The Journal of Systems & Software 196 (2023) 111555

Contents lists available at ScienceDirect

The Journal of Systems & Software

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

A survey of blockchain consensus safety and security: State-of-the-art, challenges, and future work☆

Qihao Bao, Bixin Li *, Tianyuan Hu, Xueyong Sun

School of Computer Science and Engineering, Southeast University, Nanjing, 211189, China

ARTICLE INFO

Article history:
Received 12 November 2021
Received in revised form 27 September 2022
Accepted 31 October 2022
Available online 4 November 2022

Keywords:
Blockchain consensus
Consensus safety
Consensus security
Challenges

ABSTRACT

Blockchain technology has been widely used in finance, Internet of Things, insurance, and other fields since it was proposed in 2008. As a part of blockchain technology, the consensus protocol has a profound impact on the safety and security of blockchain systems. Therefore, there have been a lot of research on the safety and security of blockchain consensus protocols. However, the research shows us the diversity and complexity which motivate us to do this survey. In this paper, we provide a comprehensive survey focusing on the safety and security of blockchain consensus protocols, where we discuss in detail the classification of blockchain consensus protocols, the potential safety and security problems, safety and security assurance approaches, and validation methods etc. Furthermore, we also identify research challenges and current research gaps, and suggest future work to be further investigated in the end of this paper.

1. Introduction

Since blockchain technology was proposed with Bitcoin in 2008 (Nakamoto, 2008), it is always the hotpoint issues in both academic circles and industrial domain (Sun et al., 2016; Fanning and Centers, 2016; Shrier et al., 2016; Al-Jaroodi and Mohamed, 2019; Sathya and Banik, 2020). Blockchain technology provides technical infrastructure for different levels of blockchain platform and advocates the idea of decentralization applications (DApp), which supports comprehensive industrial applications and provides wide services for different domains (Kim and Laskowski, 2017; Chen et al., 2017a; Zheng et al., 2019).

With the wide application of blockchain technology, both safety and security problems are becoming increasingly serious. For example, the financial security risks are exposed now and then, which has caused more and more safety and security accidents (Slowmist, 2022). According to incomplete statistics of the blockchain security company NoneAge (NoneAge, 2019), the number of blockchain safety and security accidents in 2019 was up to more than 140, resulting in economic losses of more than $5 billion.

During all factors or reasons that will trigger safety and security accidents, consensus protocols play an important role (NoneAge, 2019). Consensus layer is an important part of blockchain technology, and the safety and security of consensus protocols directly affect the normal operation of the whole blockchain system. For instance, the design defects of the blockchain consensus protocols may cause the blockchain system to reach an unsafe state during operation, which can cause major safety accidents. Otherwise, attackers can exploit the vulnerabilities of consensus mechanism to attack consensus protocols, which can cause damage to the system and make huge profits from it.

In view of this, many researchers have studied the safety and security problems of consensus protocols, and achieved some good results. However, since the researchers come from different disciplines, with different perspectives of attention and different technical methods, it is regrettable that the existing research results are not systematic and the research ideas are not clear enough. And there is currently no research on a comprehensive systematic study focusing on the safety and security of blockchain consensus protocols. We hope that this paper fills this gap.

In this paper, we present a comprehensive survey focusing on the safety and security aspects of blockchain consensus protocols and its related concepts, covering 42 papers published between 2012 and 2021. To the best of our knowledge, this is the first large-scale and comprehensive survey on the safety and security of blockchain consensus protocols.

The rest of the paper is organized as follows. We briefly introduce blockchain and give an overview of blockchain consensus in Section 2; we describe the survey method used in our study in Section 3; we present the survey results in Section 4; we discuss the deficiencies of existing research, current research challenges, and compare the related work in Section 5; we propose future research directions in Section 6; and finally, we conclude the paper in Section 7.

☆ Editor: Matthias Galster.

* Corresponding author.
E-mail addresses: qihao_bao@seu.edu.cn (Q. Bao), bx.li@seu.edu.cn (B. Li).

https://doi.org/10.1016/j.jss.2022.111555
0164-1212/

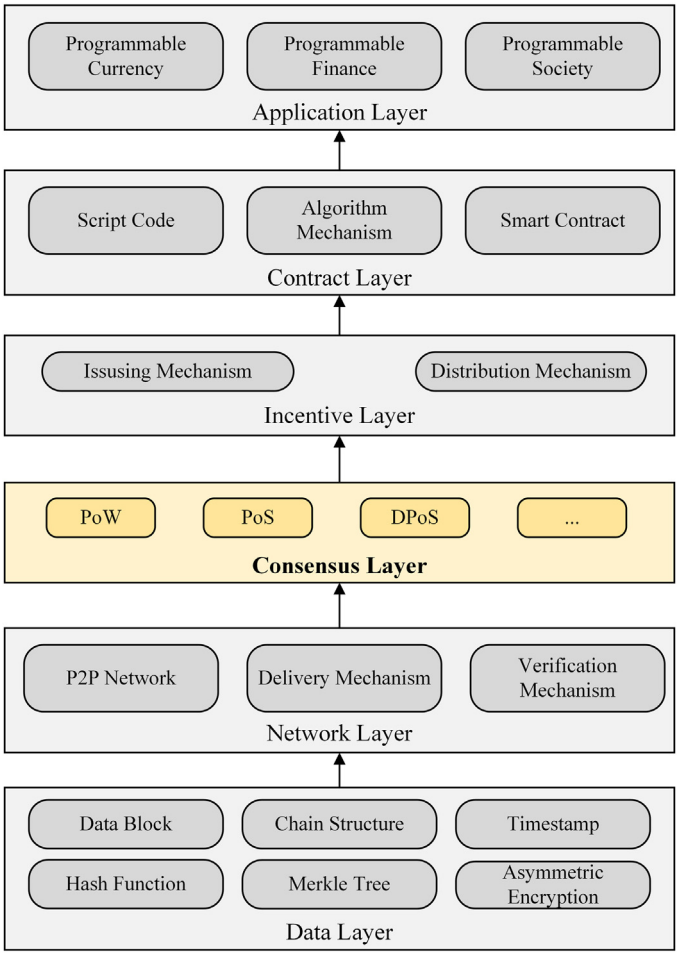

Fig. 1. Blockchain infrastructure model.

2. Background

In this section, we briefly introduce blockchain and give an overview of blockchain consensus.

2.1. Blockchain

The blockchain is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree). Based on the design, the blockchain is resistant to modification of its data. Moreover, the data of the blockchain can be traced back and transparent to everyone. Essentially, the blockchain is also a distributed shared database.

A blockchain is linked by blocks, which are mainly composed of a block header and a block body. The block header consists of a version number, a timestamp, a target hash bit of the current block, a nonce, the hash value of the previous block and a Merkel root. The detailed data of transactions is stored in the block body. Each block is related to the previous block, and each node in the distributed system can verify its correctness at any time. The miner who obtains the accounting right links the current block to the previous block, forming the latest blockchain. Each block is connected in turn to form the longest main chain from the Genesis block to the current block, thus recording the complete history of blockchain data.

The infrastructure model of the blockchain is shown in Fig. 1. Generally speaking, the blockchain system consists of data layer, network layer, consensus layer, incentive layer, contract layer, and application layer (Yang et al., 2019). The data layer encapsulates the underlying data blocks, the related data encryption, and timestamp technologies. The network layer includes distributed networking mechanism, data transmission mechanism, and data verification mechanism. The consensus layer contains various consensus protocols for reaching an agreement among nodes. The incentive layer is used to reward each node for their work on the blockchain, mainly including the issuance and distribution of tokens. The contract layer mainly contains various scripts, algorithms, and smart contracts, which is the basis of the programmable characteristics of blockchain. The application layer covers a variety of application scenarios and cases of blockchain. Since blockchain is a decentralized database, the consensus layer has become the core content of blockchain, serving as a connecting link between the preceding and the following. The consensus layer largely determines the mutual trust between the nodes of the entire blockchain system, and also determines the trust of other users to the data on the blockchain.

According to the mechanism by which nodes are permitted to join the blockchain system, the blockchain can be divided into the permissionless blockchain and the permissioned blockchain (Politou et al., 2021). The permissionless blockchain is also known as the public blockchain. The permissioned blockchain can be divided into the consortium blockchain and the private blockchain according to application scenarios (Zheng et al., 2017; Dib et al., 2018; Xia et al., 2021).

2.2. Consensus protocol

In the early studies, the consensus protocol was mainly aimed at solving the problem of how to reach an agreement among nodes in the distributed system (Pease et al., 1980; Fischer et al., 1985; Biely et al., 2011). With the emergence of malicious nodes (Lamport et al., 1982), the focus of consensus research is gradually developing towards Byzantine fault-tolerant protocol (Castro et al., 1999; Cachin et al., 2001; Castro and Liskov, 2002; Abd-El-Malek et al., 2005). The blockchain, especially the public blockchain, must adopt the BFT consensus protocol because the system needs to resist the attack of malicious nodes (Zheng et al., 2017), while the consortium blockchain and the private blockchain can use the Byzantine fault-tolerant or crash fault-tolerant protocols because of the strict access mechanism for the nodes (Yuan et al., 2018).

Blockchain consensus protocol generally involves two processes: *block generation node election* and *main chain consensus* (Xia et al., 2021). The block generation node election is used to elect the block generation node in the blockchain system, while the main chain consensus is used to reach an agreement on the blockchain data among all nodes in the blockchain system. Among the existing blockchain consensus protocols, some cover both processes, while others cover only one. Based on these two processes, blockchain consensus protocol can be categorized as follows (Yuan et al., 2018; Xia et al., 2021).

2.2.1. Block generation node election

The block generation node election mechanism of blockchain consensus protocol is similar to the leader election problem in the traditional distributed protocol. According to the leader election strategy, the block generation node election mechanism can be divided into the following five types.

(1) Proof-based Consensus

Proof-based consensus elects a leader from the nodes of the blockchain system through some kinds of proof or qualification.

Table 1
Research questions.

Ref.	Question
General questions	
GQ1	Which of the different stakeholders are involved in the blockchain consensus safety and security?
GQ2	What are safety and security problems in the blockchain consensus protocol?
GQ3	Which approaches have been used to guarantee the blockchain consensus safety and security?
GQ4	Which methods are used to validate the research?
GQ5	What are challenges in blockchain consensus safety and security assurance?
Focused questions	
FQ1	What are the attacks against the blockchain consensus protocol?
FQ2	How harmful are attacks in blockchain consensus protocols?
FQ3	What are the deficiencies of existing blockchain consensus safety and security research?
Statistical questions	
SQ1	How much activity about the research of blockchain consensus safety and security has there been in recent years?
SQ2	What is the distribution of publication venues?

The leader is qualified to validate the transactions, package them in a new block, and then broadcast to other nodes over the network. Proof-based consensus is represented by Proof-of-Work (PoW) (Dwork and Naor, 1992; Jakobsson and Juels, 1999) and Proof-of-Stake (PoS) (PoS, 2022).

(2) Voting-based Consensus

Voting-based consensus is a consensus protocol that identifies consensus opinions through a balanced voting system. In each round of consensus, nodes elect the accounting node of the current round by "voting". The node that firstly obtains more than half of the votes will get the accounting right. Voting-based consensus is represented by Paxos (Lamport, 1998) and Raft (Ongaro and Ousterhout, 2014).

(3) Randomness-based Consensus

In the randomness-based consensus protocol, nodes directly determine the accounting nodes of each round in a random way. Randomness-based consensus is represented by Algorand (Gilad et al., 2017) and Proof-of-Elapsed-Time (PoET) (Buntinx, 2017).

(4) Alliance-based Consensus

Alliance-based consensus first elects a group of representative nodes in a particular way, and then the representative nodes take over the accounting rights in turn or by election. Alliance-based consensus is represented by Delegated-Proof-of-Stake (DPoS) (DPoS, 2022).

(5) Combination-based Consensus

Combination-based consensus elects accounting nodes by taking a combination of multiple consensus protocols, such as PoW + PoS consensus (Ren, 2014; Bentov et al., 2014), DPoS + BFT consensus (Zhang et al., 2021), etc.

2.2.2. Main chain consensus

According to whether the block data meets the final consistency, the main chain consensus can be divided into probabilistic consensus and deterministic consensus.

(1) Probabilistic Consensus

In the probabilistic consensus, the block data reach an agreement with a certain probability, and the probability increases gradually as time goes by. It cannot be guaranteed that the block data cannot be changed in the future. This consistency is also called weak consistency. The probabilistic consensus is widely used in the permissionless blockchain, which is represented by the longest chain rule (Nakamoto, 2008) and GHOST (Sompolinsky and Zohar, 2015).

(2) Deterministic Consensus

Probabilistic consensus has a natural tradeoff between transaction delay and security, which limits the application scenarios of blockchain technology. Therefore, some research uses the deterministic consensus to ensure the strong consistency of block data. In the deterministic consensus, the block data cannot be changed once it is agreed. This consistency is also known as strong consistency, which is represented by HoneyBadger (Miller et al., 2016) and Tendermint (Buchman, 2016).

3. Methodology

Guided by Keele (2007) and Petersen et al. (2015), we followed a structured and systematic method to perform the blockchain consensus safety and security survey. The detailed methodology used is described in this section.

3.1. Research questions

Specifying research questions is most important in any systematic review. Our research questions are classified into three categories: General Question (GQ), Focused Question (FQ), and Statistical Question (SQ). Table 1 lists all research questions.

GQs concern general questions related to the safety and security of blockchain consensus protocols. GQ1 refers to the question of WHO takes part in the blockchain consensus safety and security assurance process. GQ2 refers to the question of WHY blockchain consensus safety and security need to be guaranteed, especially the new challenges that do not exist in the traditional consensus protocol. GQ3 refers to the question of HOW to guarantee the blockchain consensus safety and security by specific techniques and methods. In view of different types of safety and security problems, what technologies have been researched to achieve the safety and security of the blockchain consensus protocol? GQ4 considers trustworthiness and reliability of the proposed techniques. Different safety and security assurance approaches can achieve different purposes, and their forms and targets are different. Finally, GQ5 refers to the current challenges of blockchain consensus safety and security assurance.

FQs address the problems specific to the safety and security of blockchain consensus protocol. FQ1 focuses on the types and strategies of consensus attacks. FQ2 focuses on the consequences of consensus protocol attacks and their severity analysis. Finally, FQ3 focuses on the deficiencies of existing consensus protocol safety and security studies.

SQs from a publishing statistical point of view provide another way to reflect the quality of current research. The problem with SQ1 and SQ2 is when and where the research paper was published. It is hoped that they can not only provide the research trend of this topic, but also provide the research maturity of this topic.

3.2. Search strategy and literature selection

In order to have a comprehensive understanding of blockchain consensus safety and security, we selected the following four online literature repositories belonging to publishers of technical research:

- ACM Digital Library
- Elsevier Science Direct

- IEEE Xplore Digital Library
- Springer Online Library

These literature repositories were chosen because most relevant articles about blockchain consensus safety and security are available through them. The search keywords initially used were broad in order to cover as many papers with different uses of terminology as possible. The initial set of keywords included <blockchain consensus> and (<safety> or <security>). In addition, to collect papers as comprehensively as possible, we also conducted a search in the Google academic database using the same keyword set. After completing the search, we manually removed duplicate papers and obtained 1069 papers.

To ensure the quality and relevance of these papers, we further filtered the papers we found with the following exclusion criteria:

(1) Studies not written in English.
(2) Studies not related to the field of computer science.
(3) Studies not related to blockchain consensus safety or security.
(4) Master or Ph.D. theses.
(5) Studies without a full-text.
(6) Studies not yet been published but documented on arXiv.

After filtering, 78 related papers were left. In the final study selection phase, a full text analysis was conducted on the remaining 78 papers. For this step, we used the inclusion strategy. The inclusion criteria aimed at providing answers to the research questions and included one of the following conditions:

(1) The research focused on the safety and security problems existing in the blockchain consensus protocol.
(2) The research focused on the detection, evaluation, or validation techniques for blockchain consensus safety and security.
(3) The research was significantly helpful to the improvement of blockchain consensus safety and security.

After the final study selection phase, a total of 42 papers were selected. Admittedly, it is impossible to find all the papers on blockchain consensus safety and security. However, we are confident that our survey covers the majority of relevant papers.

3.2.1. Quality assessment

It is critical to assess the quality of primary studies. The quality criteria are listed as follows:

Q1: Is there a clear statement about the aim of the research?
Q2: Is there an adequate description of the research context?
Q3: Is there a review about the related work of problems?
Q4: Is there a description of the blockchain consensus safety and security assurance approaches used in the research?
Q5: Has the approach been validated?
Q6: Is the conclusion related to the aim and purpose of research defined?
Q7: Is there a clear statement of findings?
Q8: Does the study recommend further research?

Table 2 shows the results of applying the quality assessment criteria to each primary study, where the ✓ indicates "yes" and ✗ indicates "no". Although Karame et al. (2012) and the other 22 papers do not satisfy all of our designed criteria, we decided not to eliminate them since the absence of Q3 and Q8 does not affect the study outcomes.

3.3. Data extraction

Finally, this paper designs a data extraction template to collect information that addresses the research questions, mainly for the 42 primary studies. Table 3 shows each of the data extraction details of the primary studies, which enables this paper to extract all the details from the primary studies and understand how these studies addressed the research questions related to the blockchain consensus safety and security.

4. Results

In this section, we present results from 42 assessed primary studies related to our research topic. An overview of all primary studies is shown in Table 4, where id, citation, and the basic description of each study are included. We then answer our research questions in the following subsections through elaborative information synthesis.

A theoretical research framework has been summarized in Fig. 2 based on the overview of 42 important studies, which shows the research roadmap in this area, and shows that the roadmap includes following four key steps: ① start from two following two directions, respectively: the safety of consensus protocol and the security of consensus protocol; ② find the problems and challenges in above two directions; ③ find feasible approaches to solve corresponding safety or security problems and challenges; and ④ validate the effectiveness and efficiency of their approaches by theory analysis and experimental study.

4.1. Stakeholders

In the process of reaching a consensus on the blockchain, many blockchain consensus stakeholders are involved in the safety and security assurance of blockchain consensus protocols to different degrees. However, since stakeholders from diverse perspectives have different accessibility to the blockchain system, their emphasis may be disparate. Clarifying the relationship between different stakeholders and the safety and security of consensus protocols is of guiding significance to solve the safety and security problems of consensus protocols and guarantee the safety and security of consensus protocols.

4.1.1. Consensus protocol designer

The standard development of consensus protocols needs perfect design text, and the defects of design text may mislead developers. Complex and recurring problems may exist in the application scenarios of different consensus protocols. Designers need to abstract these problems and propose conceptual design solutions. With the rapid development of the blockchain system and security development environment, new security risks and various attack strategies continue to emerge, and the lack of design scheme will lead to the fragile consensus protocol. At the most basic level, designers need to ensure that the design consensus protocol is safe within fault tolerance.

4.1.2. Consensus protocol developer

The quality of the consensus protocol is directly influenced by the developers of the consensus protocol. If developers do not strictly follow the design of consensus protocols, safety and security problems such as function loss and error may occur during the use of consensus protocols. At the same time, developers need to consider some practical issues during the development process, including network parameters, throughput, latency, node number, and so on. These problems may affect the safety and security of the consensus protocol.

Table 2
Quality assessment for primary studies.

Primary study	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Karame et al. (2012)	✓	✓	✓	✓	✓	✓	✓	✗
Eyal and Sirer (2014)	✓	✓	✓	✓	✓	✓	✓	✓
Bentov et al. (2014)	✓	✓	✗	✓	✓	✓	✓	✓
Heilman et al. (2015)	✓	✓	✓	✓	✓	✓	✓	✗
Lewenberg et al. (2015)	✓	✓	✓	✓	✓	✓	✓	✓
Luu et al. (2016)	✓	✓	✓	✓	✓	✓	✓	✗
Kogias et al. (2016)	✓	✓	✓	✓	✓	✓	✓	✓
Gervais et al. (2016)	✓	✓	✓	✓	✓	✓	✓	✗
Sapirshtein et al. (2016)	✓	✓	✓	✓	✓	✓	✓	✗
Nayak et al. (2016)	✓	✓	✓	✓	✓	✓	✓	✓
Bonneau (2016)	✓	✓	✗	✓	✓	✓	✓	✗
Pass et al. (2017)	✓	✓	✓	✓	✓	✓	✓	✗
Kwon et al. (2017)	✓	✓	✓	✓	✓	✓	✓	✓
Liao and Katz (2017)	✓	✓	✓	✓	✓	✓	✓	✓
Chen et al. (2017b)	✓	✓	✓	✓	✓	✓	✓	✓
Solat and Potop-Butucaru (2017)	✓	✓	✗	✓	✓	✓	✓	✗
Kiayias and Panagiotakos (2017)	✓	✓	✗	✓	✓	✓	✓	✓
Zhang and Preneel (2017)	✓	✓	✗	✓	✓	✓	✓	✓
Li et al. (2017)	✓	✓	✓	✓	✓	✓	✓	✗
Natoli and Gramoli (2017)	✓	✓	✓	✓	✓	✓	✓	✓
Kiffer et al. (2018)	✓	✓	✓	✓	✓	✓	✓	✓
Wei et al. (2018)	✓	✓	✓	✓	✓	✓	✓	✗
Gaži et al. (2018)	✓	✓	✗	✓	✓	✓	✓	✗
Ritz and Zugenmaier (2018)	✓	✓	✓	✓	✓	✓	✓	✓
Zou et al. (2019)	✓	✓	✓	✓	✓	✓	✓	✓
Yang et al. (2019)	✓	✓	✓	✓	✓	✓	✓	✓
Tholoniat and Gramoli (2019)	✓	✓	✓	✓	✓	✓	✓	✗
Saltini (2019)	✓	✓	✓	✓	✓	✓	✓	✓
Xian et al. (2019)	✓	✓	✓	✓	✓	✓	✓	✗
Zhang and Preneel (2019)	✓	✓	✓	✓	✓	✓	✓	✓
Szalachowski et al. (2019)	✓	✓	✓	✓	✓	✓	✓	✓
de Oliveira et al. (2020)	✓	✓	✓	✓	✓	✓	✓	✓
Bissias and Levine (2020)	✓	✓	✓	✓	✓	✓	✓	✗
Xiao et al. (2020)	✓	✓	✗	✓	✓	✓	✓	✓
Yu et al. (2020)	✓	✓	✓	✓	✓	✓	✓	✗
Li et al. (2020)	✓	✓	✓	✓	✓	✓	✓	✓
Ekparinya et al. (2020)	✓	✓	✗	✓	✓	✓	✓	✗
Otte et al. (2020)	✓	✓	✓	✓	✓	✓	✓	✓
Sun et al. (2020)	✓	✓	✓	✓	✓	✓	✓	✓
Aluko and Kolonin (2021)	✓	✓	✓	✓	✓	✓	✓	✗
Zhang and Jacobsen (2021)	✓	✓	✓	✓	✓	✓	✓	✗
Karakostas and Kiayias (2021)	✓	✓	✗	✓	✓	✓	✓	✗

Table 3
Data extraction form.

ID	Field	Description	Research question
Content information of article			
1	Title	Title of the primary study	
2	Abstract	Abstract of the primary study	GQ2
3	Objective	What are the objectives of the study?	GQ2
4	Personnel	Who is involved in blockchain consensus safety and security?	GQ1
4	Problem	What safety and security problems do blockchain consensus face?	GQ2
5	Approach	Which safety and security assurance approaches are applied to blockchain consensus?	GQ3
6	Validation	Which is the approach used to validate the study?	GQ4
7	Validity	Limitation, threat to validity	
8	Future Work	Future work and challenges of the study	GQ5
9	Conclusion	Conclusion of the study	GQ2
Reference information of article			
10	Authors	Authors of the primary study	
11	Year	Publication year of the primary study	SQ1
12	Type	Publication type of the primary study	SQ2
13	Source	Name of publication where the primary study was published	SQ2

4.1.3. Consensus protocol tester

Testing is an important part of blockchain consensus protocols before they are officially released. It has an impact on the safety and security of consensus protocols. Effective testing can verify the actual results of the consensus protocol conform to design expectations. If the tester does not have a clear design of the consensus protocol and its functional requirements, the tester cannot confirm whether the consensus protocol meets expectations. Secondly, if testers are not familiar with the consensus protocol, they cannot effectively analyze the testing report, thus affecting the overall development efficiency of the consensus protocol and ignoring the potential threats in the consensus protocol.

4.1.4. Consensus protocol maintainer

In the blockchain, especially the consortium blockchain and the private blockchain, maintainers are required to monitor the actual operation of consensus protocols in a timely manner and take corresponding measures to abnormal behaviors. Therefore,

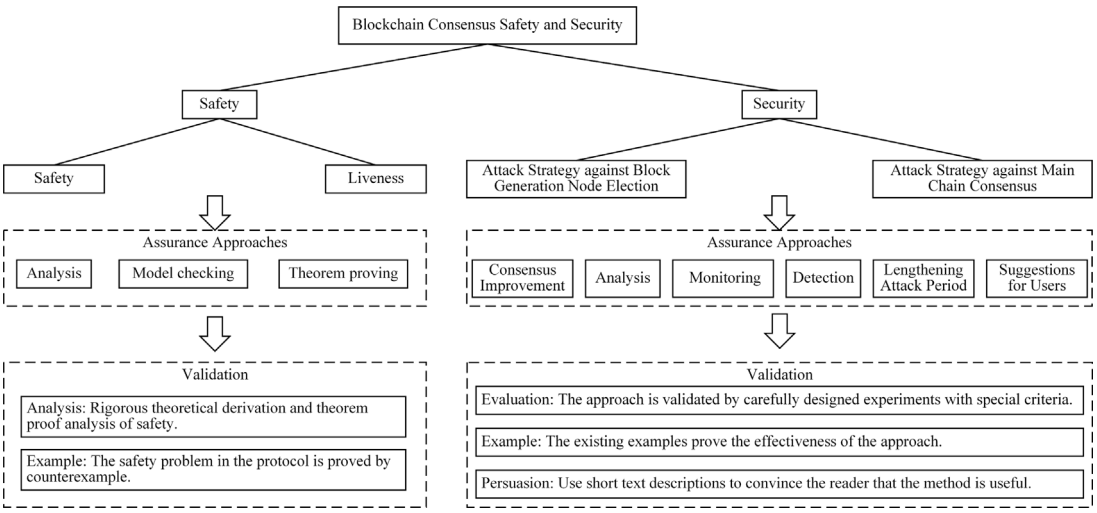

Fig. 2. Theoretical research framework of consensus protocol safety and security.

Table 4
An overview of primary studies.

ID	Primary study	Description
S1	Karame et al. (2012)	Analyzing and countering double-spending in fast Bitcoin payments
S2	Eyal and Sirer (2014)	A backward-compatible progressive modification to raise the threshold for selfish mining
S3	Bentov et al. (2014)	Proof-of-Activity (PoA), combining PoW with PoS to defend against attacks on Bitcoin
S4	Heilman et al. (2015)	Countermeasures that make eclipse attacks more difficult
S5	Lewenberg et al. (2015)	The inclusive protocol, which can increase the cost of attack
S6	Luu et al. (2016)	Security analysis for how ELASTICO prevents potential threats and works securely
S7	Kogias et al. (2016)	ByzCoin, mitigating double spending and selfish mining attacks by producing collectively signed transaction blocks within one minute of transaction submission
S8	Gervais et al. (2016)	A quantitative framework to analyze the security and performance implications of various consensus and network parameters of PoW blockchains
S9	Sapirshtein et al. (2016)	An efficient algorithm which can compute an optimal selfish mining policy
S10	Nayak et al. (2016)	Expanding the mining strategy space to include "stubborn" strategies and combining with an eclipse attack
S11	Bonneau (2016)	Analyzing bribery attacks and the mitigating factors
S12	Pass et al. (2017)	Analysis of the blockchain protocol in asynchronous networks
S13	Kwon et al. (2017)	Analyzing fork after withholding (FAW) attacks and discussing countermeasures
S14	Liao and Katz (2017)	A game-theoretic analysis and simulation of the whale attack
S15	Chen et al. (2017b)	A theoretical framework for evaluating blockchain systems based on PoET
S16	Solat and Potop-Butucaru (2017)	Preventing block withholding (BWH) attacks by using a novel timestamp-free technique
S17	Kiayias and Panagiotakos (2017)	Formally verifying that the GHOST backbone protocol is sufficient to construct a robust transaction ledger
S18	Zhang and Preneel (2017)	A backward-compatible defense against selfish mining in Bitcoin
S19	Li et al. (2017)	Analyzing and evaluating the security of two proposed PoS protocols
S20	Natoli and Gramoli (2017)	Arguing for a non-forkable blockchain design that protects against balance attacks
S21	Kiffer et al. (2018)	A simple Markov-chain based method for analyzing consistency properties of blockchain protocols
S22	Wei et al. (2018)	Setting honest miner against long delay attacks
S23	Gaži et al. (2018)	Several mechanisms which can prevent stake-bleeding attacks
S24	Ritz and Zugenmaier (2018)	Monte Carlo simulation, quantifying the impact of tertiary blocks on the profitability of selfish mining and the security of blockchain in Ethereum
S25	Zou et al. (2019)	Proof-of-Trust (PoT), a new consensus protocol that can defend against Sybil attacks
S26	Yang et al. (2019)	Delegated Proof of Stake With Downgrade (DDPoS), improving the security by quickly replacing the malicious nodes
S27	Tholoniat and Gramoli (2019)	Formal verification of blockchain Byzantine fault tolerance
S28	Saltini (2019)	Analysis of Istanbul Byzantine Fault Tolerance (IBFT) liveness
S29	Xian et al. (2019)	Improving existing PoS and PoW algorithms to resolve censorship attacks
S30	Zhang and Preneel (2019)	A multi-metric evaluation framework, which can quantitatively analyze PoW protocols' chain quality and attack resistance
S31	Szalachowski et al. (2019)	StrongChain, improving the security by mitigating known attack vectors and preventing new ones
S32	de Oliveira et al. (2020)	Blockchain reputation-based consensus (BRBC), efficient to expel all nodes that acted with more than 50% of malicious actions
S33	Bissias and Levine (2020)	Bobtail, improving blockchain security with low-variance mining
S34	Xiao et al. (2020)	An analytical model, assessing the impact of network connectivity on the consensus security of PoW blockchain
S35	Yu et al. (2020)	Formally proving the safety and liveness properties of OHIE
S36	Li et al. (2020)	Robust-Proof-of-Stake (RPoS), effectively avoiding coin age accumulation and Nothing-at-Stake attack
S37	Ekparinya et al. (2020)	Discussing the attack of the clones against Proof-of-Authority and countermeasures
S38	Otte et al. (2020)	TrustChain, a Sybil-resistant scalable blockchain
S39	Sun et al. (2020)	Voting-based decentralized consensus (VDC), improving the efficiency and security of consortium blockchain
S40	Aluko and Kolonin (2021)	Safety and security analysis for Proof-of-Reputation(PoR)
S41	Zhang and Jacobsen (2021)	Prosecutor, a new BFT consensus protocol which can dynamically penalize suspected faulty behavior and suppress Byzantine servers over time
S42	Karakostas and Kiayias (2021)	Putting forth the checkpointing mechanism as a protection from 51% attacks

Table 5
Distribution of primary studies by stakeholders.

Primary studies	Stakeholders				
	Designers	Developers	Testers	Maintainers	Users
S1		✓			✓
S2		✓			
S3	✓		✓		
S4		✓			✓
S5	✓		✓		
S6	✓		✓		
S7	✓		✓		
S8		✓			
S9		✓	✓		
S10		✓			✓
S11					✓
S12			✓		
S13				✓	
S14			✓		
S15	✓		✓		
S16		✓			
S17			✓		
S18		✓			
S19	✓		✓		
S20			✓		
S21			✓		
S22		✓	✓		
S23	✓		✓		
S24			✓		
S25			✓		
S26	✓		✓		
S27			✓		
S28			✓		
S29		✓			
S30			✓		
S31	✓	✓	✓		
S32	✓		✓		
S33	✓		✓		
S34			✓		
S35	✓		✓		
S36	✓		✓		
S37			✓		
S38	✓		✓		
S39	✓		✓		
S40	✓		✓		
S41	✓		✓		
S42		✓			
Total	17	12	31	1	4

maintainers must be familiar with the design and development of consensus protocols and have a good understanding of the potential risks of consensus protocols. Maintainers also must have sufficient professional skills to respond to and handle unexpected security problems to ensure the security of consensus protocols.

4.1.5. Consensus protocol user

The users of the consensus protocol are nodes in the blockchain system, mainly including miners, traders, etc., who use the consensus protocols to mine, trade, and so on in the blockchain system. They focus on whether the consensus protocols provide the expected functionality and their own revenues. In most cases, they do not care whether the consensus protocols are modified if the changes do not affect their usage experience. However, some users do not abide by the usage rules in order to obtain greater benefits, which may bring security risks to the consensus protocol.

For research question GQ1 (Which of the different stakeholders are involved in the blockchain consensus safety and security?), statistical results from the primary studies are shown in Table 5. Consensus tester is the most frequently cited stakeholder emerging in about three quarters (73.8%, 31/42) of the primary studies, whereas consensus designer is cited in more than one-third of the studies (40.5%, 17/42). Only one study (S13) is conducted from the perspective of consensus maintainers. This is mainly because consensus protocols are difficult to upgrade and maintain once deployed. It is noteworthy that most studies involve more than one stakeholder. Study S31 studies the safety and security of consensus protocol from three perspectives of designers, developers, and testers. In addition, we are able to find 17 studies involving designers, all of which are also associated with testers. This is because researchers test and verify more secure consensus protocols as soon as they are designed.

4.2. Safety and security problems

4.2.1. Safety problems

The consensus protocol safety refers to the system reaching an unsafe state due to the triggering of a condition during the process of reaching an agreement. In this state, the system crashes or even breaks down, which leads to abnormal system operation and major safety incidents. Many safety problems exist in the consensus protocols. In this section, we will introduce the safety problems that existed in the consensus protocols.

Before blockchain technology was proposed, the consensus protocol studied by researchers was mainly non-Byzantine fault-tolerant protocol, or crash fault-tolerant protocol, which assumes that there are only fault nodes in the system but no malicious nodes. Therefore, when researchers study the safety of consensus protocol, they focus on the analysis of the safety and liveness, which was first proposed by Lamport in 1977 (Lamport, 1977). Safety research on blockchain consensus protocols also mainly includes safety and liveness problems.

(1) Safety

A safety property is one which states that something will not happen (Lamport, 1977; Alpern and Schneider, 1987). For example, the honest nodes in the blockchain eventually fail to agree on the data, or the nodes eventually agree on the malicious data from the malicious nodes. In the blockchain consensus protocol, safety is mainly reflected in validity and consistency.

(a) Validity

Three studies (S6, S25, and S27) focused on validity when studying blockchain consensus safety. Validity means the value that the honest nodes ultimately agree on must be the value that comes from the honest node's proposal within the ratio of malicious nodes allowed by consensus protocols. If the honest node reaches a consensus on the malicious node's proposal, it means that the consensus protocol has been broken by the malicious node.

(b) Consistency

Seven studies (S6, S7, S12, S21, S25, S27, and S35) focused on consistency when studying blockchain consensus safety. Consistency is the consistency of data, which can be understood as the consistency of data values across multiple nodes in a distributed system. Consistency is one of the most important and basic characteristics of consensus protocols. However, due to the design of some consensus protocols, the nodes in the blockchain system cannot achieve data consistency, which eventually leads to the collapse of the blockchain system. In addition, the consistency of consensus protocol can be divided into strong consistency and weak consistency.

(2) Liveness

A liveness property is one which states that something must happen (Lamport, 1977; Alpern and Schneider, 1985, 1987). For example, the process P2 will not stay in the mutually-exclusive critical region forever, so that P1 will eventually be able to enter the mutually-exclusive critical region. That is, the situation in which P1 enters the mutually-exclusive critical region will happen eventually.

A total of eight studies (S7, S12, S17, S25, S27, S28, S35, and S40) have examined the liveness of blockchain consensus

protocols. In the blockchain consensus protocol, liveness is mainly reflected in termination. Termination concerns the problem of whether honest nodes eventually agree on a value. That is, the honest nodes immediately proceed to the next step after receiving enough responses regardless of the network status. If the number of responses is insufficient, the honest nodes will exit the consensus after a certain time or number of voting rounds, so as to avoid the situation of an infinite loop or deadlock.

4.2.2. Security problems

In the blockchain consensus protocol, security problems are mainly manifested as various attacks. In this section, we present the attacks against the current blockchain consensus protocol, including a description, classification, summary of the severity of their damage, and consensus protocols in which they apply.

Malicious nodes attempt to attack the blockchain consensus protocol in order to achieve their sabotages and gain profits. As described in Section 2.2, the blockchain consensus protocol generally involves two processes: block generation node election and main chain consensus. An attacker attacks a consensus protocol from these two processes.

(1) Attack Strategy against Block Generation Node Election

The attack strategy against block generation node election increases the chances that the malicious node will become a block generation node by a number of means. As discussed in the literature, the block is the basic structural unit of the blockchain, which means that controlling the generation of the block can control the blockchain system. There are mainly the following attacks.

(a) Sybil Attack (AT1)

Researchers have been working on the Sybil attack before blockchain technology was even proposed (Newsome et al., 2004). With the increasing use of blockchain technology, the Sybil attack has become a famous area of researchers (Otte et al., 2020). The Sybil attack is an attack that damages the network by nodes, targeting the entire network rather than individual nodes. The Sybil attack is a form of attack that works in the peer-to-peer network: the attacker uses a single node to forge multiple identities in the peer-to-peer network, so as to weaken the redundancy of the network, reduce the robustness of the network, and monitor or interfere with the normal activities of the network. Therefore, the Sybil attack can also attack any consensus protocol of peer-to-peer network.

(b) Eclipse Attack (AT2)

The eclipse attack is a way of attacking distributed networks by isolating and attacking nodes with public IP addresses (Heilman et al., 2015). The eclipse attack involves the adversary targeting a specific node (as opposed to the network as a whole) so as to cut off all of their inbound/outbound communications with other peers (which effectively suffocates the victim). A successful eclipse attack enables a would-be bad actor to isolate and subsequently prevent their target from attaining a true picture of real network activity and the current ledger state. The opponent then can cause general disruption and segregate the targeted peer from other nodes to mount further attacks.

The eclipse attack is an attack against a peer-to-peer network. By controlling the access connection of a node, the attacker makes the node isolated and only communicates with the malicious node. Therefore, the eclipse attack can attack any consensus protocol of the peer-to-peer network.

(c) Grinding Attack (AT3)

The grinding attack, also called stake grinding attack is to control the seed of random value to get more chance to be the block producer (Li et al., 2017). In PoS, the probability to be the block producer is proportional to stakes. This probability should rely on random values at the end, however, it is very hard to create a real random value in the blockchain. Some PoS consensus protocols use block headers, block creation time, and many other values as a seed of random value, and it means attackers may try to set or modify these values to get a random value that they want. Since this calculation is not easy, it may not be practical in some cases. In addition to the PoS consensus protocol, some voting-based consensus may also be subject to this attack.

(d) Collusion Attack (AT4)

The collusion attack is separated into two different classes (Zou et al., 2019). The first one is a collusion between the leader and some of the validators. Leaders prioritize nominating the validators that they collude with so that the collators remain leaders. The second class is the collusion between the leader and the ledger management nodes. When Byzantine nodes reach a certain number, the attacker could vote to make a blockchain system accept some illegal transactions. The collusion attack focuses on voting-based consensus protocols, especially start-up blockchain systems where the number of nodes is small.

(e) Cartel Attack (AT5)

Generally speaking, as long as the number of honest people in the consensus system reaches a certain number, no matter how the rest of the people do evil or attack the network, the network can run stably. However, in order to gain profits, some validators often form alliances with larger token holders to reach the threshold of system destruction. This alliance we call Cartel (Xian et al., 2019). Cartel will ignore some or all blocks generated by miners which are not members of Cartel. Different from the collusion attack, Cartel attack mainly focuses on proof-based consensus protocols, and the attackers are validators in the blockchain system. Cartel attack can overwhelm the system, either stop generating blocks, or cause the network to crash, or deny certain transactions when a block is generated. Regrettably, there is no solution to this attack currently. We can only hope that most people are honest.

We summarized the attack strategies against block generation node election, including the attack objectives, attack ways, and attack consequences, as shown in Table 6.

(2) Attack Strategy against Main Chain Consensus

The attack strategy against main chain consensus makes it impossible for nodes in the blockchain system to reach a consensus on the main chain through some means such as forking the blockchain or affecting the communication between nodes. The blockchain is a chain that starts from the Genesis block and is linked by one block after another. By forking the blockchain, malicious nodes can delay or prevent the agreement of the nodes in the system, thus creating conditions for the destruction of the blockchain system to obtain illegal benefits. There are mainly the following attacks.

(a) Double-spending Attack (AT6)

The double-spending attack is one of the most common and most harmful attacks in blockchain digital currencies. As the name implies, it is the same single digital token that can be spent more than once by forking the blockchain (Karame et al., 2012). This will not only cause economic losses to merchants, but also result in wasted efforts of miners, and over time, there will be no trust in the blockchain system. In view of the forks of proof-based consensus protocols in the process of mining, the double-spending attack often occurs in these consensus protocols, such as PoW and PoS. According to the different attack modes, the double-spending attack can be divided into the following five kinds of attacks.

– *51% Attack (Karame et al., 2012).* The 51% attack is an attack derived from the specific rules of the blockchain consensus protocol (Sayeed and Marco-Gisbert, 2019). An attacker can launch an attack with more than 50% computational power, stake, or other "power". In Bitcoin, for example, Attacker A buys goods from

Table 6
Summary of attack strategy against block generation node election.

Attack name	Attack objective	Attack ways	Attack consequence
Sybil attack (Otte et al., 2020)	Whole network	Using a single node to forge multiple identities	Weakening the redundancy of the network, reducing the robustness of the network, and monitoring or interfering with the normal activities of the network
Eclipse attack (Heilman et al., 2015)	Nodes with public IP addresses	Isolating nodes	Preventing target nodes from attaining a true picture of real network activity and the current ledger state
Grinding attack (Li et al., 2017)	Random function	Controlling the seed of random value	Controlling the block generation
Collusion attack (Zou et al., 2019)	Blockchain system	Leader colluding with validators or ledger management nodes	Accepting illegal transactions
Cartel attack (Xian et al., 2019)	Blockchain system	Validators forming alliances with larger token holders	Overwhelming the system, stopping block generation, causing network to crash, denying transactions

Merchant B and pays 10 bitcoins. Merchant B sends the goods to Attacker A until the transaction is 6 blocks deep. At the same time, Attacker A transfers the same bitcoin to himself and generates a new block. Since Attacker A can generate blocks faster than other miners with more than 51% computational power, Attacker A can construct a longer blockchain. As soon as Attacker A releases the longer chain, the entire network of miners will recognize the chain as the main chain. Then the previous chain will be discarded and the above transactions will be rolled back. In the end, Attacker A gets the goods without spending any bitcoins, achieving the double-spending attack. Although the 51% attack is common in proof-based consensus, the 51% attack is less likely to occur in the PoS consensus protocol because the attack will reduce the market value of tokens.

– *Finney Attack (Kogias et al., 2016).* The name "Finney" comes from Hal Finney, who was the first to describe a 0 confirmed (unconfirmed) transaction for the double-spending attack (Rathod and Motwani, 2018). The Finney attack is primarily achieved by controlling the broadcast time of the block, targeting the merchants that accept 0 confirmation. Suppose the attacker mines a block that contains a transaction in which Address 1 transfers a certain number of tokens to Address 2, but both addresses belong to the attacker. But instead of broadcasting the block, the attacker immediately finds a merchant and sends the tokens to the merchant's Address 3 with his Address 1. After the transaction sent to the merchant is broadcast, if the merchant accepts 0 confirmation, the attacker will broadcast the block he had previously mined, and the transaction sent to him will precede the transaction sent to the merchant. For the attacker, by controlling the broadcast time of the block, "double-spending" of the same token is achieved.

– *Vector76 Attack (Kogias et al., 2016).* The Vector76 attack, is also known as the "One Confirmation Attack", which means that a transaction can be rolled back even if it is confirmed (Rathod and Motwani, 2018). The Vector76 attack can occur if the e-wallet meets the following points: the e-wallet accepts one confirmation payment, the e-wallet accepts a direct connection from another node, and the e-wallet uses a static IP address for nodes.

– *Replay Attack (Li et al., 2017).* In blockchain technology, the replay attack means that "transactions on one chain are often legal on the other". Suppose that two hard-forked chains have the same protocols for generating addresses and private keys and the same transaction format so that a transaction on one chain is likely to be perfectly legal on the other. Therefore, transactions initiated on one of the chains may also be confirmed if they are re-broadcast on the other chain. Once this attack occurs, it has a similar effect to the double-spending attack.

– *Censorship Attack (Xian et al., 2019).* Strictly speaking, the censorship attack is a variant of the 51% attack. The censorship attack is defined as "a majority attacking coalition building a chain which rejects transactions or messages that an ordinary validator, miner or client would accept". However, considering that attackers are a close-knit group, the network latency between them is minimal, while other honest validators are distributed in different locations. Therefore, the censorship attack can occur even if the attackers do not account for the majority. In addition to 51% attacks (if the attackers are in the majority), the censorship attack can cause other side effects. Censorship attacks can achieve huge benefits at a low cost and are even harder to identify.

(b) Selfish Mining (AT7)

Unlike some of the other attacks on blockchains and cryptocurrencies, selfish mining is not aimed at disrupting the proper functioning of the blockchain network, but is more purely for greater profit. Miners do not broadcast immediately after finding a block, but broadcast under certain conditions, invalidating other miners' work for a certain period of time (Eyal and Sirer, 2014). As a result, miners choose to concentrate their computational power together to form a pool for joint mining, with the aim of obtaining greater profit and even controlling the blockchain.

Since miners in the PoW consensus works on finding a difficult proof of work for their block, they can achieve selfish mining by commanding a certain amount of computational power. They can also collude with other miners, or command other miners' power through eclipse attacks, thus building pools and increasing the success rate of selfish mining.

(c) Bribery Attack (AT8)

The bribery attack is an attack that recombines blocks that span from days to months to modify part of the block data that benefits the attacker (Bonneau, 2016). For example, Attacker A purchases a commodity from Merchant B with some tokens. Merchant B allows Attacker A to take the commodity without confirmation of the block. At present, Attacker A immediately bribes most nodes in the network to accept the modified transactions. As long as the cost of the bribery is lower than the value of the commodity, the attack is successful.

(d) Stubborn Mining (AT9)

Stubborn mining is a much more optimal attack strategy than selfish mining (Nayak et al., 2016). The attacker stubbornly mines on his own private blockchain until the private blockchain he builds becomes the main chain, no matter how many blocks other miners have mined. Such a stubborn approach can bring more profits to attackers than selfish mining. Attackers can also combine stubborn mining with eclipse attacks to make a larger profit with a smaller cost.

(e) BWH Attack (AT10)

In a mining pool, the owner of the pool distributes rewards to the miners based on their efforts. Attackers can thus sabotage the pool by submitting a partial proof of work (PPoW) instead of a

full proof of work (FPoW) and make a profit (Kwon et al., 2017). If the attacker finds the FPoW, he will throw it away. Suppose that Miner A's computational power accounts for 30% of the total blockchain system, and the Mining Pool B's computational power accounts for 40%. If A mines normally, he can get 30% of the total revenue. But now he divides his computational power into 20% and 10%. Miner A mines in the Mining Pool B with the 10% computational power, and mines normally with the 20% computational power. Since miners only submit PPoW to the pool, the computational power of the blockchain system is 90%. Miner A's real revenue is $2/9+4/9\times 1/(1+4)\approx 0.311$, which is larger than 0.3. Therefore, if the computational power is divided properly, the attackers can make a larger profit and sabotage the pool.

(f) FAW Attack (AT11)

The FAW attack is a combination of the BHW attack and selfish mining (Kwon et al., 2017). Unlike the BHW attack, attackers will keep the FPoW instead of throwing it away if he finds it. When a new block is generated by another miner outside the mining pool, the attacker submits a previously discovered FPoW to cause a fork.

(g) Balance Attack (AT12)

Because there is an upper limit to the number of nodes that each node can communicate with, the attacker can transiently disrupt communications between subgroups of similar computational power (Natoli and Gramoli, 2017). During this time, the attacker issues transactions in one subgroup, say the transaction subgroup, and mines blocks in another subgroup, say the block subgroup, up to the point where the tree of the block subgroup outweighs, with high probability, the tree of the transaction subgroup. The novelty of the balance attack is to leverage the GHOST protocol that accounts for sibling or uncle blocks to select a chain of blocks. This strategy allows the attacker to mine a branch possibly in isolation of the rest of the network before merging its branch to one of the competing blockchain to influence the branch selection process.

(h) Whale Attack (AT13)

When miners are rational, they tend to package transactions with a larger transaction fee (Liao and Katz, 2017). In this case, the attacker will try to "bribe" other (rational) miners to mine on the fork by issuing a whale transaction (i.e., a transaction with an anomalously large transaction fee) that is only valid on the forked branch. The whale attack not only disrupts the consistency of blockchain but also profiteers for attackers. Currently, the whale attack mainly undermines the consensus protocol that adopting this consensus blockchain could fork, especially PoW.

(i) Long-Range Attack (AT14)

The attacker creates a longer blockchain branch that starts with the Genesis block and attempts to replace the current legal backbone (Li et al., 2017; Gaži et al., 2018; Deirmentzoglou et al., 2019). This way of the attack is called the long-range attack. At present, the long-range attack falls into three different categories, namely the simple attack, the posterior corruption attack, and the stake bleeding attack.

– *Simple Attack (Li et al., 2017; Gaži et al., 2018).* The simple attack is the most common long-range attack, which is often confused with the posterior corruption attack in many studies. The simple attack is an attack in which an attacker creates as many blocks as possible on a fork chain over a unit of time, exceeding the length of the original main chain. On the fork chain, there is often only one validated person who dominates the fork. Therefore, he can ignore the blocks generated by other nodes, only generate his own blocks, and speed up the generation of his own blocks. Then the block generation speed of the fork chain may exceed that of the main chain. The solution to this attack is also very simple. It can be solved by validating the timestamp of each block generation time. For unreasonable timestamps, such as multiple blocks generating at the same time, it is directly considered unreasonable. The main chain can be determined correctly.

– *Posterior Corruption Attack (Gaži et al., 2018).* The posterior corruption attack is more advanced than the simple attack. In this attack, the fork chain validator accelerates the length surpassing of the fork chain by obtaining the private key of the old verifier. Typical defenses are checkpointing or key-evolving cryptography.

– *Stake Bleeding Attack (Gaži et al., 2018).* The fork chain validator speeds up the block output of the fork chain by extending the block generation time of the main chain and accumulating the stake of the fork chain. This is the stake bleeding attack. This attack can be addressed by adopting a strategy of moving checkpoints. Setting checkpoints at a specified block height and making sure that the block transactions prior to the checkpoint are immutable can prevent this attack. Since it costs less to create a block in the PoS consensus, an attacker can create another chain at a lower cost to launch a long-range attack.

(j) Nothing at Stake (AT15)

Suppose that an attacker forks the current chain, there is no need for the "miner" holding the coin to figure out which chain will win. The best strategy is to mine on all branches at the same time, because the stake owner will be of benefit regardless of which branch wins in the end (Li et al., 2017). Once such an attack is successful, a chain could split into multiple chains, which causes many problems.

(k) Delay Attack (AT16)

By delaying the information communication of the nodes in the system, the attacker makes the block unable to propagate to each node immediately, thus thwarting the growth rate of the honest chain (Wei et al., 2018). At the same time, the attacker mines his own private chain so that the private chain is longer than the honest chain. An attacker can achieve a double-spending through the delay attack, or fork the blockchain.

(l) Feather-forking Attack (AT17)

In the feather-forking attack, an attacker publicly claims to fork the blockchain to invalidate all blocks confirming the target transactions (Zhang and Preneel, 2019). The attacker will continue to mine on the forked chain until the main chain reaches a certain length. In order to avoid loss, honest miners have to compromise with the attacker to join the attacker in the censorship. Although the attack may seem unprofitable, once successful, the attacker can approve or decline transactions at will, becoming the de facto owner of the blockchain system.

(m) Cloning Attack (AT18)

The clone attack is an attack against the Proof-of-Authority consensus protocol (Ekparinya et al., 2020). One or two sealer attacker(s) convince half of the honest sealers that the transaction has been properly committed by cloning the private key. The attacker then deletes the transaction and a double-spending attack can be successfully launched. Thanks to the cloning, the attacker only needs to delay the message between the two halves of honest sealers in order to convince half of the honest sealers that the transaction has been committed.

(n) Coinage Accumulation Attack (AT19)

The attack usually exists in the consensus that the nodes compete for creating blocks with token holding time rather than token holding amount. For example, the PoS consensus protocol is based on the coinage to choose which miners can package blocks. The coinage is an index calculated according to the amount and time of holding the tokens. The more the coinage is accumulated, the greater the probability that the miners will succeed in finding a block. With the block packaged, the corresponding miner's coinage is reset to zero. In this case, in order to improve their chances of finding a block, miners will accumulate their own

Table 7
Summary of attack strategy against main chain consensus.

Attack name	Attack objective	Attack ways	Attack consequence
Double-spending attack (Karame et al., 2012)	Merchants, miners, and blockchain system	A sum of money being spent more than once by forking the blockchain	Causing economic losses to merchants, wasting efforts of miners, blockchain losing trust
Selfish mining (Eyal and Sirer, 2014)	Miners	Not broadcasting immediately after finding a block, but broadcast under certain conditions	Invalidating other miners' work, forming the centralized mining pool
Bribery attack (Bonneau, 2016)	Merchants	Bribing most nodes in the network to accept the modified transactions	Part of the block data being modified
Stubborn mining (Nayak et al., 2016)	Miners	Stubbornly mining until the private blockchain becomes the main chain	Invalidating other miners' work, forming the centralized mining pool
BWH attack (Kwon et al., 2017)	Miners and mining pools	Submitting a PPoW instead of a FPoW to the mining pool	Sabotaging the mining pool
FAW attack (Kwon et al., 2017)	Miners and mining pools	Not submitting a discovered FPoW until miners generate a block	Sabotaging the mining pool
Balance attack (Natoli and Gramoli, 2017)	Subgroups of similar computational power	Disrupting communications between subgroups to fork the blockchain	Disrupting the consistency of blockchain
Whale attack (Liao and Katz, 2017)	Merchants	Issuing a whale transaction that is only valid on the forked branch	Disrupting the consistency of blockchain and causing economic losses to merchants
Long-range attack (Li et al., 2017; Gaži et al., 2018; Deirmentzoglou et al., 2019)	Blockchain system	Creating a longer blockchain branch that starts with the Genesis block	Blockchain being controlled by the attackers
Nothing at stake (Li et al., 2017)	None	Mining on all branches at the same time	Disrupting the consistency of blockchain
Delay attack (Wei et al., 2018)	Nodes in the blockchain	Delaying the information communication of the nodes in the system	Thwarting the growth rate of the honest chain and forking the blockchain
Feather-forking attack (Zhang and Preneel, 2019)	Miners and blockchain system	Publicly claiming to fork the blockchain and forcing honest miners to compromise	Blockchain being controlled by the attackers
Cloning attack (Ekparinya et al., 2020)	Merchants	Convincing half of the honest sealers that the transaction has been properly committed by cloning the private key	Causing economic losses to merchants
Coinage accumulation attack (Li et al., 2020)	Miners and blockchain system	Accumulating the tokens to improve the chances of finding a block	Wasting efforts of miners and affecting the normal operation of the system

tokens for a long time (Li et al., 2020). The problem will not cause damage to the blockchain system, but it can affect the normal operation of the system and even cause the blockchain system to crash over time. The current solution is to set an upper limit on the token holding time.

We summarized the attack strategies against the main chain consensus, including the attack objectives, attack ways, and attack consequences, as shown in Table 7. It is important to note that nothing at stake does not have a specific objective, as the original intention of the attacker is to make a profit while mining.

Table 8 shows, for each primary study, the specific problems the studies proposed. It indicates that most studies (78.6%, 33/42) concentrate on the attack strategy against main chain consensus. A similar number of studies raised the other three questions. One reason is that the attack strategy against the main chain consensus covers a wide range of attacks. In addition, these attacks can bring abundant income to the attackers, which are favored by the attackers. For these reasons, consensus safety and security researchers focus on this problem. Nearly a third of the studies (31.0%, 13/42) concentrate on more than one question, indicating some correlation between these problems.

4.3. Safety and security assurance approaches

In this section, we present the state-of-the-art of blockchain consensus safety and security assurance approaches, including a description, classification, and summary of their strengths and weaknesses.

As described in Section 4.2, the current blockchain consensus protocols are mainly faced with safety and security problems. The security problem is mainly manifested in the attack of malicious nodes against the blockchain consensus protocols. In view of these two problems existing in the blockchain consensus protocol, many researchers have studied them and proposed many different approaches to guarantee the safety and security of the protocols.

4.3.1. Safety assurance approaches

The current research on blockchain consensus safety mainly focuses on the safety and liveness problems, including validity, consistency and termination problems. At present, there are three main safety assurance approaches: safety and liveness analysis, model checking and theorem proving.

(1) Analysis

Analysis includes static analysis and dynamic analysis. The current studies mainly use static analysis to study the safety and liveness of consensus protocols (Kogias et al., 2016; Aluko and Kolonin, 2021). Static analysis mainly focuses on the design, implementation and corresponding functions of consensus protocols. In essence, all blockchain consensus protocols have the nature of fault tolerance, that is, the consensus protocol can still ensure normal operation when the proportion of malicious or crash nodes is allowed. Therefore, based on the design scheme of consensus protocol, the researchers analyzed whether the consensus protocol could guarantee the normal operation

Table 8
Problems that have been proposed by primary studies.

Primary studies	Safety problems		Security problems	
	Safety	Liveness	Attack strategy against block generation node election	Attack strategy against main chain consensus
S1				✓
S2				✓
S3				✓
S4			✓	
S5				✓
S6	✓			
S7	✓	✓		✓
S8			✓	✓
S9				✓
S10			✓	✓
S11				✓
S12	✓	✓		
S13				✓
S14				✓
S15				✓
S16				✓
S17		✓		
S18				✓
S19				✓
S20				✓
S21	✓			✓
S22				✓
S23				✓
S24				✓
S25	✓	✓	✓	
S26			✓	
S27	✓	✓		
S28		✓		
S29				✓
S30				✓
S31				✓
S32			✓	✓
S33				✓
S34			✓	✓
S35	✓	✓		
S36				✓
S37				✓
S38			✓	✓
S39				✓
S40	✓	✓	✓	✓
S41				✓
S42				✓
Total	8	8	10	33

and complete realization of its functions by combining various possible behaviors of honest nodes and malicious nodes in the system under the condition of the preset fault tolerance rate.

Kogias et al. analyzed safety and liveness of Byzcoin under Byzantine faults (Kogias et al., 2016). The analysis showed that the Byzcoin consensus can tolerate at most f faulty group members among $3f + 2$ total and mitigate double-spending attacks and selfish mining. Aluko and Kolonin analyzed PoR and found that the consensus mechanism is generally safe as long as the bound on the byzantine nodes holds (Aluko and Kolonin, 2021). In addition, the consensus mechanism is also live.

Static analysis is generally performed by documents reading, algorithm analysis, code reading and understanding etc. This kind of approach is relatively simple but efficient, and does not need to spend a lot of time, material resources and other resources. Therefore, researchers often take the lead in safety and liveness analysis when the consensus protocol is just proposed.

This approach also has the disadvantage of not being persuasive. Simple analysis smacks of self-sufficiency and fails to convince other stakeholders. However, analysis is still one of the most commonly used methods today.

(2) Model Checking

Model checking is a formal verification method for finite state systems. It searches the finite-state space of the system exhaustively through algorithms and checks whether each state of the system (model) satisfies the expected properties.

The researchers first model the consensus protocol and describe the properties that the consensus protocol needs to meet. Finally, they use formal tools to automatically check the consensus protocol. If the consensus protocol does not meet the described properties during the checking, the checking will be stopped and a counterexample will be given.

Tholoniat and Gramoli modeled the state of the processes in the Byzantine fault-tolerant components of Red Belly Blockchain and the properties that the system needs to satisfy, and then verified by ByMC model checker (Tholoniat and Gramoli, 2019).

Model checking is often conducted during the consensus development and testing phases. It has the advantages of full automation and fast verification, but it also has the problem of state space explosion.

(3) Theorem Proving

Theorem proving is a technique in which a system and its properties are represented by a mathematical logic formula, which is another formal verification method. Logic is given by a formal system with axioms and inference rules. Theorem proving is essentially a process of finding characteristic proof from system axioms. The proof takes axioms or rules, and may deduce definitions and lemmas to prove that the consensus protocol has corresponding properties.

Table 9
Comparison of safety assurance approaches.

Approach	Phase	Advantage	Disadvantage
Analysis	Design	Simple and efficient	Unpersuasive
Model checking	Development and testing	Fully automated and fast	State space explosion
Theorem proving	Development and testing	No state space explosion	Difficult and requiring researchers to have a higher mathematical theory foundation

Luu et al. claimed that ELASTICO can prevent some potential threats and work normally by proving the theorems (Luu et al., 2016). Pass et al. provided formal definitions and proved that the blockchain consensus mechanism satisfies a strong forms of consistency and liveness in an asynchronous network with adversarial delays that are α-priori bounded (Pass et al., 2017). Kiayias and Panagiotakos introduced a new formal framework, which relies on trees rather than chains to analyze blockchain protocols (Kiayias and Panagiotakos, 2017). By extracting and formally describing the GHOST protocol and the Bitcoin protocols, the framework proves that the GHOST protocol implements a "robust transaction ledger" with liveness and persistence. Kiffer et al. developed a simple method based on Markov-chain to analyze consistency properties of blockchain protocols (Kiffer et al., 2018). The method defines the conditions that the consistency of blockchain needs to satisfy and proves them using theorems. Zou et al. verified that PoT can achieve validity and consistency with theorem proving (Zou et al., 2019). They also analyzed the liveness of PoT. Saltini analyzed the correctness of IBFT, which is one of a family of Proof-of-Authority blockchain consensus protocols, and proved that IBFT cannot guarantee Byzantine fault-tolerant liveness when operating on a partially synchronous network (Saltini, 2019). Yu et al. proposed a novel permissionless blockchain protocol OHIE and proved the safety and liveness properties of OHIE with the theorem (Yu et al., 2020).

Like model checking, theorem proving takes place in the development and testing phases. Theorem proving solves the problem of state space explosion, but it is difficult and often requires researchers to have a higher mathematical theory foundation.

We compare and summarize the above three approaches, mainly including their phases, advantages and disadvantages, as shown in Table 9.

4.3.2. Security assurance approaches

The main research object of the blockchain consensus security assurance approach is the blockchain consensus attacks. The purpose of these approaches is to reduce the likelihood of an attack on the blockchain consensus or reduce the damage caused by the attacks. At present, the main security assurance approaches include improvement based on consensus mechanisms, analysis, monitoring, detection, lengthening attack period, suggestions for users, and so on.

(1) Improvement based on Consensus Mechanisms

In order to better protect the consensus protocols from attacks, some researchers also put forward some improvement on the consensus protocols. These improvements are fundamentally resistant to consensus attacks, but sometimes they have other aspects of sacrifice, such as performance. These improvements mainly include the following two aspects.

Improving existing consensus protocols. In view of the shortcomings of the existing consensus protocols, it should be improved. The researchers modified the protocol's parameters, such as the delay time and the number of the block confirmation, to make the protocol more secure against corresponding attacks. The researchers modify the protocol parameters or some configuration mechanisms to make the protocol more secure and able to resist corresponding attacks.

One approach is to make a simple, backward-compatible bitcoin protocol change to raise the threshold for selfish mining. Eyal and Sirer proposed when a miner learns of competing branches of the same length, it should propagate all of them, and choose which one to mine on uniformly at random (Eyal and Sirer, 2014). Zhang and Preneel proposed to neglect blocks that are not published in time and appreciate blocks that incorporate links to competing blocks of their predecessors (Zhang and Preneel, 2017). Consequently, a block that is kept secret until a competing block is published contributes to neither or both branches, hence it confers no advantage in winning the block race.

Another approach is to use nodes to secure the consensus protocol. Wei et al. improved the anti-attack ability of blockchain by setting the honest miners (Wei et al., 2018). Yang et al. ensured the security and good operation of system by detecting and downgrading the malicious nodes timely (Yang et al., 2019). Xian et al. improved the Tendermint consensus mechanism by introducing three new types of messages and an auxiliary role of network nodes (Xian et al., 2019). Szalachowski et al. revised the Bitcoin consensus mechanism that introduces transparency and incentivizes participants to collaborate rather than to compete (Szalachowski et al., 2019). Ekparinya et al. clearly defined the quantitative relationship between Byzantine nodes, sealers, and authority nodes to protect Proof-of-Authority from cloning attacks (Ekparinya et al., 2020). Karakostas and Kiayias proposed to employ an external set of parties to finalize blocks after their creation to defend against double-spending attacks (Karakostas and Kiayias, 2021).

In addition, Li et al. improved PoS and proposed two variants (Li et al., 2017). One leverages a dedicated digital signature scheme that reveals the identity of the validator if the validator attempts to work on multiple blocks at the same height. Another protocol leverages existing pervasive Trusted Execution Environments (TEEs) to limit the block generation requests by any given validator to a maximum of one at a given height. The attacker can be accurately identified by evaluating the suspect scores of all validators (Xian et al., 2019). Bissias and Levine proposed an alternative process for PoW-based block discovery that results in an inter-block time with significantly lower variance (Bissias and Levine, 2020). This modification significantly thwarts double-spending attacks and selfish mining. Otte et al. established the validity and integrity of transactions to replace PoW, which is Sybil-resistant (Otte et al., 2020).

Designing new consensus protocols. In addition to the improvement of existing consensus protocols, researchers also proposed some new consensus protocols. These consensus protocols have higher security and anti-attack capability.

One approach is designing new consensus protocols based on the characteristics of the chain. Lewenberg et al. proposed an inclusive block chain protocol which improved the chain structure of the blockchain into a directed acyclic graph structure (Lewenberg et al., 2015). The protocol has good security against the double-spending attack. Natoli and Gramoli argued for a non-forkable blockchain design that protects against balance attacks (Natoli and Gramoli, 2017). The consensus protocol of this design makes it impossible for the blockchain to fork so that no adversary can exploit forks to influence the decision in favor of a particular branch.

Table 10
Improvement based on consensus mechanisms from 20 primary studies.

Category	Label	Description	Compatibility	Resources
Improving existing consensus protocols	P1	Miners propagating all competing branches of the same length and choosing which one to mine on uniformly at random	Backward	S2
	P2	Neglecting blocks that are not published in time and appreciate blocks that incorporate links to competing blocks of their predecessors	Backward	S18
	P3	Revealing the identity of the validator with a dedicated digital signature	Backward	S19
	P4	Limiting the block generation requests to a maximum of one at a given height with TEE	Backward	S19
	P5	Setting the honest miners	Backward	S22
	P6	Detecting and downgrading the malicious nodes timely	Backward	S26
	P7	Introducing three new types of messages and an auxiliary role of network nodes	Backward	S29
	P8	Evaluating the suspect scores of all validators	Backward	S29
	P9	Introducing transparency and incentivizing participants to collaborate rather than to compete	Backward	S31
	P10	An inter-block time with significantly lower variance	Backward	S33
	P11	Defining the quantitative relationship between Byzantine nodes, sealers, and authority nodes	Backward	S37
	P12	Establishing the validity and integrity of transactions to replace PoW	Backward	S38
	P13	Employing an external set of parties to finalize blocks after their creation to defend against double-spending attacks	Backward	S42
Designing new consensus protocols	P14	Extending Bitcoin's PoW via PoS	/	S3
	P15	Improving the chain structure of the blockchain into a directed acyclic graph structure	Backward	S5
	P16	Collective signing	Backward	S7
	P17	A non-forkable blockchain design	/	S20
	P18	Requiring a node must have the reputation score higher than a given network trust threshold	/	S32
	P19	Using the amount of coins to select miners and limiting the maximum value of the coin age	Backward	S36
	P20	Combined with VRF	/	S39
	P21	Nodes with the highest reputation values eventually become part of a consensus group that determines the state of the blockchain	/	S40
	P22	Dynamically penalizing suspected faulty behavior and suppressing Byzantine servers over time	/	S41

Several other new consensus protocols have been proposed. PoA extends Bitcoin's PoW via PoS to offer good security against possibly practical future attacks on Bitcoin (Bentov et al., 2014). ByzCoin enhances Bitcoin security and performance with strong consistency via collective signing (Kogias et al., 2016). BRBC requires a node must have the reputation score higher than a given network trust threshold before being allowed to insert a new block in the chain (de Oliveira et al., 2020). Li et al. proposed a new RPoS consensus protocol, which uses the amount of coins to select miners and limits the maximum value of the coin age to effectively avoid coin age accumulation attack and nothing at stake (Li et al., 2020). Sun et al. proposed a VDC (Sun et al., 2020). Combined with the verifiable random function (VRF), VDC realizes better fairness of user profits and time efficiency with acceptable energy consumption and without sacrificing security. Aluko and Kolonin proposed PoR where the nodes with the highest reputation values eventually become part of a consensus group that determines the state of the blockchain (Aluko and Kolonin, 2021). Zhang and Jacobsen proposed a new BFT consensus protocol called Prosecutor which leverages PoW and Raft that dynamically penalizes suspected faulty behavior and suppresses Byzantine servers over time (Zhang and Jacobsen, 2021).

We have summarized 22 security assurance approaches that improve consensus mechanisms, as shown in Table 10. The descriptions and compatibility of these approaches are described. Only two studies (S19 and S29) proposed two approaches to improve the security of consensus protocols, and the other studies only proposed one approach. In addition, it can be seen from the table that the approaches through improving consensus protocols are backward compatible. On one hand, the improved consensus protocols change or add specific mechanisms to existing protocols. The improved consensus protocol still serves the real scenario of the original consensus protocol. On the other hand, new consensus protocols are proposed to solve a problem. They do not need to consider the previous consensus protocols, but concentrate on future development and use.

(2) Analysis

In order to better analyze how to counter consensus attacks, many researchers model consensus protocols. By adjusting the system parameters in the model, we can simulate and analyze various cases of consensus protocols in different environments, and study the attack cost and profit of the attackers. Based on this, a defense strategy can also be designed. Up to now, researchers have used the following four analysis approaches to analyze blockchain consensus protocols.

Game-theoretic analysis. There are many different types of miners in the blockchain consensus process, such as malicious miners, honest miners and so on. The researchers set up the various types of miners' computational power and the information they could command, simulated the game between them, and obtained the conditions under which the various types of miners could earn profits. Finally, they gave advice on the protection of blockchain consensus protocols. Liao and Katz carried out a game-theory analysis and simulation of the whale attack, and showed

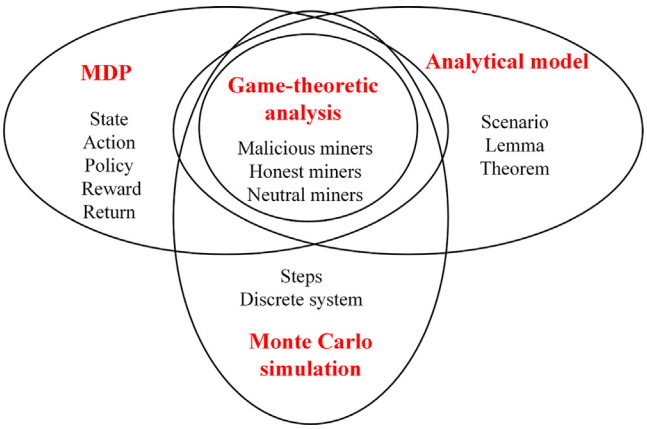

Fig. 3. Properties of analysis approaches and their relationships.

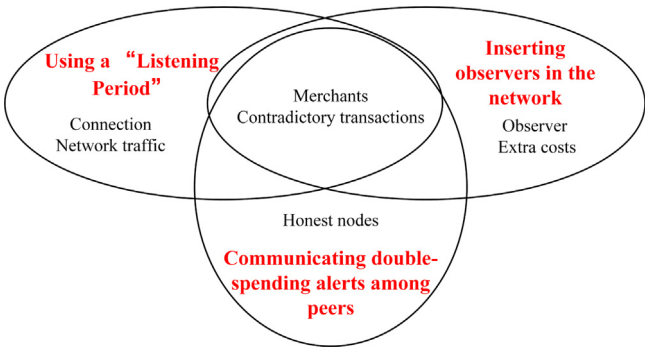

Fig. 4. Properties of monitoring approaches and their relationships.

conditions under which it yields an expected positive payoff for the attackers (Liao and Katz, 2017).

Analysis based on Markov Decision Processes (MDP). MDP is a mathematical model of sequential decision, which is used to simulate the random strategies and rewards that can be realized by agents in the environment where the system state has Markov properties. MDP contains five model elements: State, Action, Policy, Reward, and Return. The model based on MDP combines the adversary's computational power, the impact of some auxiliary attacks such as eclipse attacks, block rewards, and real-world network and consensus parameters to devise the optimal adversarial strategies against attacks. At present, the attacks that use this approach to counter mainly include the double-spending attack, selfish mining, stubborn mining, and the feather-forking attack.

Gervais et al. devised the optimal adversarial strategies for the double-spending attack and selfish mining while taking into account real-world constraints such as network propagation, different block sizes, block generation intervals, information propagation mechanisms, and the impact of eclipse attacks based on MDP (Gervais et al., 2016). Sapirshtein et al. investigated the profit threshold — the minimal fraction of resources required for the profitable selfish mining (Sapirshtein et al., 2016). Nayak et al. modeled Bitcoin mining using a Markov decision process to expand the mining strategy space to include novel stubborn strategies (Nayak et al., 2016). The results showed that selfish mining is not (in general) optimal. Zhang and Preneel introduced a multi-metric evaluation framework based on MDP to quantitatively analyze PoW protocols' chain quality and attack resistance (Zhang and Preneel, 2019). The conclusion is that no PoW protocol achieves the ideal chain quality or is resistant to selfish mining, the double-spending attack, and the feather-forking attack.

Analysis based on Monte Carlo simulation. Because in the model based on MDP, random tie breaking and uncle block inclusion could not be expressed in closed form without various assumptions sacrificing accuracy. Ritz and Zugenmaier designed a Monte Carlo simulation method (Ritz and Zugenmaier, 2018). It runs a defined number of steps within a discrete system modeling block generation events. The underlying data structure captures all relevant aspects of a blockchain and is evaluated after all blocks have been created.

Analysis based on analytical model. Chen et al. developed an abstract analytical model of PoET for theoretical analysis and assessment (Chen et al., 2017b). To assess the impact of network connectivity on the consensus security of PoW blockchain, Xiao et al. proposed an analytical model (Xiao et al., 2020). The analytical model is applied to two adversarial scenarios: (1) honest-but-potentially-colluding, (2) selfish mining. It provides a paradigm for assessing the security impact of various factors in a distributed consensus system.

We describe the relationships between these analysis approaches as shown in Fig. 3, and we show the properties of each approach in the figure. As we can see from the figure, each approach involves honest miners, malicious miners and neutral miners, since these are the most fundamental and important players in the security analysis of blockchain consensus protocols. The security of consensus protocols is meant to prevent sabotage by attackers or malicious miners. Therefore, all of these analysis approaches are actually the variants of game-theoretic analysis.

(3) Monitoring

In addition to analyzing the security of consensus protocols, monitoring consensus protocols is also a feasible security assurance approach. Based on different perspectives, Karame et al. proposed the following three monitoring approaches (Karame et al., 2012).

Using a "Listening Period". If the Bitcoin daemon receives a transaction in which the input has been spent, it generates an error locally. However, this error will not be shown to Bitcoin users. So it can take a few seconds for the merchants to monitor before providing service to the user. Merchants monitor all transactions received during that time and check for attempts to double-spending the coins. This countermeasure is based on the intuition that since each transaction takes few seconds to propagate to each node in the Bitcoin network, merchants will likely receive two contradictory transactions during the listening period (and before granting attackers the service). However, because the connectivity of Bitcoin peers varies largely with network traffic, merchants need to constantly monitor their connection counts.

Inserting observers in the network. Another possible technique to extend the proposal based on the monitoring cycle would be for a merchant to insert a node it controls within the Bitcoin network as an "observer", and the node directly forwards all transactions it receives to the merchant. In this case, if the merchant or its observer receives a questionable transaction, it can sense the double-spending attempt within seconds. However, because each observer can monitor a limited number of transactions in a short time, this approach requires a large number of observers to ensure that the attack attempt can be detected in real time, which means extra costs for merchants.

Communicating double-spending alerts among peers. Different from the above two approaches of monitoring the blockchain system by merchants, this approach is the honest nodes monitoring the system. When the honest nodes receive more than two contradictory transactions, it alerts other nodes and merchants of the possible intention of double-spending attacks.

Fig. 4 shows the properties of monitoring approaches and their relationships. From the figure, it can be found that all these three approaches protect merchants by monitoring contradictory transactions. The approach of using a "Listening Period" depends on network traffic and node connections, while the approach of inserting observers in the network incurs additional costs due to the large number of observers required. The approach of communicating double-spending alerts among peers avoids the disadvantages of the above two approaches by introducing honest nodes.

(4) Detection

Attackers always cause some anomalies when they attack the blockchain consensus protocols. By detecting these anomalies in a timely manner, the security of consensus protocols can be effectively ensured.

Address detection. In a blockchain system, each participant has an address. But some addresses are actually illegal or invalid. These addresses can not only reduce the cost of the attacks, but also improve the probability of success of the attacks. One effective way to detect addresses is to connect them. If the address cannot be connected successfully, it indicates that the address is invalid and needs to be deleted as soon as possible. Heilman et al. listed some detection approaches for addresses, including test before evicting, feeler connections, and so on (Heilman et al., 2015).

FPoW detection. It is a very difficult task to detect the attackers of blockchain consensus protocols because of the different attack objectives and attack ways. However, for some specific attacks, the attacker will leave some clues. In a FAW attack, the attacker submits some stale FPoWs to get a reward. Therefore, Kwon et al. suggested that the mining pool manager can determine whether the miner who submitted the FPoWs is the attacker by detecting whether the FPoW is stale and thus take eviction action (Kwon et al., 2017).

Chain density detection. In all PoS protocols, it is allowed that some of the parties may not be online all the time (despite the fact that they are elected to participate in the protocol), resulting in a number of "slots" that are left empty without a corresponding block. The absence of their participation is something that can be detected by observing the blockchain. Therefore, Gaži et al. proposed to detect and remove blockchains with such defects to distinguish them from the correct blockchains generated by honest parties by detecting the chain density over a certain period of time (Gaži et al., 2018).

These three approaches are used to prevent eclipse attacks, FAW attacks, and long-range attacks respectively. Therefore, although these three approaches guarantee security through detection, the objects and methods they detect are different.

(5) Lengthening Attack Period

In blockchain systems, a transaction can be completed in a short period of time as the number of nodes increases. Therefore, the attacker can usually launch an attack in a very short time to gain benefits. By lengthening the attack period, security personnel not only have enough time to detect these attacks, but also can take timely measures.

Confusing attackers. In an eclipse attack, the attacker accomplishes the eclipse by attacking the victim's tried table and new table. By increasing the table size, adding some associations between tables or some other operations, the attacker will be confused and the attack time will be prolonged (Heilman et al., 2015).

Adding extra confirmations. As described in Section 4.2.2, some merchants, in view of the increasing number of transactions, will accept 0 confirmation transactions or 1 confirmation transactions in order to ensure the smooth flow of transactions.

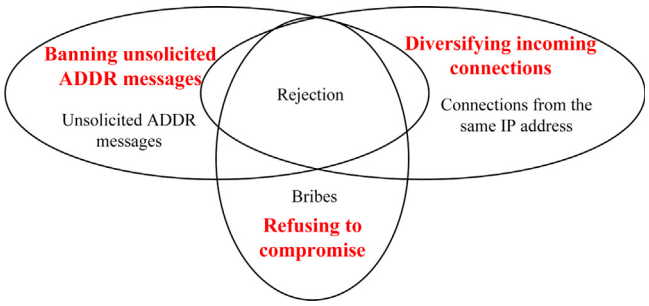

Fig. 5. Properties of suggestions and their relationships.

However, even Bitcoin, which has six confirmations as the standard, is vulnerable to attacks. Therefore, adding extra confirmations, especially for large transactions, can increase the time and cost of an attack (Bonneau, 2016).

(6) Suggestions for Users

In addition to these security assurance approaches, the researchers have some suggestions for users. While these suggestions may not be effective, they also play a role in the security of blockchain consensus protocols.

Banning unsolicited ADDR messages. ADDR messages, containing up to 1000 IP address and their timestamps, are used to obtain network information from peers. A node could choose not to accept large unsolicited ADDR messages (with > 10 addresses) from incoming peers, and only solicit ADDR messages from outgoing connections when its new table is too empty (Heilman et al., 2015). This prevents adversarial incoming connections from flooding a victim's new table with trash addresses.

Diversifying incoming connections. Today, all incoming connections to a Bitcoin node can come from the same IP address, making it easy for a single computer to monopolize a victim's incoming connections during an eclipse attack. Heilman et al. recommended that a node only accept a limited number of connections from the same IP address (Heilman et al., 2015).

Refusing to compromise. When the attackers launch an attack, sometimes in order to improve the probability of success of the attack, they may use some bribes to the miners or some means to force the miners to join them. However, as a honest miner, we should refuse to compromise with them, because such actions will ultimately destroy the blockchain system and harm the rights and interests of all (Bonneau, 2016). Every miner must refuse to compromise, even if we may lose profits.

We summarize these suggestions as shown in Fig. 5. Banning unsolicited ADDR messages and diversifying incoming connections are both security suggestions for eclipse attacks, while refusing to compromise is for whale attacks. All three are essentially rejections, but they are rejections to unsolicited ADDR messages, connections from the same IP address, and bribes from attackers.

(6) Others

Another two security assurance approaches were proposed to defend against BWH attacks and long-range attacks respectively. By exploiting the Poisson nature of PoW and the current knowledge on the propagation of information in Bitcoin, Solat and Potop-Butucaru proposed a novel time-stamped technique to prevent BWH attacks (Solat and Potop-Butucaru, 2017). This approach is based on the idea that if a selfish miner keeps a block private for more than a fixed interval, its block will be rejected by all honest miners. Gaži et al. proposed to include "context", i.e., the hash value of the most recent block, into each transaction (Gaži et al., 2018). Using context sensitivity, the validity of a transaction would require the presence of that hash

Table 11
17 assurance approaches from 14 primary studies.

Category	Security assurance approach	Label	Description	Resources
Analysis	Game-theoretic analysis	P23	Simulating the game between miners and obtaining the conditions under which the miners could earn profits	S14
	Analysis based on MDP	P24	Simulating the random strategies and rewards with State, Action, Policy, Reward, and Return	S8, S9, S10, and S30
	Analysis based on Monte Carlo simulation	P25	Capturing all relevant aspects of a blockchain and being evaluated after all blocks have been created within a discrete system modeling block generation events	S24
	Analysis based on analytical model	P26	Assessing the security impact of various factors in specific scenarios	S15 and S34
Monitoring	Using a "Listening Period"	P27	Merchants monitoring all transactions received and checking for attempts to double-spending the coins	S1
	Inserting observers in the network	P28	Inserting a node within the Bitcoin network as an "observer", and the node directly forwarding all transactions it receives to the merchant	S1
	Communicating double-spending alerts among peers	P29	Honest nodes alerting other nodes and merchants of the possible intention of double-spending attacks when receiving contradictory transactions	S1
Detection	Address detection	P30	Detecting addresses by connecting them	S4
	FPoW detection	P31	Determining whether the miner who submitted the FPoWs is the attacker by detecting whether the FPoW is stale	S13
	Chain density detection	P32	Detecting the chain density over a certain period of time	S23
Lengthening attack period	Confusing attackers	P33	Confusing attackers by increasing the table size, adding some associations between tables or some other operations	S4
	Adding extra confirmations	P34	Adding extra confirmations to increase the time and cost of an attack	S11
Suggestions for users	Banning unsolicited ADDR messages	P35	Not to accept large unsolicited ADDR messages (with > 10 addresses) from incoming peers	S4
	Diversifying incoming connections	P36	Only accepting a limited number of connections from the same IP address	S4
	Refusing to compromise	P37	Refusing to compromise with attackers or join them	S11
Others	Time-stamped technique	P38	Rejecting the block kept private for more than a fixed interval	S16
	Context sensitivity	P39	Including "context" into each transaction to allow only adversarially generated transactions to be transferred to the private blockchain	S23

in the blockchain. This would allow only adversarially generated transactions to be transferred to the private blockchain, thus fully neutralizing long-range attacks.

Table 11 summarizes 17 security assurance approaches from 14 primary studies, including their categories, descriptions, and resources.

As shown in Fig. 6, we have sorted out the attack rationale and corresponding security assurance approaches. From the figure we have the following observations:

(1) The eclipse attack, the double-spending attack, and selfish mining are the three most popular attacks. 24 of 39 security assurance approaches are used to defend against these three attacks. This also reflects the seriousness of these three kinds of attacks from the side.
(2) Although researchers have proposed 19 attacks against the blockchain consensus protocol, security assurance approaches against only 17 of these attacks have been studied. There are still two attacks (AT3 and AT5) that researchers have not studied or found a way to defend against. This may be because, on the one hand, it is difficult to implement these two attacks. Therefore, attackers are more willing to use the other attacks to gain illegal gains or damage the blockchain, leading researchers to spend more energy on the protection of other attacks. On the other hand, due to the limitations of technology and hardware equipment, researchers have not found good counter-measures against these two attacks.

4.4. Validation

In this section, we present the validation of the blockchain consensus safety and security assurance approaches.

4.4.1. Types of validation

We adopt the classification of validation methods proposed by Shaw (2003), which is widely used in software engineering to evaluate method validity. Validation methods are divided into the following six categories.

Analysis. Researchers validate the proposed methods through rigorous analysis or rigorous derivation and proof.

Experience. The approach has been applied in real-world scenarios or projects and the evidence on correctness/ usefulness/ effectiveness collected.

Evaluation. Researchers used special criteria to validate the method through well-designed experiments. It is supported by abundant experimental data. A set of examples is used to illustrate the proposed approach, with an assessment of stated criteria on the gathered information from the execution of examples.

Example. A few small-scale examples are used to illustrate the proposed approach, without any evaluation or comparison of the execution result.

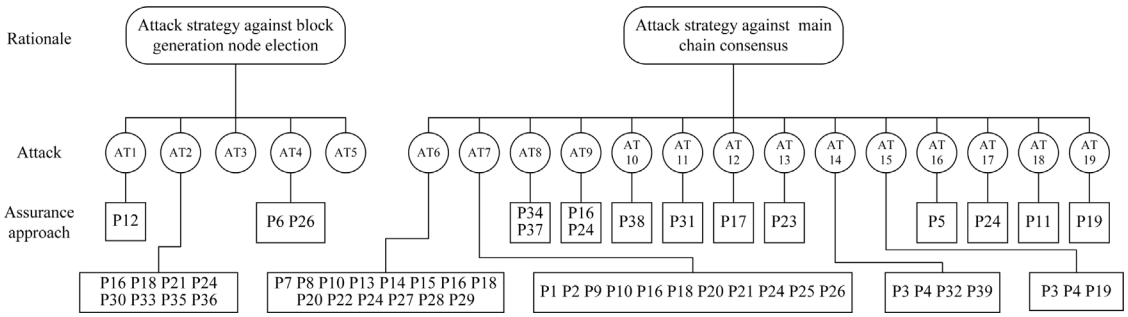

Fig. 6. Blockchain consensus protocol attack classification and security assurance approaches.

Table 12
Distribution of primary studies by validation methods.

ID	Analysis	Evaluation	Example	Persuasion	Consensus
S1		✓		✓	PoW
S2	✓	✓			PoW
S3	✓				PoA
S4	✓	✓		✓	PoW
S5	✓	✓			Inclusive block chain protocols
S6	✓				ELASTICO
S7	✓	✓			ByzCoin
S8		✓			PoW
S9		✓			PoW
S10		✓			PoW
S11				✓	PoW
S12	✓				PoW
S13	✓				PoW
S14		✓			PoW
S15	✓				PoET
S16	✓				PoW
S17	✓				GHOST
S18		✓			PoW
S19	✓				PoS
S20		✓			GHOST
S21	✓				PoW, GHOST
S22	✓	✓			PoW
S23				✓	PoS
S24		✓			GHOST
S25	✓	✓			PoT
S26		✓			DDPoS
S27	✓		✓		Red Belly
S28	✓				IBFT
S29	✓				PoW, PoS
S30		✓			PoW
S31	✓	✓			StrongChain
S32	✓	✓			BRBC
S33	✓	✓			Bobtail
S34		✓			PoW
S35	✓				OHIE
S36	✓				RPoS
S37			✓		Proof-of-Authority
S38	✓				TrustChain
S39	✓	✓			VDC
S40	✓				PoR
S41		✓			Prosecutor
S42	✓				VDC
Total	27	22	2	4	

Persuasion. An abbreviated text description is presented to convince readers that the approach is useful.

Blatant Assertion. No serious attention is paid to validation.

Table 12 shows the distribution of validation methods applied to the primary studies. We found 27 studies using analysis. 22 studies evaluated safety and security assurance approaches based on collected performance information. 2 studies worked with simplified examples to illustrate their approaches. Finally, only 4 studies used persuasion as the validation method. From the results, analysis is the most used validation method, while evaluation is also widely applied. And more than a quarter of the studies (28.6%, 12/42) used more than one method. The other two methods (experience and blatant assertion) were not used in the 42 studies.

In addition, we collate the consensus protocols verified by these studies. From the table, it can be seen that nearly half of the studies (40.5%, 17/42) are studying the safety and security of PoW consensus protocols. This is in line with our expectations as PoW is currently the most popular consensus protocol and is used by the two largest blockchain systems, Bitcoin and Ethereum. At the

same time, there are only two studies (S21, S29) that guarantee the safety and security of more than one consensus protocol.

4.4.2. Analysis

Analysis is a widely used qualitative validation method. The quality of the analysis method depends on whether the researcher's reasoning is sound and rigorous. There is no assessment of the quality of the analytical methods, and it is up to the reader to judge for himself.

In the case of S6, Luu et al. provided security analysis for how ELASTICO prevents potential threats and works securely (Luu et al., 2016). First, they clarified several assumptions about the state of the network and a definition of good randomness. They then proved the security properties of ELASTICO according to six lemmas, such as honest identities taking a dominate portion in all the generated identities, a committee (including the final committee and other committees) correctly deciding a single value, and so on. With these lemmas in hand, the correctness of the ELASTICO protocol is obvious.

4.4.3. Evaluation

Evaluation is a quantitative validation method, the most distinctive feature of which is that researchers will set some metrics in advance to measure and analyze the method. In all studies using evaluation methods, the following two types of metrics are mainly included.

(1) Blockchain Evaluation Metric

Blockchain evaluation metric mainly evaluates the quality of the blockchain itself, as well as its own anti-attack ability and so on. At present, it mainly includes the possibility of merchants being attacked by the double-spending attack, chain quality, incentive compatibility, censorship susceptibility, the expected number of blocks, profitability threshold, double-spending amount, and 50%-attack threshold.

(a) Possibility of Merchants Being Attacked by the Double-spending Attack

The possibility of merchants being attacked by the double-spending attack is mainly for PoW consensus protocol (Karame et al., 2012). For instance, some merchants accept 0 confirmed transactions for efficiency, which undoubtedly increases the possibility of being attacked by double-spending attacks.

(b) Chain Quality

Chain quality aims at expressing the number of honest-player contributions that are contained in a sufficiently long and continuous part of an honest player's chain (Zhang and Preneel, 2019). Suppose B_h and B_a represent the number of blocks mined by honest miners and attackers respectively, then we have:

$$Q = \frac{B_h}{B_h + B_a}. \tag{1}$$

Ideally, $Q = 1 - \alpha$, which α represents the computational power controlled by the attackers. We can find that the chain quality of a protocol is not related to its reward mechanism.

(c) Incentive Compatibility

Incentive compatibility measures a protocol's selfish mining resistance, which can be defined as relative revenue of the honest miners (Zhang and Preneel, 2019). Suppose $\sum R_h$ and $\sum R_a$ represent the total revenue received by the honest miners and attackers during a certain period of time respectively, then we have:

$$I = \frac{\sum R_h}{\sum R_h + \sum R_a}. \tag{2}$$

As with the chain quality metric, in the ideal case $I = 1 - \alpha$. The difference, however, is that incentive compatibility are tightly related to the reward mechanism of a protocol.

(d) Censorship Susceptibility

Censorship susceptibility measures the maximum fraction of income loss the attacker incurs on compliant miners in a censorship retaliation attack, which is inspired by the feather-forking attack (Zhang and Preneel, 2019). Suppose $\sum O_a$ represents the total reward loss due to the attack, then we have:

$$C = \frac{\sum O_a}{\sum O_a + \sum R_a}. \tag{3}$$

(e) Expected Number of Blocks

The expected number of blocks measures the minimum number of blocks required by the attack node for a successful double-spending attack (Zhang and Preneel, 2019). In order to successfully complete a double-spending attack, the attacker needs to mine a certain number of blocks in order to allow the number of blocks in his private chain containing conflicting transactions to exceed the number of blocks in the main chain, so as to cover blocks in the main chain. The larger the expected number of blocks, the more difficult it is to successfully complete the double-spending attack, and the more secure the blockchain system is.

(f) Profitability Threshold (PRTH)

Profitability threshold measures fraction of computational power controlled by the selfish mining pool when it first achieves positive relative mining gain of the node during its expansion (Zhang and Preneel, 2019; Xiao et al., 2020). The smaller the PRTH, the less computational power the attacker needs to obtain additional mining income. This means that the less difficult it is for an attacker to carry out a successful attack, and the less secure the blockchain system will be.

(g) Double-spending Amount

Double-spending amount measures the minimum transaction value that makes double-spending more profitable than honest mining (Gervais et al., 2016). Assuming that there is a transaction with a value of V, the reward that an attacker receives through a double-spend attack is Ra, and the reward for the honest miner's mining is Rh. When Ra equals Rh, the double-spending amount is V. When the value of the transaction is less than the double-spending amount, the reward for honest mining is higher. Then merchants can trade with more confidence. On the contrary, merchants need to be alert to the risk of the double-spending attack.

(h) 50%-Attack Threshold (AT50)

50%-attack threshold measures the minimum fraction of computational power controlled by adversarial nodes whose combined MR exceeds 50%, where MR represents the percentage of the revenue earned by miners in the total revenue of all miners in the network (Xiao et al., 2020). Theoretically, AT50 should exceed 50%. But sometimes when attackers get a 51% attack reward, their computational power does not exceed 50%.

(2) Attack Evaluation Metric

Attack evaluation metric starts with the attacker's behavior and the attack result, and evaluates according to the possibility of a successful attack and its benefits. At present, it mainly includes relative gain, cost of launching an attack, subversion gain, and the probability of a successful attack.

(a) Relative Gain

Relative gain is used to evaluate the revenue of the attacker under different mining strategies (Nayak et al., 2016), which can be defined as

$$relative\_gain = \frac{gain_X - gain_Y}{\alpha}, \tag{4}$$

where $gain_X$ and $gain_Y$ is the fraction of revenue earned by the attacker under strategies X and Y, and α is the computational power controlled by the attacker. In the case of Bitcoin, for

example, the attacker's honest mining earned 35% of the total revenue while occupying 35% of the computational power. But if the attacker does selfish mining, the attacker's revenue could reach 50% of the total revenue. Then the relative gain is $(0.5 - 0.35)/0.35 \approx 0.429$, which means that the revenue of selfish mining is much bigger than the revenue of honest mining. The attacker is more likely to attack.

(b) Cost of Launching An Attack

Cost of launching an attack includes time, computational power, space, and other resources (Liao and Katz, 2017). According to different attacks, the evaluation object is also different. In the case of proof-based consensus, the metric is always to evaluate computational power.

(c) Subversion Gain

Subversion gain measures the profitability of the double-spending attack, which is quantified as the time averaged illegal upper bound profit in a specific attack model (Zhang and Preneel, 2019). In a blockchain system, when the block reaches δ confirmations, with $\delta = 6$ in Bitcoin, the transaction is officially confirmed and the goods will be delivered. If the attacker orphans k blocks with a double-spending attack, he will get a reward, which can be defined as

$$R_{ds} = \begin{cases} 0, & k < \delta \\ (k + 1 - \delta)V_{ds}, & k \geq \delta, \end{cases} \tag{5}$$

where V_{ds} is the double-spending reward that the attacker can receive whenever the attacker orphans a block. Then the subversion gain of the attacker is defined as

$$S = \frac{\sum R_a + \sum R_{ds}}{t} - \alpha, \tag{6}$$

where t represents the lasting time, measured as the number of block generation intervals; α is the time-averaged mining reward without the double-spending attack.

(d) Probability of A Successful Attack

The probability of a successful attack measures the probability that an attacker will succeed in a consensus attack (Bissias and Levine, 2020). The results vary according to the consensus and attacks. For instance, when an attacker performs a double-spending attack on PoW, if the computational power is less than 50%, the success probability is close to zero. However, if the attacker uses selfish mining, the success probability is more than 50 percent as long as the computational power is above one third.

4.4.4. Example

When some researchers cannot validate a method, they can only prove it by listing a specific example. The validity of the method is proved by the existence of this example. But obviously, this method is not convincing enough. After all, examples may be special and not universal. It should be noted that one way to validate the model checking is through counter-examples. The counter-example is given to show that the model does not possess particular properties.

In the case of S27, Tholoniat and Gramoli illustrated the severity of the problem by listing six vulnerabilities of blockchain consensus including two new counter-examples (Tholoniat and Gramoli, 2019). When they made formal verification of two Byzantine fault-tolerant components of Red Belly Blockchain, they proved that the consensus protocol has safety problems according to the counter-examples found. Based on the counterexample, when the Byzantine nodes do not broadcast or deliver the correct values at certain points, the honest nodes may eventually fail to reach a consensus.

4.4.5. Persuasion

Validation purely by persuasion is rarely sufficient for a research paper. Persuasion comes entirely from the author's imagination without any theoretical basis. Sometimes the author makes some analysis, but it is too superficial, without any theory or data to support them.

In the case of S1, Karame et al. proposed three countermeasures against double-spending attacks on fast payments in Bitcoin (Karame et al., 2012). But they only evaluated the first two approaches. For the third approach, they simply analyzed its feasibility. In fact, there is no evidence that this is an effective countermeasure. They are trying to convince the reader, through some text, that this approach is feasible.

4.5. Distribution of studies

We are also interested in other related information, such as when and where the primary studies were published. This provides a direct vision of the recent research trend of blockchain consensus safety and security. We collect the publication year and source for each primary study. In addition, we categorize all primary studies into three publication types: conference paper, journal paper, and workshop paper. Fig. 7 provides a chronology for all primary studies. We put the abbreviation of conference (or journal) name as a representative of the source beside each study. For those workshop papers, A@B is used as the format of the source where A is the abbreviation of the workshop name and B is the name of the conference that holds the workshop. The collected data helps to answer research question SQ1 (How much activity about the research of blockchain consensus safety and security has there been in recent years?) and SQ2 (What is the distribution of publication venues?).

The dashed line in Fig. 7 summarizes the number of primary studies published per year from 2012 to 2021. Although the time period defined in our search strategy is from 2008 (when blockchain was proposed) to the present, no primary study was found until 2012. This may be because, on the one hand, blockchain technology emerged in 2008 but researchers were not interested in it or even did not recognize it at first; on the other hand, with the continuous development and wide application of blockchain technology, blockchain consensus safety and security assurance approaches are urgently required to improve the safety and security of blockchain system. Excluding 2013, around one to seven primary studies per year were found with a gradual upward trend from 2012 to 2017. Not until 2017 did we find a marked increase in the number of primary studies (nine). This fact indicates that more and more effort is being dedicated to this important area of research. However, there was a relatively big decrease in 2018, where only four primary studies were published in total. But then the number of primary studies showed an upward trend again. As 2021 is not yet over, only three studies have been found so far. Consequently, more data is needed in the next few years to determine whether the research in this area has reached a peak.

The table in Fig. 7 summarizes the percentage of primary studies according to the venue of publication. Among 30 primary studies, 73.81% (31 of 42) were published in conferences. More than one paper was published in six of the conferences (CCS, FC, USENIX Security, SSS, S&P, and NDSS). The remaining 13 studies were published in different conferences. Nineteen percent (8 of 42) were published in journals. Finally, 7.14% (3 of 42) were published in workshops.

From this collected data, we offer three observations:

(1) The number of papers on blockchain consensus safety and security is growing but still relatively small. However, given this specialized research area and a relatively short

Publication Type	Percentage
Conference Paper	73.81%
Journal Paper	19.05%
Workshop Paper	7.14%

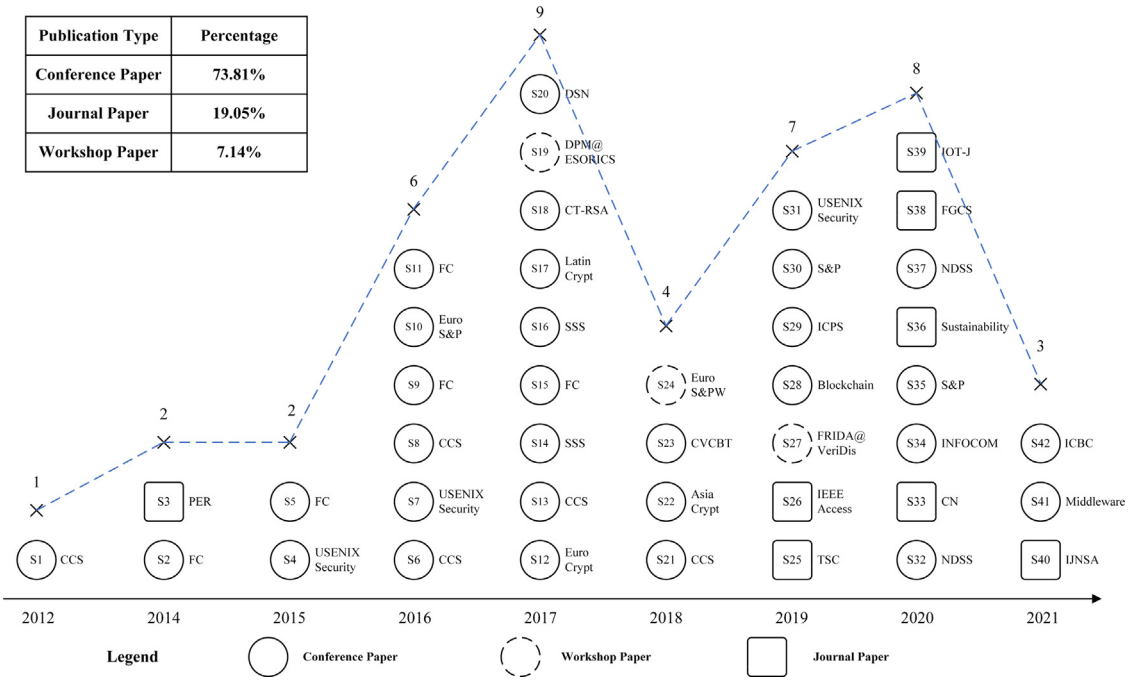

Fig. 7. Publication chronology.

history, the number of primary studies can be considered significant. In addition, as there are still many challenges to be addressed, more research work may be in progress or published in the future.

(2) Most papers were published in proceedings (34, including conference and workshops). This might be because many interesting methods have been proposed in recent years, but their theoretical frameworks are still not solid.

(3) Two primary studies (S25 and S33) appeared in reputable journals (TSC and CN). However, no papers were published in top journals in the software engineering or network security area (e.g., TSE and TDSC). Furthermore, 10 papers were published in top conferences (CCS, USENIX Security, and S&P). The lack of publications in top journals is probably due to the lack of elaborate evaluation of proposed approaches, and especially due to the small size of case studies.

5. Discussion

5.1. Deficiencies of current study

This section presents the deficiencies of current study on the blockchain consensus safety and security.

5.1.1. Lack of analysis for multiple attack strategies

Most of the current studies focus on only one attack strategy, which is far from sufficient. Some attacks can cause considerable damage at minimal cost through combination. For example, by combining stubborn mining with an eclipse attack, attackers can obtain more revenue than normal mining when they command more than 9% of computational power of the network. Therefore, analyzing a single attack strategy will ignore some of the potential security risks. But which attack strategies to combine analysis and how to analyze, this is a problem that needs to be explored.

5.1.2. Lack of guidance for simulation design

Due to the mechanism of the blockchain system itself, researchers cannot conduct experimental analysis in the actual system. In the simulation experiment, the researcher evaluates the security of the system in a specific model by modifying the parameters of the blockchain system, so to find out the potential security risks of the system or look for strategies to counter the attack. However, researchers usually only consider a few specific experimental parameters in the model, such as the computational power of attackers, block reward, network delay, and so on. In fact, the actual blockchain system is far more complex than the simulated environment. Only considering the changes of a few parameters cannot guarantee the security of this consensus protocol in the actual blockchain system.

Although simulations may have limitations compared with actual blockchain systems, they are indispensable in the field of consensus security research. Researchers can change parameters at will in a simulated environment, or even depart from real-world parameters, to find more meaningful results.

5.1.3. Lack of research on the comparison between approaches

As discussed in Section 4.3, there are many approaches or techniques to guarantee consensus safety and security, many researchers have also proposed some strategies to prevent attacks. But there is no comparison between these approaches or strategies, and no research has examined the strengths and weaknesses of each approach. A systematic approach evaluation system will help researchers to study consensus safety and security more specifically, and also help designers of blockchain systems to design more secure consensus protocols.

5.1.4. Lack of verification or evaluation tools on consensus safety and security

As discussed in Section 4.3, researchers have proposed many approaches and models to verify or evaluate the blockchain consensus safety and security. However, there are very few verification or evaluation tools on consensus safety and security for blockchain systems. Currently, researchers and designers of blockchain systems can only implement algorithms to verify and

evaluate the security of consensus protocols. There is, therefore, a desire and need to develop and make available more consensus security verification or evaluation tools to support both research and actual development.

5.2. Challenges of blockchain consensus safety and security

5.2.1. Security situational awareness for potential threats

Situational awareness is the perception of environmental elements and events with respect to time or space, the comprehension of their meaning, and the projection of their future status, which was first proposed by the US Air Force in the 1980s. In the blockchain system, once a transaction has occurred, there is no way to undo it. Especially on the value Internet, which relies on public chain technology, there is no way to withdraw a transaction once it has taken place. Therefore, prevention in blockchain security is far more important than in traditional information security. However, due to the decentralization and anonymity of blockchain and the cryptography, distributed storage, and other technologies involved, it is an extremely difficult task to realize the security situational awareness of blockchain consensus protocols. Current technology can only detect some suspicious addresses or transactions, but cannot identify potential attackers, let alone perceive the attack strategy or attack purpose.

5.2.2. Defending at the first sign of attacks

Due to the complexity and diversity of consensus attacks, the existing security assurance approaches or technologies are disastrously behind attacks. Some of the monitoring and testing approaches described in Section 4.3.2 can only serve as warnings and precautions. They are unable to defend their victims at the first sign of attack. If there was an approach to defend against an attacker in the first place, it would not only protect, but also minimize the damage to the victim.

5.2.3. Tracing and locating attackers

Although blockchain is an open and transparent data structure, due to its unique anonymity, blockchain users can only be located at the address, but cannot be further identified to a real person. In addition, the decentralization of blockchain also means that there is no authoritative central control, but can only rely on the system itself for protection. This allows attackers to carry out malicious attacks with impunity. If an attacker can be accurately traced and located, it can not only deter attackers and improve the security of blockchain, but also recover the losses of victims. However, this clearly contradicts the nature of blockchain.

5.3. Related work

Gramoli explored the use of the Ethereum blockchain protocol in the context of a private chain where the set of participants is controlled (Gramoli, 2016). This study only discussed the malicious behaviors and double-spending attacks in the private chain, but the other dangers were not described, and the relevant security assurance approaches were not systematically introduced. Deirmentzoglou et al. provided a systematic literature review on long-range attacks for PoS protocols, and introduced 7 countermeasures for the attack (Deirmentzoglou et al., 2019). Then, Gramoli went on to discuss mainstream blockchain consensus algorithms and how to revisit the classical Byzantine consensus in the context of blockchain, and warned about the dangers of using these blockchains without accurately understanding the assurances provided by their consensus algorithms, mainly including attacks on Bitcoin and Ethereum consensus protocols (Gramoli, 2020). This paper covered some aspects of blockchain consensus, but did not systematically and comprehensively analyze the security assurance technology of consensus.

There are some surveys on the security of blockchain systems or practical applications such as Bitcoin. Conti et al. proposed a systematic survey that covered the security and privacy aspects of Bitcoin (Conti et al., 2018). They reviewed existing vulnerabilities in Bitcoin and its key underlying technologies such as blockchain and PoW-based consensus protocols, while examining the feasibility and robustness of state-of-the-art security solutions. Zhang and Zhou discussed the basic architecture of blockchain and its potential security and trust issues at the data, networking, consensus, smart contract, and application layers (Zhang and Zhou, 2020). Homoliak et al. proposed a blockchain security reference architecture (SRA), which adopted a stacked model (similar to ISO/OSI) to describe the nature and hierarchy of various security and privacy aspects (Homoliak et al., 2020). All of these studies covered but were not limited to the security of blockchain consensus, so their research on blockchain consensus was not systematic and comprehensive.

Compared with the above review literature, the advantage of this paper lies in the comprehensive analysis of blockchain consensus safety and security. We start from the two aspects of consensus safety and consensus security, and systematically analyze stakeholders, safety and security problems, assurance approaches and approach validation. Finally, we make statistics of the common research in the relevant research papers.

5.4. Limitation of this review

There are two potential limitations to this approach: (1) the scope of retrieval is limited, so there are inevitably omissions; (2) the data extraction process is difficult to ensure 100% accuracy and completeness.

Firstly, in order to ensure the fairness of the topic selection process, this paper designs a retrieval strategy to search for relevant literatures through keywords. However, the selection of keywords is subjective, which may lead to the omission of some relevant studies. In addition, a comprehensive and significant set of electronic databases with computer science at its core was selected as the search scope to cover as much relevant research as possible. However, due to the short development time and relatively new technology of blockchain consensus protocols, compared with academic research papers, technical websites related to blockchain consensus protocol may have stronger timeliness. Therefore, this paper may omit some relevant studies that should have been included in this paper.

Secondly, there may be some inaccuracy in the extraction results. In the pilot of the data extraction process, we extracted the details of each preliminary study using a predefined extraction form. However, we often find that in some studies, the data extraction process is incomplete due to the lack of sufficient information. More specifically, we often find that the technology or method is not adequately described, validation method issues are not always addressed, the analysis of the experimental results is not well explained, and the sample and study settings are not clearly presented. This affects the accuracy of the extraction results.

6. Future work

After comprehensive analysis of existing studies on blockchain safety and security, this paper summarizes the following future research directions.

6.1. Comprehensive study of security and performance

Currently, some blockchain systems, such as Litecoin and Dogecoin, reduce the block generation time to 2.5 min and 1 min respectively in order to reduce transaction latency. But it also greatly reduces the security of the blockchain consensus protocol. Security and performance have been two key topics of concern for blockchain consensus protocols. Failure to consider security for the performance can result in incalculable losses, while blindly improving blockchain consensus security without considering its performance is not in line with the long-term development of blockchain. Unfortunately, there are relatively few studies that consider both security and performance of blockchain consensus protocols. Therefore, how to improve the security of blockchain consensus on the basis of ensuring its performance is a problem worthy of consideration and research.

6.2. Organic integration of priori and posterior approaches

In order to reduce the safety and security problems of blockchain consensus protocols, future research should first focus on the priori-based safety and security assurance approach. Considering different functional requirements and application scenarios, we should first focus on how to design and develop secure consensus protocols. The safety of consensus protocol is guaranteed and the anti-attack capability of consensus protocols is improved by the priori approach.

However, in the decentralized blockchain system, it is not sound to only apply the priori approach to guarantee the safety and security of blockchain consensus, and the posterior-based safety and security assurance approach needs to be strengthened. The approach focuses on comprehensive security detection through analysis and testing, real-time monitoring and analysis of the deployed consensus protocol.

By organically integrating the priori and posterior approaches can the safety and security of consensus protocols be guaranteed at all stages of its life cycle.

6.3. Organic integration of qualitative and quantitative approaches

In the process of consensus design and development, different possible risk factors are reasonably analyzed through qualitative analysis, and consensus safety and security problems are effectively classified and graded, to clarify the overall security situation and existing problems of the current blockchain consensus. However, qualitative analysis approaches are usually combined with personnel experience, and the accuracy of the analysis basis may be affected by subjective factors.

The quantitative evaluation approach is objective and provides accurate data analysis. In the process of consensus testing and maintenance, the safety and security problems in blockchain consensus are detected and discovered in time by quantitative analysis.

Combining qualitative analysis and quantitative analysis, the two approaches are organically integrated. According to the characteristics of each stage in the life cycle of consensus protocols, qualitative analysis method is used to analyze the overall safety and security of blockchain consensus. Then, quantitative analysis methods are used to collect data for in-depth analysis, and cost-effective counter-measures are selected according to accurate analysis results.

7. Conclusion

In this paper, we have presented a survey covering 42 papers on blockchain consensus safety and security since blockchain technology was first proposed. In addition to discussing different stakeholders involved in the safety and security assurance of blockchain consensus protocols, we analyzed the safety and security problems in the proposed blockchain consensus, and sorted a list of 19 attacks against the blockchain consensus. We also classified the approaches used to guarantee the blockchain consensus safety and security into different categories, and investigated the validation of the blockchain consensus safety and security assurance approaches. Furthermore, we traced the distribution of blockchain consensus safety and security papers. Finally, we enumerated some current consensus safety and security research deficiencies and challenges, and discussed future research directions. We believe that this paper has a comprehensive reference value for the research of consensus safety and security, as well as a certain guiding significance.

CRediT authorship contribution statement

Qihao Bao: Conceptualization, Methodology, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Bixin Li:** Methodology, Writing – review & editing, Supervision, Funding acquisition. **Tianyuan Hu:** Writing – review & editing. **Xueyong Sun:** Investigation, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFE0105500, in part by the Research Council of Norway under Grant 309494, and in part by the Key Research and Development Program of Jiangsu Province, China under Grant BE2021002-3.

References

Abd-El-Malek, M., Ganger, G.R., Goodson, G.R., Reiter, M.K., Wylie, J.J., 2005. Fault-scalable Byzantine fault-tolerant services. Oper. Syst. Rev. 39 (5), 59–74.

Al-Jaroodi, J., Mohamed, N., 2019. Industrial applications of blockchain. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, pp. 0550–0555.

Alpern, B., Schneider, F.B., 1985. Defining liveness. Inform. Process. Lett. 21 (4), 181–185.

Alpern, B., Schneider, F.B., 1987. Recognizing safety and liveness. Distrib. Comput. 2 (3), 117–126.

Aluko, O., Kolonin, A., 2021. Proof-of-reputation: An alternative consensus mechanism for blockchain systems. Int. J. Netw. Secur. Appl. (IJNSA) 13.

Bentov, I., Lee, C., Mizrahi, A., Rosenfeld, M., 2014. Proof of activity: Extending bitcoin's proof of work via proof of stake. ACM SIGMETRICS Perform. Eval. Rev. 42 (3), 34–37.

Biely, M., Schmid, U., Weiss, B., 2011. Synchronous consensus under hybrid process and link failures. Theoret. Comput. Sci. 412 (40), 5602–5630.

Bissias, G., Levine, B.N., 2020. Bobtail: Improved blockchain security with low-variance mining. In: Network and Distributed System Security Symposium, San Diego, USA.

Bonneau, J., 2016. Why buy when you can rent? In: International Conference on Financial Cryptography and Data Security, Christ Church, Barbados. Springer, pp. 19–26.
Buchman, E., 2016. Tendermint: Byzantine Fault Tolerance in the Age of Blockchains (Ph.D. thesis). University of Guelph.
Buntinx, J.-P., 2017. What is Proof of Elapsed Time? . https://themerkle.com/what-is-proof-of-elapsed-time/.
Cachin, C., Kursawe, K., Petzold, F., Shoup, V., 2001. Secure and efficient asynchronous broadcast protocols. In: Annual International Cryptology Conference, Santa Barbara, USA. Springer, pp. 524–541.
Castro, M., Liskov, B., 2002. Practical Byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst. (TOCS) 20 (4), 398–461.
Castro, M., Liskov, B., et al., 1999. Practical byzantine fault tolerance. In: Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, USA, Vol. 99. pp. 173–186.
Chen, P.-W., Jiang, B.-S., Wang, C.-H., 2017a. Blockchain-based payment collection supervision system using pervasive Bitcoin digital wallet. In: 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE Computer Society, pp. 139–146.
Chen, L., Xu, L., Shah, N., Gao, Z., Lu, Y., Shi, W., 2017b. On security analysis of proof-of-elapsed-time (poet). In: International Symposium on Stabilization, Safety, and Security of Distributed Systems, Boston, USA. Springer, pp. 282–297.
Conti, M., Kumar, E.S., Lal, C., Ruj, S., 2018. A survey on security and privacy issues of bitcoin. IEEE Commun. Surv. Tutor. 20 (4), 3416–3452.
de Oliveira, M.T., Reis, L.H., Medeiros, D.S., Carrano, R.C., Olabarriaga, S.D., Mattos, D.M., 2020. Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications. Comput. Netw. 179, 107367.
Deirmentzoglou, E., Papakyriakopoulos, G., Patsakis, C., 2019. A survey on long-range attacks for proof of stake protocols. IEEE Access 7, 28712–28725.
Dib, O., Brousmiche, K.-L., Durand, A., Thea, E., Hamida, E.B., 2018. Consortium blockchains: Overview, applications and challenges. Int. J. Adv. Telecommun. 11 (1&2), 51–64.
DPoS, 2022. Delegated proof of stake . https://www.geeksforgeeks.org/delegated-proof-of-stake/.
Dwork, C., Naor, M., 1992. Pricing via processing or combatting junk mail. In: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology. Santa Barbara, USA. Springer, pp. 139–147.
Ekparinya, P., Gramoli, V., Jourjon, G., 2020. The attack of the clones against proof-of-authority. In: Network and Distributed System Security Symposium, San Diego, USA.
Eyal, I., Sirer, E.G., 2014. Majority is not enough: Bitcoin mining is vulnerable. In: International Conference on Financial Cryptography and Data Security, Santa Barbara, USA. Springer, pp. 436–454.
Fanning, K., Centers, D.P., 2016. Blockchain and its coming impact on financial services. J. Corp. Account. Financ. 27 (5), 53–57.
Fischer, M.J., Lynch, N.A., Paterson, M.S., 1985. Impossibility of distributed consensus with one faulty process. J. ACM 32 (2), 374–382.
Gaži, P., Kiayias, A., Russell, A., 2018. Stake-bleeding attacks on proof-of-stake blockchains. In: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), Zug, Switzerland. IEEE, pp. 85–92.
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: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria. pp. 3–16.
Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N., 2017. Algorand: Scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th Symposium on Operating Systems Principles. pp. 51–68.
Gramoli, V., 2016. On the danger of private blockchains. In: Workshop on Distributed Cryptocurrencies and Consensus Ledgers (DCCL'16), Chicago, USA.
Gramoli, V., 2020. From blockchain consensus back to Byzantine consensus. Future Gener. Comput. Syst. 107, 760–769.
Heilman, E., Kendler, A., Zohar, A., Goldberg, S., 2015. Eclipse attacks on bitcoin's peer-to-peer network. In: 24th USENIX Security Symposium (USENIX Security 15), Washington, D.C., USA. pp. 129–144.
Homoliak, I., Venugopalan, S., Reijsbergen, D., Hum, Q., Schumi, R., Szalachowski, P., 2020. The security reference architecture for blockchains: Toward a standardized model for studying vulnerabilities, threats, and defenses. IEEE Commun. Surv. Tutor. 23 (1), 341–390.
Jakobsson, M., Juels, A., 1999. Proofs of work and bread pudding protocols. In: Secure Information Networks. Springer, pp. 258–272.
Karakostas, D., Kiayias, A., 2021. Securing proof-of-work ledgers via checkpointing. In: 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). IEEE, pp. 1–5.
Karame, G.O., Androulaki, E., Capkun, S., 2012. Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin. In: Conference on Computer & Communication Security, Raleigh, USA. pp. 1–17.
Keele, S., 2007. Guidelines for Performing Systematic Literature Reviews in Software Engineering. tech. rep, Citeseer.
Kiayias, A., Panagiotakos, G., 2017. On trees, chains and fast transactions in the blockchain. In: International Conference on Cryptology and Information Security in Latin America, Havana, Cuba. Springer, pp. 327–351.
Kiffer, L., Rajaraman, R., Shelat, A., 2018. A better method to analyze blockchain consistency. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, Canada. pp. 729–744.
Kim, H., Laskowski, M., 2017. A perspective on blockchain smart contracts: Reducing uncertainty and complexity in value exchange. In: 2017 26th International Conference on Computer Communication and Networks (ICCCN). IEEE, pp. 1–6.
Kogias, E.K., Jovanovic, P., Gailly, N., Khoffi, I., Gasser, L., Ford, B., 2016. Enhancing bitcoin security and performance with strong consistency via collective signing. In: 25th USENIX Security Symposium (USENIX Security 16), Austin, USA. pp. 279–296.
Kwon, Y., Kim, D., Son, Y., Vasserman, E., Kim, Y., 2017. Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, USA. pp. 195–209.
Lamport, L., 1977. Proving the correctness of multiprocess programs. IEEE Trans. Softw. Eng. (2), 125–143.
Lamport, L., 1998. The part-time parliament. ACM Trans. Comput. Syst. 16 (2), 133–169.
Lamport, L., Robert, S., Pease, M., 1982. The Byzantine generals problem. ACM Trans. Program. Lang. Syst. 4 (3), 382–401.
Lewenberg, Y., Sompolinsky, Y., Zohar, A., 2015. Inclusive block chain protocols. In: International Conference on Financial Cryptography and Data Security, San Juan, Puerto Rico. Springer, pp. 528–547.
Li, W., Andreina, S., Bohli, J.-M., Karame, G., 2017. Securing proof-of-stake blockchain protocols. In: European Symposium on Research in Computer Security International Workshop on Data Privacy Management, Cryptocurrencies and Blockchain Technology, Oslo, Norway. Springer, pp. 297–315.
Li, A., Wei, X., He, Z., 2020. Robust proof of stake: A new consensus protocol for sustainable blockchain systems. Sustainability 12 (7), 2824.
Liao, K., Katz, J., 2017. Incentivizing blockchain forks via whale transactions. In: International Conference on Financial Cryptography and Data Security, Sliema, Malta. Springer, pp. 264–279.
Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., Saxena, P., 2016. A secure sharding protocol for open blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria. pp. 17–30.
Miller, A., Xia, Y., Croman, K., Shi, E., Song, D., 2016. The honey badger of BFT protocols. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria. pp. 31–42.
Nakamoto, S., 2008. Bitcoin: A peer-to-peer electronic cash system. Decentralized Bus. Rev. 21260.
Natoli, C., Gramoli, V., 2017. The balance attack or why forkable blockchains are ill-suited for consortium. In: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Denver, USA. IEEE, pp. 579–590.
Nayak, K., Kumar, S., Miller, A., Shi, E., 2016. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P), SaarbrÜCken, Germany,. IEEE, pp. 305–320.
Newsome, J., Shi, E., Song, D., Perrig, A., 2004. The sybil attack in sensor networks: analysis & defenses. In: Third International Symposium on Information Processing in Sensor Networks, Berkeley, USA. IEEE, pp. 259–268.
NoneAge, 2019. NoneAge . https://www.noneage.com/.
Ongaro, D., Ousterhout, J., 2014. In search of an understandable consensus algorithm. In: 2014 USENIX Annual Technical Conference (USENIX ATC 14). pp. 305–319.
Otte, P., de Vos, M., Pouwelse, J., 2020. TrustChain: A Sybil-resistant scalable blockchain. Future Gener. Comput. Syst. 107, 770–780.
Pass, R., Seeman, L., Shelat, A., 2017. Analysis of the blockchain protocol in asynchronous networks. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France. Springer, pp. 643–673.
Pease, M., Shostak, R., Lamport, L., 1980. Reaching agreement in the presence of faults. J. ACM 27 (2), 228–234.
Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. Inf. Softw. Technol. 64, 1–18.
Politou, E., Casino, F., Alepis, E., Patsakis, C., 2021. Blockchain mutability: Challenges and proposed solutions. IEEE Trans. Emerg. Top. Comput. 9 (4), 1972–1986.
PoS, 2022. Proof of stake . https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/.
Rathod, N., Motwani, D., 2018. Security threats on Blockchain and its countermeasures. Int. Res. J. Eng. Technol. 5 (11), 1636–1642.
Ren, L., 2014. Proof of Stake Velocity: Building the Social Currency of the Digital Age. Self-Published White Paper.

Ritz, F., Zugenmaier, A., 2018. The impact of uncle rewards on selfish mining in ethereum. In: 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), London, UK. IEEE, pp. 50–57.
Saltini, R., 2019. IBFT liveness analysis. In: 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, USA. IEEE, pp. 245–252.
Sapirshtein, A., Sompolinsky, Y., Zohar, A., 2016. Optimal selfish mining strategies in bitcoin. In: International Conference on Financial Cryptography and Data Security, Christ Church, Barbados. Springer, pp. 515–532.
Sathya, A., Banik, B.G., 2020. A comprehensive study of blockchain services: future of cryptography. Int. J. Adv. Comput. Sci. Appl.(IJACSA) 11 (10), 279–288.
Sayeed, S., Marco-Gisbert, H., 2019. Assessing blockchain consensus and security mechanisms against the 51% attack. Appl. Sci. 9 (9), 1788.
Shaw, M., 2003. Writing good software engineering research papers. In: Proceedings of the 25th International Conference on Software Engineering, Portland, USA. IEEE, pp. 726–736.
Shrier, D., Wu, W., Pentland, A., 2016. Blockchain & infrastructure (identity, data security). Mass. Inst. Technol.-Connection Sci. 1 (3), 1–19.
Slowmist, 2022. Hack Events . https://hacked.slowmist.io/.
Solat, S., Potop-Butucaru, M., 2017. Brief announcement: Zeroblock: Timestamp-free prevention of block-withholding attack in bitcoin. In: International Symposium on Stabilization, Safety, and Security of Distributed Systems, Boston, USA. Springer, pp. 356–360.
Sompolinsky, Y., Zohar, A., 2015. Secure high-rate transaction processing in bitcoin. In: International Conference on Financial Cryptography and Data Security. Springer, pp. 507–527.
Sun, G., Dai, M., Sun, J., Yu, H., 2020. Voting-based decentralized consensus design for improving the efficiency and security of consortium blockchain. IEEE Internet Things J. 8 (8), 6257–6272.
Sun, J., Yan, J., Zhang, K.Z., 2016. Blockchain-based sharing services: What blockchain technology can contribute to smart cities. Financ. Innov. 2 (1), 1–9.
Szalachowski, P., Reijsbergen, D., Homoliak, I., Sun, S., 2019. Strongchain: Transparent and collaborative proof-of-work consensus. In: 28th USENIX Security Symposium (USENIX Security 19), Santa Clara, USA. pp. 819–836.
Tholoniat, P., Gramoli, V., 2019. Formal verification of blockchain Byzantine fault tolerance. In: The 6th Workshop on Formal Reasoning in Distributed Algorithms (FRIDA'19), Budapest, Hungary.
Wei, P., Yuan, Q., Zheng, Y., 2018. Security of the blockchain against long delay attack. In: International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, Australia. Springer, pp. 250–275.
Xia, Q., Dou, W., Guo, K., Liang, G., Zuo, C., Zhang, F., 2021. Survey on blockchain consensus protocol. J. Softw. 32 (2), 277–299.
Xian, X., Zhou, Y., Guo, Y., Yang, Z., Liu, W., 2019. Improved consensus mechanisms against censorship attacks. In: 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), Taiwan, China. IEEE, pp. 718–723.
Xiao, Y., Zhang, N., Lou, W., Hou, Y.T., 2020. Modeling the impact of network connectivity on consensus security of proof-of-work blockchain. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, pp. 1648–1657.
Yang, F., Zhou, W., Wu, Q., Long, R., Xiong, N.N., Zhou, M., 2019. Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism. IEEE Access 7, 118541–118555.
Yu, H., Nikolić, I., Hou, R., Saxena, P., 2020. Ohie: Blockchain scaling made simple. In: 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, USA. IEEE, pp. 90–105.
Yuan, Y., Ni, X.-C., Zeng, S., Wang, F., 2018. Blockchain consensus algorithms: the state of the art and future trends. Acta Automat. Sinica 44 (11), 2011–2022.
Zhang, G., Jacobsen, H.-A., 2021. Prosecutor: an efficient BFT consensus algorithm with behavior-aware penalization against Byzantine attacks. In: Proceedings of the 22nd International Middleware Conference, QuÉbec City, Canada. pp. 52–63.
Zhang, R., Preneel, B., 2017. Publish or perish: A backward-compatible defense against selfish mining in bitcoin. In: Cryptographers' Track At the RSA Conference, San Francisco, USA. Springer, pp. 277–292.
Zhang, R., Preneel, B., 2019. Lay down the common metrics: Evaluating proof-of-work consensus protocols' security. In: 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, USA. IEEE, pp. 175–192.
Zhang, J., Tian, R., Cao, Y., Yuan, X., Yu, Z., Yan, X., Zhang, X., 2021. A hybrid model for central bank digital currency based on blockchain. IEEE Access 9, 53589–53601.
Zhang, P., Zhou, M., 2020. Security and trust in blockchains: Architecture, key technologies, and open issues. IEEE Trans. Comput. Soc. Syst. 7 (3), 790–801.
Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H., 2017. An overview of blockchain technology: Architecture, consensus, and future trends. In: 2017 IEEE International Congress on Big Data (BigData Congress). IEEE, pp. 557–564.
Zheng, W., Zheng, Z., Chen, X., Dai, K., Li, P., Chen, R., 2019. Nutbaas: A blockchain-as-a-service platform. IEEE Access 7, 134422–134433.
Zou, J., Ye, B., Qu, L., Wang, Y., Orgun, M.A., Li, L., 2019. A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services. IEEE Trans. Serv. Comput. 12 (3), 429–445.

Qihao Bao is currently working toward the Ph.D. degree at the Southeast University, Nanjing, China. His research interests include software engineering, software testing, and blockchain security.

Bixin Li received the Ph.D. degree in computer science from Nanjing University, in 2001. He is currently a full Professor of School of Computer Science and Engineering at the Southeast University, Nanjing, China. His research interests include: Program slicing and its application; Software evolution and maintenance; and Software modeling, analysis, testing and verification.

Tianyuan Hu is currently working toward the Ph.D. degree at the Southeast University, Nanjing, China. Her research interests include software engineering, blockchain security.

Xueyong Sun is currently working toward the Master degree at the Southeast University, Nanjing, China. His research interests include software engineering, blockchain security.