit.com/r/ethereum.
Ethereum Wiki: Contract security techniques and tips, https://github.com/ethereum/wiki/wiki/Safety.
Etherscan: Rubixi code, https://etherscan.io/address/0xe82719202e5965Cf5D9B6673B7503a3b92DE20be.
Eyal, I., Sirer, E., 2014. Majority is not enough: bitcoin mining is vulnerable. In: Financial Cryptography and Data Security, pp. 436–454.
Eyal, I., Sirer, E., 2014. Majority is not enough: bitcoin mining is vulnerable. In: Financial Cryptography and Data Security, pp. 436–454.
Eyal, I., Sirer, E., 2014. Majority is not enough: bitcoin mining is vulnerable. of Lecture Notes in Computer Science. In: Christin, N., Safavi-Naini, R. (Eds.), Financial Cryptography and Data Security, vol. 8437. Springer Berlin Heidelberg, pp. 436–454. https://doi.org/10.1007/978-3-662- 45472-5_28. Available from:
Feld, S., Schnfeld, M., Werner, M., 2014. Analyzing the deployment of bitcoin's {P2P} network under an ASlevel perspective. Proc. Comput. Sci. 32, 1121–1126. https://doi.org/10.1016/j. procs.2014.05.542.
financial-institutions https://www.statista.com/statistics/648044/blockchain-usage-by-financial-institutions/.
Fouladi, Ramin Fadaei, Eren Kayatas, Cemil, Anarim, Emin, 2016. Frequency based DDoS attack detection approach using naive Bayes classification. In: 2016 39th International Conference on Telecommunications and Signal Processing (TSP). IEEE, pp. 104–107.
Garay, J.A., Kiayias, A., Leonardos, N., 2015. The Bitcoin backbone protocol: analysis and applications. In: EUROCRYPT, pp. 281–310.
Garay, J., Kiayias, A., Leonardos, N., 2015. The bitcoin backbone protocol: analysis and applications. of Lecture Notes in Computer Science. In: Oswald, E., Fischlin, M. (Eds.), Advances in Cryptology—EUROCRYPT 2015, vol. 9057. Springer Berlin Heidelberg, pp. 281–310. https://doi.org/10.1007/978-3-662-46803-6_10. Available from:
García, Berral, Lluís, Josep, Mastrokalo, Nicolas Poggi, Alonso López, Javier, Gavaldà Mestre, Ricard, Torres Viñals, Jordi, Parashar, Manish, 2008. Adaptive distributed mechanism againts flooding network attacks based on machine learning. In: The First ACM Workshop on AISec. ACM Press, NY, pp. 43–49.
Gervais, A., Karame, G.O., Wüst, K., Glykantzis, V., Ritzdorf, H., Capkun, S., 2016. On the security and performance of proof of work blockchains. In: The 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 3–16.
Gervais, Arthur, Karame, Ghassan O., Wust, Karl, Glykantzis, Vasileios, Hubert, Ritzdorf, Capkun, Srdjan, 2016. On the security and performance of proof of work blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16. ACM, New York, NY, USA, pp. 3–16.
Gervais, A., Karame, G.O., Wüst, K., Glykantzis, V., Ritzdorf, H., Capkun, S., 2016. On the security and performance of proof of work blockchains. In: The ACM SIGSAC Conference on Computer and Communications Security, pp. 3–16.
Ghosh, M., Richardson, M., Ford, B., Jansen, R., 2014. A torpath to torcoin, proof-ofbandwidth altcoins for compensating relays. https://www.smithandcrown.com/open-research/a-torpath-to-torcoin-proof-of-bandwidth-altcoins-for-compensating-relays/.
Greenspan, G., 2016. Introducing MultiChain Streams. MultiChain. September 15. http://www.multichain.com/blog/2016/09/introducing-multichain-streams/.
Gupta, Rajesh, Tanwar, Sudeep, Tyagi, Sudhanshu, Kumar, Neeraj, Mohammad, S., 2019. Obaidat, and balqies sadoun. "Habits: blockchain-based telesurgery framework for healthcare 4.0. In: 2019 International Conference on Computer, Information and Telecommunication Systems (CITS). IEEE, pp. 1–5.
He, Debiao, Kumar, Neeraj, Zeadally, Sherali, Vinel, Alexey, Laurence, T., Yang, 2017. Efficient and privacy-preserving data aggregation scheme for smart grid against internal adversaries. IEEE Trans. Smart Grid 8 (5), 2411–2419.
Herrera-Joancomart, J., 2015. Research and challenges on bitcoin anonymity. of Lecture Notes in Computer Science. In: Garcia-Alfaro, J., HerreraJoancomart, J., Lupu, E., Posegga, J., Aldini, A., Martinelli, F., et al. (Eds.), Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, vol. 8872. Springer International Publishing, pp. 3–16. https://doi.org/10.1007/978-3-319-17016-9_1. Available from:
Hertig, A., 2017. Litecoin's SegWit Activation: Why it Matters and What's Next. CoinDesk. April 26. https://www.coindesk.com/litecoins-segwit-activationwhy-it-matters-and-whats-next/.
Hsieh, Chang-Jung, Chan, Ting-Yuan, 2016. Detection DDoS attacks based on neural-network using Apache Spark. In: 2016 International Conference on Applied System Innovation (ICASI). IEEE, pp. 1–4.
Huckle, S., Bhattacharya, R., White, M., Beloff, N., 2016. Internet of things, blockchain and shared economy applications. Proc. Comput. Sci. 98, 461–466.
Hurich, P., 2016. The virtual is real: an argument for characterizing bitcoins as private property. In: Banking & Finance Law Review, vol. 31. Carswell Publishing, p. 573.
NRI, 2015. Survey on blockchain technologies and related services. Tech. Rep. [Online]. Available: http://www.meti.go.jp/english/press/2016/pdf/0531 01f.pdf.
Intel, 2017. Proof of Elapsed Time (Poet). http://intelledger.github.io/.
Introduction "Introduction to Ripple for Bitcoiners," last modified December 10, 2013. https://wiki.ripple.com/Introduction_to_Ripple_for_Bitcoiners .
Jindal, Anish, Singh Aujla, Gagangeet, Kumar, Neeraj, 2019. SURVIVOR: a blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment. Comput. Network. 153, 36–48.
Karl, 2016. Security of Blockchain Technologies. Ph.D. thesis. Swiss Federal Institute of Technology.
Karl, 2016. Ethereum Eclipse Attacks. http://e-collection.library.ethz.ch/view/eth:49728.
Kaur, Harmanjeet, Kumar, Neeraj, Batra, Shalini, 2018. An efficient multi-party scheme for privacy preserving collaborative filtering for healthcare recommender system. Future Generat. Comput. Syst. 86, 297–307.
King of the Ether Throne: Post mortem investigation, https://www.kingoftheether.com/postmortem.html.
King of the Ether Throne: source code, https://github.com/kieranelby/KingOfTheEtherThrone/blob/v0.4.0/contracts/KingOfTheEtherThrone.sol.
Kosba, A., Miller, A., Shi, E., Wen, Z., Papamanthou, C., Hawk, 2016. The blockchain model of cryptography and privacy-preserving smart contracts. In: IEEE Symposium on Security and Privacy, pp. 839–858.
Koshy, P., Koshy, D., McDaniel, P., 2014. An analysis of anonymity in bitcoin using P2P network traffic. of Lecture Notes in Computer Science. In: Christin, N., Safavi-Naini, R. (Eds.), Financial Cryptography and Data Security, vol. 8437. Springer Berlin Heidelberg, pp. 469–485. https://doi.org/10.1007/978-3-662-45472-5_30. Available from:
Kraft, D., 2016. Difficulty control for blockchain-based consensus systems. Peer-to-Peer Netw. Appl. 9 (2), 397–413.
Kroll, J.A., Davey, I.C., Felten, E.W., 2013. The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries.
Kumari, Saru, Karuppiah, Marimuthu, Das, Ashok Kumar, Xiong, Li, Wu, Fan, Kumar, Neeraj, 2018. A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers. J. Supercomput. 74 (12), 6428–6453.
Lee Kuo Chuen, D. (Ed.), 2015. Handbook of Digital Currency, first ed. Elsevier [Online]. Available: http://EconPapers.repec.org/RePEc:eee:monogr:9780128021170.
Li, Xinghua, Wang, Yunwei, Vijayakumar, Pandi, He, Debiao, Kumar, Neeraj, Ma, Jianfeng, 2019. Blockchain-based mutual-healing group key distribution scheme in unmanned aerial vehicles ad-hoc network. IEEE Trans. Veh. Technol. 68 (11), 11309–11322.
Lim, I.K., Kim, Y.H., Lee, J.G., Lee, J.P., Nam-Gung, H., Lee, J.K., 2014. The analysis and countermeasures on security breach of bitcoin. of Lecture Notes in Computer Science. In: Murgante, B., Misra, S., Rocha, A.C., Torre, C., Rocha, J., Falco, M., et al. (Eds.), Computational Science and its Applications ICCSA 2014, vol. 8582. Springer International Publishing, pp. 720–732. https://doi.org/10.1007/978-3-319-09147-1_52. Available from:
Hyperledger Business Blockchain Technologies," The Linux Foundation. https://www.hyperledger.org/projects.
Litecoin Project. https://litecoin.org/.
Luu, L., Teutsch, J., Kulkarni, R., Saxena, P., 2015. Demystifying incentives in the consensus computer. In: ACM CCS, pp. 706–719.
Luu, L., Teutsch, J., Kulkarni, R., Saxena, P., 2015. Demystifying incentives in the consensus computer. In: ACM CCS, pp. 706–719.

Luu, L., Teutsch, J., Kulkarni, R., Saxena, P., 2015. Demystifying incentives in the consensus computer. In: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security. CCS'15. ACM, New York, NY, USA, pp. 706–719. https://doi.org/10.1145/2810103.2813659. Available from:

Luu, L., Chu, D.H., Olickel, H., Saxena, P., Hobor, A., 2016. Making smart contracts smarter. In: ACM CCS. http://eprint.iacr.org/2016/633.

Luu, L., Chu, D.-H., Olickel, H., Saxena, P., Hobor, A., 2016. Making smart contracts smarter. In: The 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 254–269.

Luu, L., Chu, D., Olickel, H., Saxena, P., Hobor, A., 2016. Oyente: an Analysis Tool for Smart Contracts. https://www.comp.nus.edu.sg/~loiluu/oyente.html.

Luu, L., Velner, Y., Teutsch, J., Saxena, P., 2017. Smart pool: Practical decentralized pooled mining. In: USENIX Security Symposium.

Mann, C., Loebenberger, D., 2015. Two-factor Authentication for the bitcoin protocol. Available from:. In: Foresti, S. (Ed.), Security and Trust Management. Vol. 9331 of Lecture Notes in Computer Science. Springer International Publishing, pp. 155–171. https://doi.org/10.1007/978-3-319-24858-5_10.

Meiklejohn, S., Orlandi, C., 2015. Privacy-enhancing overlays in bitcoin. of Lecture Notes in Computer Science. In: Brenner, M., Christin, N., Johnson, B., Rohloff, K. (Eds.), Financial Cryptography and Data Security, vol. 8976. Springer Berlin Heidelberg, pp. 127–141. https://doi.org/10.1007/978- 3-662-48051-9_10. Available from:

Meitei, Irom Lalit, Singh, Khundrakpam Johnson, De, Tanmay, 2016. Detection of DDoS DNS amplification attack using classification algorithm. In: Proceedings of the International Conference on Informatics and Analytics, pp. 1–6.

Miglani, Arzoo, Kumar, Neeraj, Chamola, Vinay, Zeadally, Sherali, 2020. Blockchain for internet of energy management: review, solutions, and challenges. Comput. Commun.

Mistry, Ishan, Tanwar, Sudeep, Tyagi, Sudhanshu, Kumar, Neeraj, 2020. Blockchain for 5G-enabled IoT for industrial automation: a systematic review, solutions, and challenges. Mech. Syst. Signal Process. 135, 106382.

Moser, M., Bohme, R., Breuker, D., 2013. An inquiry into money laundering tools in the Bitcoin ecosystem. In: eCrime Researchers Summit. eCRS, pp. 1–14, 2013.

Nakamoto, S., 2008. Bitcoin: A Peer-To-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf.

Nakamoto, S., 2008. Bitcoin: A Peer-To-Peer Electronic Cash System [Online]. Available: https://bitcoin.org/bitcoin.pdf.

Nakamoto, S., 2008. Bitcoin: A Peer-To-Peer Electronic Cash System [Online]. Available: https://bitcoin.org/bitcoin.pdf.

Nakamoto, Satoshi, 2009. Bitcoin: A Peer-To-Peer Electronic Cash System.

Narayanan, A., Bonneau, J., Felten, E., Miller, A., Goldfede, S., 2016. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press.

Natoli, C., Gramoli, V., 2016. The balance attack against proof-ofwork blockchains: the R3 testbed as an example. abs/1612 CoRR, 09426.

Ndibwile, Jema David, Govardhan, A., Okada, Kazuya, Kadobayashi, Youki, 2015. Web Server protection against application layer DDoS attacks using machine learning and traffic authentication. In: 2015 IEEE 39th Annual Computer Software and Applications Conference, vol. 3. IEEE, pp. 261–267.

Neto, J.B., Hirata, C.M., 2013. Lifecycle for Management of E-Contracts Based on Web Service, vol. 7.

P. technologies, 2017. Proof of Authority Chains. https://github.com/paritytech/parity/wiki/Proof-of-Authority-Chains.

P. technologies, 2017. Parity. https://parity.io/.

Paul, G., Sarkar, P., Mukherjee, S., 2014. Towards a more democratic mining in bitcoins. of Lecture Notes in Computer Science. In: Prakash, A., Shyamasundar, R. (Eds.), Information Systems Security, vol. 8880. Springer International Publishing, pp. 185–203. https://doi.org/10.1007/978-3- 319-13841-1_11. Available from:

Pearson, J., 2016. The Ethereum Hard Fork Spawned a Shaky Rebellion. Motherboard. July 27. https://motherboard.vice.com/en_us/article/theethereum-hard-fork-spawned-a-shaky-rebellion-ethereum-classic-etc-eth.

Peraković, Dragan, Periša, Marko, Cvitić, Ivan, Husnjak, Siniša, 2016. Artificial neuron network implementation in detection and classification of DDoS traffic. In: 2016 24th Telecommunications Forum (TELFOR). IEEE, pp. 1–4.

Ramadhan, Gilang, Kurniawan, Yusuf, Kim, Chang-Soo, 2016. Design of TCP SYN Flood DDoS attack detection using artificial immune systems. In: 2016 6th International Conference on System Engineering and Technology (ICSET). IEEE, pp. 72–76.

Rathore, Shailendra, Sharma, Pradip Kumar, Loia, Vincenzo, Jeong, Young-Sik, Hyuk Park, Jong, 2017. Social network security: issues, challenges, threats, and solutions. Inf. Sci. 421, 43–69.

Rathore, Shailendra, Loia, Vincenzo, Hyuk Park, Jong, 2018. SpamSpotter: an efficient spammer detection framework based on intelligent decision support system on Facebook. Appl. Soft Comput. 67, 920–932.

Robinson, RR Rejimol, Thomas, Ciza, 2015. Ranking of machine learning algorithms based on the performance in classifying DDoS attacks. In: 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS). IEEE, pp. 185–190.

Ruffing, T., Moreno-Sanchez, P., Kate, A., 2014. CoinShuffle: practical decentralized coin mixing for bitcoin. of Lecture Notes in Computer Science. In: Kutyowski, M., Vaidya, J. (Eds.), Computer Security—ESORICS 2014, vol. 8713. Springer International Publishing, pp. 345–364. https://doi.org/10.1007/978-3-319-11212-1_20. Available from:

Saxena, A., Misra, J., Dhar, A., 2014. Increasing anonymity in bitcoin. of Lecture Notes in Computer Science. In: Bhme, R., Brenner, M., Moore, T., Smith, M. (Eds.), Financial Cryptography and Data Security, vol. 8438. Springer Berlin Heidelberg, pp. 122–139. https://doi.org/10.1007/978-3-662- 44774-1_9. Available from:

Sharma, Pradip Kumar, Kumar, Neeraj, Hyuk Park, Jong, 2018. Blockchain-based distributed framework for automotive industry in a smart city. IEEE Trans. Indust. Inform. 15 (7), 4197–4205.

Shojafar, M., Cordeschi, N., Baccarelli, E., 2016. Energy-efficient adaptive resource management for real-time vehicular cloud services. IEEE Trans. Cloud Comput. (99) https://doi.org/10.1109/TCC.2016.2551747, 1–1.

Solidity: security considerations, http://solidity.readthedocs.io/en/develop/index.html.

Sompolinsky, Y., Zohar, A., 2013. Accelerating bitcoin's transaction processing. fast money grows on trees, not chains. IACR Cryptology ePrint Archive 2013 (881).

Sompolinsky, Y., Zohar, A., 2015. Secure high-rate transaction processing in bitcoin. In: Financial Cryptography and Data Security, pp. 507–527.

Spagnuolo, M., Maggi, F., Zanero, S., 2014. BitIodine: extracting intelligence from the bitcoin network. of Lecture Notes in Computer Science. In: Christin, N., Safavi-Naini, R. (Eds.), Financial Cryptography and Data Security, vol. 8437. Springer Berlin Heidelberg, pp. 457–468. https://doi.org/10.1007/978-3-662-45472-5_29. Available from:

State of Blockchain Q1 2016: Blockchain Funding Overtakes Bitcoin, 2016 [Online]. Available: http://www.coindesk.com/state-of-blockchain-q1-2016/.

Todd, P., 2014. Bitcoin Improvement Proposal (BIP) 65, "OP_CHECKLOCKTIMEVERIFY,". October 1. https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki.

Understanding the DAO attack, http://www.coindesk.com/understanding-dao- hack-journalists/.

Valenta, L., Rowan, B., 2015. Blindcoin: blinded, accountable mixes for bitcoin. of Lecture Notes in Computer Science. In: Brenner, M., Christin, N., Johnson, B., Rohloff, K. (Eds.), Financial Cryptography and Data Security, vol. 8976. Springer Berlin Heidelberg, pp. 112–126. https://doi.org/10.1007/978-3-662-48051-9_9. Available from:

Vandervort, D., 2014. Challenges and opportunities associated with a bitcoin-based transaction rating system. of Lecture Notes in Computer Science. In: Bhme, R., Brenner, M., Moore, T., Smith, M. (Eds.), Financial Cryptography and Data Security, vol. 8438. Springer Berlin Heidelberg, pp. 33–42. https://doi.org/10.1007/978-3-662-44774-1_3. Available from:

Vasek, M., Thornton, M., Moore, T., 2014. Empirical analysis of denial-of-service attacks in the bitcoin ecosystem. of Lecture Notes in Computer Science. In: Bhme, R., Brenner, M., Moore, T., Smith, M. (Eds.), Financial Cryptography and Data Security, vol. 8438. Springer Berlin Heidelberg, pp. 57–71. https://doi.org/10.1007/978-3-662-44774-1_5. Available from:

Vasek, M., Moore, T., 2015. Theres No free lunch, even using bitcoin: tracking the popularity and profits of virtual currency scams. of Lecture Notes in Computer Science. In: Bhme, R., Okamoto, T. (Eds.), Financial Cryptography and Data Security, vol. 8975. Springer Berlin Heidelberg, pp. 44–61. https://doi.org/10.1007/978-3-662-47854-7_4. Available from:

https://www.statista.com/statistics/647374/worldwide-blockchain-wallet-users/.

Wang, L., Liu, Y., 2015. Exploring miner evolution in bitcoin network. of Lecture Notes in Computer Science. In: Mirkovic, J., Liu, Y. (Eds.), Passive and Active Measurement, vol. 8995. Springer International Publishing, pp. 290–302. https://doi.org/10.1007/978-3-319-15509-8_22. Available from:

What Is Dash", WeUseCoins. https://www.weusecoins.com/what-is-dash/.

Wong, J., Kar, I., 2016. Everything You Need to Know about the Ethereum 'hard Fork,. Quartz Media. July 18. https://qz.com/730004/everything-you-needto-know-about-the-ethereum-hard-fork/.

Wong, J., Kar, I., 2016. Everything You Need to Know about the Ethereum 'hard Fork. Quartz Media. July 18. https://qz.com/730004/everything-you-needto-know-about-the-ethereum-hard-fork/.

Wood, G., "Ethereum: A Secure Decentralised Generalised Transaction Ledger." https://bravenewcoin.com/assets/Whitepapers/Ethereum-A-Secure-Decentralised-Generalised-Transaction-Ledger-Yellow-Paper.pdf.

Yuan, Xiaoyong, Li, Chuanhuang, Li, Xiaolin, 2017. DeepDefense: identifying DDoS attack via deep learning. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP). IEEE, pp. 1–8.

Yue, X., Wang, H., Jin, D., Li, M., Jiang, W., 2016. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. J. Med. Syst. 218.

Zhang, Y., Wen, J., 2016. The IoT electric business model: using blockchain technology for the internet of things. Peer-to-Peer Netw. Appl. 1–12.

Zhang, Zezhong, Qi, Qingqing, Kumar, Neeraj, Chilamkurti, Naveen, Jeong, Hwa-Young, 2015. A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography. Multimed. Tool. Appl. 74 (10), 3477–3488.

Zhang, F., Cecchetti, E., Croman, K., Juels, A., Shi, E., 2016. Town crier: an authenticated data feed for smart contracts. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, pp. 270–282.

Zhang, F., Cecchetti, E., Croman, K., Juels, A., Shi, E., 2017. Town Crier. http://www.town-crier.org/.

Zhao, Teng, Lo, Dan Chia-Tien, Qian, Kai, 2015. A neural-network based DDoS detection system using hadoop and HBase. In: 2015 IEEE 17th International Conference on High Performance Computing and Communications, *2015* IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems. IEEE, pp. 1326–1331.

Zheng, Z., Xie, S., Dai, H.-N., Wang, H., 2016. Blockchain challenges and opportunities: a survey, Internat. J. Web Grid Serv.

Ziegeldorf, J.H., Grossmann, F., Henze, M., Inden, N., Wehrle, K., 2015. CoinParty: secure multi-party mixing of bitcoins. In: Proceedings of the 5th ACM Conference on Data and Application Security and Privacy. CODASPY'15. ACM, New York, NY, USA, pp. 75–86. https://doi.org/10.1145/2699026.2699100. Available from:

Dr. Gupta is currently working as an Assistant Professor in Computer Science and Information Systems Division at Birla Institute of Technology and Science, Pilani, Rajasthan, India. He has done his PhD under the supervision of Dr. B. B. Gupta in Department of Computer Engineering specialization in Web Security at National Institute of Technology Kurukshetra, Haryana, India. Recently, he was working as an Assistant Professor in the Department of Computer Science and Engineering at Jaypee Institute of Information Technology (JIIT), Noida, Sec-128. Prior to this, he has also served his duties as an Assistant Professor in the Department of IT at Model Institute of Engineering and Technology (MIET), Jammu. He has completed M.Tech. in the Department of Computer Science and Engineering Specialization in Information Security from Central University of Rajasthan, Ajmer, India. He has also done his graduation in Bachelor of Engineering (B.E.) in Department of Information Technology from Padmashree Dr. D.Y. Patil Institute of Engineering and Technology Affiliated to Pune University, India. He has also spent two months in the Department of Computer Science and IT, University of Jammu for completing a portion of Post-graduation thesis work. He bagged the 1st Cash Prize in Poster Presentation at National Level in the category of ICT Applications in Techspardha'2015 and 2016 event organized by National Institute of Kurukshetra, Haryana. He has numerous online publications in International Journals and Conferences including IEEE, Elsevier, ACM, Springer, Wiley, Elsevier, IGI-Global, etc. along with several book chapters. He is also serving as reviewer for numerous peer-reviewed Journals and conferences of high repute. He is also a professional member of IEEE and ACM. His research area of interest includes Web Security, Cross-Site Scripting (XSS) attacks, Online Social Network Security, Cloud Security, IoT, and Fog Computing.

Amit Dua is Assistant Professor in Computer Science and Information System Department in BITS Pilani. He did BE (Hons) in computer science and MS in software systems both from BITS Pilani. He has completed his Ph.D. thesis from Thapar University in Ad hoc networks. He has seven years of experience in teaching, industry and research. He has worked as lecturer at NIT Kurukshetra and was part of software team at UPSINVERTER.COM. His areas of interest include ad hoc networks, network security, game theory, fuzzy logic, and smart grid. He is a reviewer of International Journal of Communication Systems, KSII Transactions on Internet and Information Systems and TPC member of several international Conferences. He is a member of IEEE and ACM. Outside the work, he is fond of various outdoor sports activities. He has won prizes for table tennis and lawn tennis at the state level and at annual institutional sports meets.

Prof. Neeraj Kumar (SM1received his Ph.D. in CSE from Shri Mata Vaishno Devi University, Katra (Jammu and Kashmir), India in 2009, and was a postdoctoral research fellow in Coventry University, Coventry, UK. He is working as a Professor in the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology (Deemed to be University), Patiala (Pb.), India. He has published more than 400 technical research papers in top-cited journals such as IEEE TKDE, IEEE TIE, IEEE TDSC, IEEE TITS, IEEE TCE, IEEE TII, IEEE TVT, IEEE ITS, IEEE SG, IEEE Netw., IEEE Comm., IEEE WC, IEEE IoTJ, IEEE SJ, Computer Networks, Information sciences, FGCS, JNCA, JPDC and ComCom. He has guided many research scholars leading to Ph.D. and M.E./M.Tech. His research is supported by funding from UGC, DST, CSIR, and TCS. He is an Associate Technical Editor of IEEE Communication Magazine. He is an Associate Editor of IJCS, Wiley, JNCA, Elsevier, Elsevier Computer Communications, and Security and Communication, Wiley. He has been a guest editor of various International Journals of repute such as - IEEE Access, IEEE Communication Magazine, IEEE Network Magazine, Computer Networks, Elsevier, Future Generation Computer Systems, Elsevier, Journal of Medical Systems. Springer, Computer and Electrical Engineering, Elsevier, Mobile Information Systems, International Journal of Ad hoc and Ubiquitous Computing, Telecommunication Systems, Springer and Journal of Supercomputing, Springer. He has been a workshop chair at IEEE Globecom 2018 and IEEE ICC 2019 and TPC Chair and member for various International conferences. He is senior member of the IEEE. He has more than 7200 citations to his credit with current h-index of 452. He has won the best papers award from IEEE Systems Journal and ICC 2018, Kansas city in 2018. He is visiting research fellow at Coventry University, Newcastle University, UK, King Abdul Aziz University, Saudi Arabia. He is included in the list of highly cited researcher list issued by web of science 2019. https://dblp.org/pers/hd/k/Kumar_0001:Neeraj. https://scholar.google.com/citations?hl=en&user=gL9gR-4AAAAJ