Computers & Security 95 (2020) 101871

Contents lists available at ScienceDirect

Computers & Security

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

Proof of X-repute blockchain consensus protocol for IoT systems

Eric Ke Wang[a], RuiPei Sun[a], Chien-Ming Chen[b], Zuodong Liang[c], Saru Kumari[d], Muhammad Khurram Khan[e,*]

[a] Harbin Institute of Technology, Shenzhen, China
[b] Shandong University of Science and Technology, Shandong, China
[c] Blockchain Group of Tencent Research Institute, Shenzhen, China
[d] Department of Mathematics, Ch. Charan Singh University, Meerut, India
[e] Centre of excellence in Information Assurance, King Saud University, Riyadh 11653, Saudi Arabia

ARTICLE INFO

Article history:
Received 25 February 2020
Revised 24 April 2020
Accepted 8 May 2020
Available online 12 May 2020

Keywords:
Blockchain
Consensus protocols
IoT
Trust management
Repute incentive

ABSTRACT

Recently, blockchain technology has been used to address the security issues of Internet of things (IoT) applications. However, some issues particular to blockchain should be solved to meet the security requirements of IoT systems. The core of blockchain technology is distributed computing, along with the collaboration mechanism of group trust under the distributed computing environment, which can solve the scalability, collaboration ability, trust relationship, and security protection challenges faced by the IoT. The existing blockchain consensus protocol can only solve the trust cooperation problem with limited credibility. Although distributed trust relationship management has advantages over centralized trust relationship management, several risks exist. Here, the most critical issue is the credibility of the blockchain consensus protocol. Based on the characteristics of the traditional public chain, we propose a repute-based consensus protocol for blockchain-enabled IoT systems. In the protocol, two methods are designed to enable the blockchain system to reach a consensus rapidly and safely. The repute rewards and punishments method settles the repute values of nodes; nodes with satisfactory behavior receive repute rewards. The repute mining method proposes repute requirements for consensus nodes; nodes with high repute value produce blocks more easily. Security analysis has been conducted using a theory model and experimental evaluation. The establishment of our repute system can improve the consensus protocol, resisting stronger attacks, and giving several users with lower computing power a greater opportunity to participate in consensus. The experimental results show that the repute-based consensus protocol has advantages in terms of security and that its resistance capability against attacks is improved.

1. Introduction

Recently, there has been a trend in Internet of things (IoT) research to combine edge computing with IoT to avoid the bottleneck problem of cloud servers. Edge computing is a distributed computing technology, which is a method of data processing near the IoT device terminal. In most cases, the device does not need to connect to the cloud platform, and the intelligent control of IoT devices can be realized through local data computing. Edge computing transfers data processing from the cloud center to the edge of the network. Computing and data storage can be distributed to the edge of the Internet near IoT terminals, sensors, and users. This can not only relieve the pressure of cloud bandwidth, computing, etc. but can also optimize a network service architecture oriented to perception driven. For example, the air conditioner, water heater, refrigerator, and security camera in a home can be coordinated through edge computing, which can ensure the best energy saving and service status, even when the cloud server is not connected. However, the distributed multi-terminal collaborative data service mode of edge computing will lead to new security problems. The original centralized defense mechanism of cloud computing cannot adapt to the new edge computing applications. The blockchain provides the possibility to establish a trust mechanism for the edge devices of the IoT. Blockchain solutions can eliminate security threats in the IoT with edge computing. Because blockchain technology can ensure that once data are recorded and stored, they cannot be changed or tampered with. This technology provides enterprises with the opportunity to access the activity log, suspicious logins, or attempts to access the data on the record. It can play

* Corresponding author.
E-mail addresses: wk_hit@hit.edu.cn (E.K. Wang), chienmingchen@ieee.org (C.-M. Chen), 164386284@qq.com (Z. Liang), mkhurram@ksu.edu.sa (M. Khurram Khan).

https://doi.org/10.1016/j.cose.2020.101871
0167-4048/

various roles, including secure data storage, digital identity verification, and fraud prevention.

Blockchain is a trust machine (Yang et al., 2018) that can solve the scalability, collaboration ability, trust relationship, and security protection challenges faced by industrial applications. It has the function of a cross subject, improving efficiency and reducing cost, and enables people in different regions to conduct large-scale collaborative work efficiently. Blockchain is a technical solution that maintains a data ledger in a decentralized way (Su et al., 2018). A data ledger is an orderly block chain storage structure. Blockchain uses cryptography-related technology to achieve its non-tampering characteristics. As a key component of the blockchain system, the consensus algorithm ensures the consistency of data ledgers among nodes and Byzantine fault tolerance (BFT). However, existing blockchain consensus protocols can only solve the trust cooperation problem with limited credibility. Although distributed trust relationship management has advantages over centralized management, several risks exist. Here, the most critical issue is the credibility of the blockchain consensus protocol (Kang et al., 2019).

Based on our formal work (Wang et al., 2020),this study improves the existing security of the current mainstream consensus algorithm and promotes the consensus nodes in the blockchain network to maintain a stable trusted state by building a distributed repute layer. Using the distributed, decentralized repute system built on the blockchain to show the trustworthiness of each consensus node can improve the security, reliability, and efficiency of the consensus mechanism.

Distributed repute can be defined as the credit rating of a member by others in a Peer to Peer(P2P) network. The distributed repute system can be perfectly combined with the blockchain. The repute can be a part of the consensus of the blockchain, and the blockchain can guarantee the security and consistency of the distributed reputation system.

The organizational structure of this paper is as follows. The second section briefly summarizes some early related work, including analysis of the blockchain consensus algorithm and of the distributed repute mechanism. The third section defines the PoX consensus model and its related assumptions and analyzes the limitations and improvements of the current consensus. In the fourth section, we propose a new consensus module, the repute layer, and discuss its design and analysis in detail. The fifth section covers the implementation of the repute layer and simulation experiments. Experimental results are provided and analyzed. Finally, the sixth section summarizes the current work and discusses the future work.

2. Related work

The consensus algorithm is the core mechanism of the blockchain system. It can be divided into two types: the first is based on the chain consensus algorithm, and the second is based on the Byzantine consensus algorithm. The chain-based consensus algorithm is a traditional blockchain consensus algorithm, represented by proof of work (PoW) and proof of stake (PoS). This algorithm first determines the branch selection algorithm and then selects a branch as the final blockchain among the multiple branches of the blockchain. The Byzantine consensus algorithm was improved based on the Paxos and Raft protocols (Liu et al., 2018a). The traditional consistency algorithm is used mainly in the field of distributed storage and does not have the characteristic of BFT. Therefore, on this basis, BFT was introduced, represented by practical BFT (pBFT) (Gramoli, 2017). This algorithm needs to reach a clear consensus in each round of operation and cannot be changed.

Recently, several new work shows that the consensus protocols should be improved for security problems. For example, compared to Bitcoin, Bitcoin-NG can increase transaction block frequency and transaction throughput while ensuring protocol security and fairness. Its micro-block idea, which separates the transaction block from the process of block maker election, embodies the modular idea of protocol design. However, there are some problems in Bitcoin-NG's incentive mechanism, and the allocation proportion of the transaction fee can be optimized. Compared to Bitcoin, an enemy's attack on Bitcoin-NG, such as selfishly mining, has higher profits (Yin et al., 2018).

Zohar (2017) focused on the study of the incentive mechanism of a blockchain, analyzed the importance of incentive mechanisms in blockchain systems based on workload proof, and found that unreasonable incentives can make the entire system unsafe.

Byzcoin uses collective signing (Cosi) (Drijvers et al., 2019) to replace MAC and adopts the Merkel tree form for each node signature's arrangement of messages. It reaches a consensus on each round of micro-blocks, including the transactions that occurred in the current time, to reach a consensus. However, the security of Cosi's signature scheme has not been proved. Gilad et al. (2019) ensured that the transaction status information obtained when the new node is added is true and effective and identified false information generated by adversaries.

The consensus protocol 'proof of elapsed' time is based on the waiting time for the hardware chip to execute a command. In particular, it uses trusted hardware to generate random numbers to determine the next block producer. Hyperledger uses the enclave module in Intel SGX, a trusted Intel chip, and participants send a random waiting that time needs to be obtained from the enclave before block layout. The node with the shortest waiting time is selected as the leader node. Zhang et al. (2017) proposed resource-efficient mining, which also uses trusted hardware to calculate the PoW with the shortest time. This point will be selected as the leader node. The consensus mechanisms mentioned above have different emphases on performance, security, and decentralization. The network is often divided into multiple areas by the mixed consensus mechanism of multiple committees. Each area runs parallel committees to deal with transactions separately. Typical schemes include OmniLedger (Kokoris-Kogias et al., 2018), and RapidChain (Zamani et al., 2018). However, the possible security problems are the efficient processing of cross chip transactions and the offset of an adversary's reconfiguration process.

Moreover, in the delegated PoS consensus algorithm, the nodes with ownership vote out a fixed number of accounting nodes. In this way, the system throughput is significantly improved, but the nodes in the network can create unpredictable situations. That is, if the accounting nodes engage in malicious behavior or the ordinary nodes refuse to vote, the system cannot find and take appropriate measures in time, resulting in system security problems. Some blockchain systems have been improved by using repute mechanisms (Huang et al., 2019; Liu et al., 2019; 2018b; Wang et al., 2019).

In the current mainstream blockchain consensus algorithm, the security problem is particularly prominent:

(1) The algorithm is not perfect, and bad behavior cannot be found immediately. As a part of the public blockchain consensus algorithm, a token incentive mechanism is used to motivate consensus nodes to operate the consensus protocol correctly and simultaneously maintain the security and stability of the underlying blockchain system. Generally, as a tool of motivation, tokens play a crucial in motivation. However, their defects are relatively noticeable: the incentive conditions need to be strictly constrained, and finding the bad behavior rapidly is a difficult task. The standard of punishment based on token incentives is too high; the common minor evils will not be punished, the system often cannot

find and remedy problems in time, and the security problem is particularly prominent.

(2) The threshold for honest nodes to participate in a consensus is remarkably high. In the public chain consensus algorithm, a costly consensus participation method is used. For example, PoW uses computer computing power as a means of consensus participation, and the higher the computing power, the easier to generate blocks. PoS uses equity as a means of consensus participation, and the higher the equity, the easier to generate blocks. This will make it difficult for ordinary honest users to participate in the consensus; hence, only a few giants will be able to participate, forming a monopoly and significantly reducing the security of the blockchain system.

(3) The consensus game is remarkably simple. The consensus algorithm can be regarded as a single game every time. Each game is based on the current state, not combined with historical behavior. For example, every time PoS competes for a block right, it only refers to the current number of rights and interests. Even if a consensus node has often behaved badly before, it is still unaffected in the current round of the game and can participate in the consensus process normally. This game mechanism will make the evil nodes have no worries, significantly increase the probability of doing evil, and lead to the fragility of the blockchain consensus system.

Based on the above problems, this paper combines the distributed reputation system with the current blockchain consensus protocol and constructs a reputation layer to promote the consensus nodes in the blockchain network to maintain a stable trusted state to ensure that the blockchain system can reach a consensus rapidly and safely.

3. Model definition

We model the PoX consensus protocol and combine the features of the IoT to propose an improvement scheme.

3.1. Proof-of-X consensus model

Participation of Entire Network Assumption: the blockchain with PoX consensus is a public chain; that is, all nodes in the blockchain network have the capability to participate in the consensus. *Attribute Proof* Every participant has an attribute called a proof method. The participants use attributes to join the competition for the right to generate a block. For example, the proof method of PoW is the workload (computing power) of participants: participants must use a given amount of computing power to obtain block accounting rights. In contrast, PoS uses stakes (different versions of PoS have their own definitions of a stake) as proof: all participants must hold certain stakes to obtain block accounting rights. *Mining Competition* When the proof method is determined, a node can participate in the consensus for competing block accounting rights. Thus, competition rules that describe how to engage in the consensus process must be defined. At present, the most popular competition mode is a mining-like mode: that is, calculating a value less than a certain threshold. PoW and PoS, for example, both use this competition mode for the right to produce a block. The higher the proof method is, the less difficult the calculation is, and the easier it is to win the competition. *Selection of Fork* Once the competition mode of computation power (called mining) is adopted, a delay of message delivery in the P2P network happens. Nodes must calculate the effective value at almost the same time to obtain a block accounting right. That is, in the blockchain, several blocks of a certain height will appear simultaneously, which is called a fork. Therefore, fork selection rules must be stipulated to ensure the nodes' data are consistent. The mining competition mode has natural fault-tolerant resistance and can maintain a 50% fault-tolerant rate (under ideal conditions). *No Accountability* When an honest node receives a new block that contains an illegal transaction, it can only discard the block: there is no effective accountability for block producers. This is because the honest node verifies the block locally and does not know whether other nodes have also received the illegal block. Thus, an honest node cannot identify the attitude of other honest nodes to the block producers. To be able to hold producers accountable, the honest node would need to reach a consensus throughout the network on punishing the block producer, which is a very difficult job. This lack of accountability means that punishments cannot be executed.

3.2. Assumptions

Network We assume the network is weakly synchronous, which describes the average behavior of public protocols over the Internet. Weak synchronization means that to provide security, we do not need any synchronization assumptions, and to provide activity, we need to specify the network delay, which bounds the message delivery time for live honest nodes. If a node's message is passed after this delay, we consider that node to be malicious. There are existing distributed consensus protocols with these requirements.

Attacks Although we assume that nodes communicate through a reliable authenticated point-to-point channel, the network can still be damaged by selfish behavior, malicious attacks, and even unintentional misconfiguration. In this research, we studied both a static and a dynamic scenario. In the first scenario, malicious nodes have been determined before the protocol runs; thus, no honest node can become malicious. In the dynamic scenario, nodes can switch from malicious to honest (cleaned) or honest to malicious, with new nodes entering the system at a certain rate. We also study two attack scenarios: the 51% attack scenario in which a node can become malicious if it has a large amount of the proof method and the Sybil attack scenario in which malicious nodes can create many new accounts to join the consensus. We consider these two attacks to be the primary threats to our protocol.

4. Repute module

For the convenience of explanation and universality, we make the following agreements:

- Node: A node is a common node in a blockchain.
- Miner: The nodes participating in the consensus of the blockchain represent the miners.
- Account: There are accounts in the blockchain, and one node can create multiple accounts. When a node participates in the consensus, an account needs to be set up, and the account then participates in the consensus.

4.1. Design criteria and problems

We propose a repute module that can be added to the PoX consensus algorithm. The following are the conditions that the consensus needs to satisfy:

- The PoX consensus model definitions in Section 4.1 and
- The assumptions in Section 4.2.

When we discuss a node with repute participating in the consensus, we will inevitably come across these difficult problems:

- A node can create multiple accounts to participate in the consensus, and each account maintains a repute value. This approach can improve the probability of generating blocks with-

out increasing the cost, which critically compromises the fairness of the system.
- When the repute value of an account is remarkably low, the user has sufficient reason to abandon the account before being punished without paying any penalty. As long as all the funds in the account are transferred to other accounts before the account is abandoned, it can rejoin with a new identity and no penalty.
- Users can gradually build up their repute and exit with a fraud. This has happened before on the bourse and Bitcoin lending platforms. Therefore, the repute mechanism must be carefully considered, and the economic costs of Byzantine behavior must be strictly analyzed.

For the first problem, which is often called a Sybil attack, an effective solution is to use an easy-to-calculate function at first but difficult to calculate subsequently or to use an artificial action that cannot be represented by a machine. Currently, PoW and PoS are common solutions. It is worth noting that our repute layer is built on the existing PoX consensus, and the existing consensus has the means to deal with such attacks. In theory, we do not need to take targeted defensive measures in the repute layer. Nevertheless, our design includes a solution to solve this problem in the repute layer, which can strengthen the new consensus protocol.

For the second and third problems, a common solution is capital collateral. If a node wants to participate in the consensus, it must take out a certain amount of funds as collateral in advance. When a node is punished, it means that the collateral has been confiscated. This method has two problems: first, a certain amount of collateral money will shut out uncivilized users, remarkably weakening the enthusiasm to become a miner; second, when a malicious node's malicious behavior leads to a far higher payout than the collateral, the malicious behavior will not be deterred.

This paper proposes an identity-based solution that can solve the above problems reasonably and effectively. This is described in Section 4.2.

4.2. Detailed design

Building Repute First, the repute value must be bounded. In this study, we denote the minimum value of the repute value by $R_{\min}$, the maximum value by $R_{\max}$, and the default initial value of registered nodes by R_{init}.

Second, the repute module includes a start-up process, proof-of-credit+X, a fork selection rule, and an incentive mechanism, which will be described in detail in the following sections.

Proof-of-X-Repute We call our method proof-of-X-repute or PoXR. The essence of blockchain consistency is to maintain the same ledger status on different nodes. At each round of consensus, a qualified node is selected to update the blockchain ledger. The question is which node to choose. In blockchains with reputable modules, the principle of consistency is to select users who contribute positively to the system based on abundant resources. Therefore, we propose a consensus algorithm for PoXR that uses the repute of nodes in the system to reduce the difficulty of achieving PoX consensus. This avoids the need for ASIC mining, machine risk, and the risk of centralization.

The formal expression of PoXR is

$$H(block, nonce) < f(r_u^m, D) \tag{1}$$

$$f(r_u^m, D) = D - D \cdot \gamma \cdot \frac{(r_u^m - R_{init})}{R_{init}}. \tag{2}$$

The blocker selection method of p does not need to be set. All nodes registered in the system can participate in consensus and compete for blockers.

Assuming that node M is the originator, the production block method used for reputation certification is shown in formulas (3) and (4), and the specific implementation is shown in Algorithm 1.

Algorithm 1: Production block method P.

Input: Miner id, competition period T, block height to be generated$H$, initial reputation R_{init},
Available reputation decay factorβ, reputation difficulty conversion factorγ, mining difficulty of block to be generated D
Output: New block $block$

```
r = GetReputation(id)
count = GetCount(H, T, id)
r_u = r/(β^count)
D_u = D − D ∗ γ ∗ (r_u − R_init)/R_init
block = CreateBlock(id, D_u)
return block
```

$$H(block, nonce) < f(r_u^m, D) \tag{3}$$

H	Hash function of mining calculation
$block$	Newly generated blocks
$nonce$	Random number required for mining calculation.

$$f(r_u^m, D) = D - D \cdot \gamma \cdot \frac{(r_u^m - R_{init})}{R_{init}} \tag{4}$$

R_{init}	Initial reputes
γ	Repute proof factors.

The output of the f function is the threshold value for the competition mode of the PoX consensus, which is determined by the current effective reputation value of M and the difficulty of the current PoX consensus. Once the generated hash value is lower than the current target value, it is considered to have an accounting right. In cases where d is fixed, the larger the reputation value used as proof, the greater the output value of threshold value f. This means that under the same original conditions, the probability of obtaining the next block right increases with an increase in reputation.

Hence, nodes will pay more attention to the reputation value of their own identity and will not easily engage in malicious behavior. This also means that reputation value will change a one-time game into a repeating game. The design criterion of the F function is to integrate the effective credit value r_u of the current miner's account into the difficulty of the PoX consensus of d.

There is no need to set the method for selecting the verifier of the reputation certification method V. All nodes in the system can participate in the consensus as the verifier.

Assuming that the validation node is n and the new block received is created by node M, the specific implementation of the validation block method used to certify reputation is shown in Algorithm 2.

There are two design methods for fork selection method F used in the repute proof. The first is to use the F method of the original consensus algorithm without the intervention of the reputation module.

The second is to incorporate the reputation module into the parameter characteristics of the original consensus algorithm. This implementation refers to the first scheme, Algorithm 3, and implements the longest chain fork selection method:

Incentive mechanism with repute module

Algorithm 2: Validation block method V.

Input: Latest block *block*, competition period T, current block's height H, initial repute R_{init}, available repute decay factorβ, repute difficulty conversion factorγ, mining difficulty of new block D
Output: Verification results *result*

if $H <=$ *block.Height* **then**
  return *false*
end if
$r = GetReputation(block.miner\_id)$
$count = GetCount(H, T, id)$
$r_u = r/(\beta^{count})$
$D_u = D - D * \gamma * (r_u - R_{init})/R_{init}$
if *block.hash* $> f(D_u)$ **then**
  return *false*
end if
if $!verify(block.Signature)$ **then**
  return *false*
end if
if $!verify(block.transactionList.Signatures)$ **then**
  return *false*
end if
if $!verify(block.transactionList)$ **then**
  return *false*
end if
Output: *true*

Algorithm 3: Fork selection method F.

Input: Fork chain $L1$, fork chain $L2$
Output: Confirmation chain

if $L1.length < L2.length$ **then**
  return $L2$
else
  return $L1$
end if

As shown in Algorithm 4. The rewards are calculated in the following method. In the case of a reward, the reward function for miner n is

$$R_n = R_n + \alpha \cdot \left(1 - \frac{Count}{T_{cmp}}\right) \cdot (R_{max} - R_n). \tag{5}$$

The idea is that if a node generates many blocks in a period of time, then the repute reward of the node will decrease linearly. The repute value of node n is r_n. When the repute value is close to the maximum, its increase becomes more difficult.

In the case of a penalty, the penalty function for miner n is

$$R_n = R_n - \theta \cdot e^{\delta t} \cdot (R_n - R_{min}). \tag{6}$$

The idea behind this adaptive parameter is that if a node can generate blocks in a decay period, then the node will not be punished for credits. If there is no block in several successive decay periods, the repute decreases exponentially. Again, r_n is the repute value of node n. When the repute value is close to its maximum value, the punishment is stronger.

In the case of attenuation, the attenuation function of node n is

$$R_n = R_n - \theta \cdot e^{\delta t} \cdot (R_n - R_{min}). \tag{7}$$

θ, δ	Repute decay factor
$R_{\min}$	Repute value lower limit
t	The miner has passed decay period t since its last block T_{dec}

Algorithm 4: Incentive method I.

Input: Latest block *block*, competition period T_{cmp}, penalty period T_{dec}, maximum reputation $R_{\max}$, minimum reputation $R_{\min}$, reputation reward factorα, reputation penalty factorθ, δ, miner register list *MinerRegisterList* miner decay map *MinerDecayMap*
Output: Reputation status R

$r = GetReputation(block.miner\_id)$
$count = GetCount(block.Height - 1, T_{cmp}, block.miner\_id)$
$rr = \alpha * (1 - count/T_{cmp}) * (R_{\max} - r)$
$R = SetReputation(block.miner\_id, r + rr)$
if *block.Height* % $T_{dec} == 0$ **then**
  for each *miner* in *MinerRegisterList* **do**
    $r = GetReputation(miner)$
    $count = GetCount(block.Height, T_{dec}, miner)$
    if $count == 0$ **then**
      $t = 1$
      $MinerDecayMap.delete(miner)$
    else
      $t = MinerDecayMap.get(miner) + 1$
      $MinerDecayMap.set(miner, t)$
    end if
    $d = \theta * e^{\delta * t} * (r - R_{\min})$
    $R = SetReputation(block.miner\_id, r - d)$
  end for
end if
Output: R

The inclusion of the adaptive parameter means that if a node can generate blocks in a decay period, then the node will not be punished for reputation. If there has been no block in several successive decay periods, the reputation decreases exponentially. r_n is the reputation value of node n. When the reputation value is close to its maximum value, the punishment is stronger.

5. Security analysis

We propose a loadable module rather than a perfect consensus protocol, so the efficiency and decentralization of the original consensus protocol have not changed remarkably. The main purpose of introducing the reputation module is to quantify the quality of the nodes, according to their behavior, to purify the system environment and largely solve the security-related problems of the current blockchain.

5.1. Security model for PoX consensus protocols

A security analysis of the PoX consensus is given in this section. According to the characteristics of mining competition in a PoX consensus, malicious attacks generally include double-cost attacks.

A double-spending attack is the inherent attack method of PoX consensuses (Karame et al., 2012). When a transaction is packed into a block and the block has been connected to the blockchain, the attacker forges a block after the parent block. The forged block does not contain the transaction in an attempt to cause a bifurcation. When the forged chain finally wins, the deal is erased. It is difficult to achieve this attack when the computing power is less than 50%. When an attacker has more than 51% of the computing power, it will succeed in launching a double-flower transaction. However, some attack strategies lead to lower security than

expected. For example, in a paper published in 2016 by Sompolinsky Gervais et al. (2016), an attack strategy comparing special currency was proposed: combining selfish mining (Eyal and Sirer, 2014; Nayak et al., 2016) and double costs can significantly reduce the attack cost for attackers. With this combination, even if the computing power is less than 50%, they may attempt and succeed with a double-flower attack, reducing the threshold of the attack.

Recently, many scholars have conducted security modeling and demonstrations of the PoW consensus (Pass et al., 2017; Zhang and Preneel, 2019). Based on the characteristics of the PoX consensus model, this study considers three aspects of measurement:

(1) Attack effectiveness. This metric measures the difficulty of block production for an attacker. Let $\mathbb{A}$ denote the largest attacker's exploitable boundary. Suppose that an attacker's X-attribute proportion is α, the attack strategy is s, and the attack time is t. Within time t, the number of blocks produced by the attacker is B_a, and the number of blocks produced by the honest nodes is B_h. The definition of $\mathbb{A}$ is then given by formula (8).

$$\mathbb{A}_s(\alpha) = \max\left\{\lim_t \frac{B_a}{B_h + B_a}\right\}. \tag{8}$$

Ideally $\mathbb{A}_s(\alpha) = \alpha$. That is, attackers can produce blocks according to their X-attribute proportion.

(2) Excitation. This metric measures whether the incentive mechanism of the PoX algorithm is fair. Let $\mathbb{E}$ denote the maximum benefit to the attacker. Suppose that the proportion of the X attribute mastered by an attacker is α, the attack strategy is s, and the attack time is t. In time t, the total benefit obtained by the attacker is R_a, and the total benefit obtained by honest nodes is R_h. The definition of $\mathbb{E}$ is shown in formula (9).

$$\mathbb{E}_s(\alpha) = \max\left\{\lim_t \frac{R_a}{R_h + R_a}\right\}. \tag{9}$$

Ideally, $\mathbb{E}_s(\alpha) = \alpha$. That is, the attacker can obtain profits according to their proportion of the X attribute.

(3) Split attack benefit. This metric measures the benefit of an attacker's successful execution of a double-flower attack. Suppose that all honest nodes contain a transaction in which the attacker pays money to the merchant, and the corresponding Shuanghua transaction is packed into its own attack chain by the attacker. When the payment transaction is confirmed by δ blocks, the merchant delivers the goods to the attacker. If the payment transaction then becomes invalid after the attack, the attacker receives the value of the payment. Let v_d be the amount of double spending per block. If the number of blocks produced by the attacker is k, the definition of the double-spending reward is shown in formula (10).

$$R_d(k, \delta, V_d) = \begin{cases} 0 & k < \delta \\ (k + 1 - \delta)V_d & k \geq \delta \end{cases} \tag{10}$$

In this formula, $k + 1 - \delta$ is the number of blocks on the attack chain that can be confirmed. The attacker will not be punished for the failure of a Shuanghua attack because even if the attack fails, the goods or services are still sent to the attacker by the merchant. Because the attacker continues to mine on the local attack chain, when the length of the attack chain exceeds that of the honest chain and the number of blocks has not reached δ, the attacker can choose to publish the attack chain, surrendering the double-flower attack, but obtaining the block reward. Let r_a be a block reward on the attack chain. Adding this parameter to the module can balance the risk of attack failure with the block reward of normal mining. $\mathbb{S}$ is the bifurcated attack revenue. Suppose that an attacker's X-attribute proportion is α, that the block reward per unit time is R_α under normal mining, that the attack strategy is s, and that the attack time is t. The definition of $\mathbb{S}$ is shown in formula (11).

$$\mathbb{S}_s(\alpha, \delta, V_d) = \max\left\{\lim_t \frac{R_a + R_d}{t} - R_\alpha\right\}. \tag{11}$$

An attacker is usually not willing to bear the risk of losing the block reward, $\mathbb{S}_s(\alpha, \delta, V_d) = 0$. Nevertheless, when v_d is sufficiently large, the attacker will violate this principle and conduct the bifurcating attack, attempting to implement the double-flower transaction.

5.2. PoXR security analysis

In the PoXR algorithm, given by formulas (3) and (4), the effective reputation value of a miner affects the block difficulty of that miner, that is, their block rate. Furthermore, their effective repute value is determined by the number of blocks the miner has during a period of time. The repute value of an attacker is defined as r; the proportion of the X-attribute mastered by the attacker is α, and the competition period is t. The impact of other nodes on the attacker is not considered. Ideally, the block probability is proportional to α and inversely proportional to block difficulty d. That is, $P_a = \frac{\alpha}{D}$.

The block difficulty of block i in T is $D_i = D \cdot (1 - \gamma \cdot \frac{(\frac{r}{\beta^i} - R_{init})}{R_{init}})$. For the convenience of proof, if D is used as a fixed variable, the effect of the reputation value on D can be transferred to α, which has no effect on the block probability. We define the proportion of valid X attributes as $\alpha_e = \alpha \cdot \frac{D}{D_i}$. This means that under the support of the node reputation value, the proportion of X attributes that are actually available to the node and the block probability are as shown in formula (12).

$$\mathbb{A}_s(\alpha) = P_a = \alpha_e = \alpha \cdot \left(\frac{R_{init}}{R_{init} - \gamma \cdot (\frac{r}{\beta^i} - R_{init})}\right). \tag{12}$$

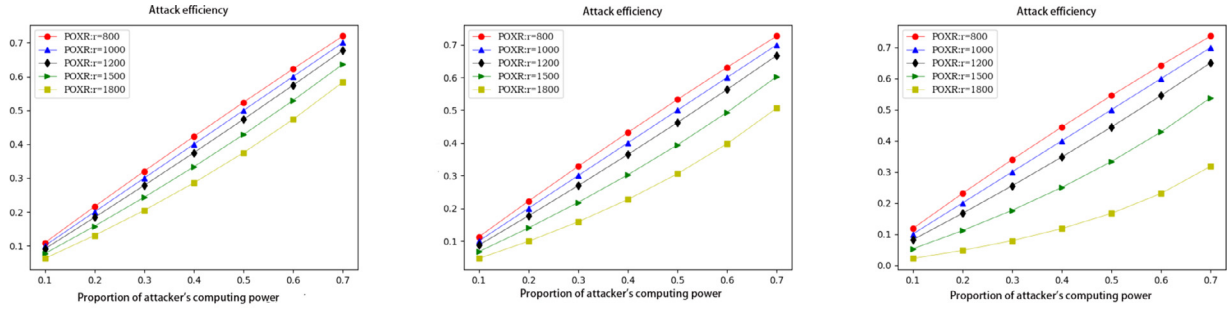
Considering the influence of an honest node on the attacker, suppose that the actual block probability of the attacker is $P_a^{'}$, the effective X-attribute proportion of the honest node is α_e^j, and the set of honest nodes is N_h. The block probability of the attacker is then shown in formula (13).

$$\mathbb{A}_s(\alpha) = P_a^{'} = \frac{\alpha_e}{\alpha_e + \sum_{j \in N_h} \alpha_e^j} \tag{13}$$

According to formula (13), when an attacker with the X-attribute proportion α produces the first block, assuming that the attacker's repute value is the initial repute value of $R_{init} = 1000$ and the repute value of the honest node is r, the attacker's attack efficiency $\mathbb{A}_s(\alpha)$ is related to α, r, as shown in Fig. 1. In the figure, as r increases, the attacker's attack performance is reduced and the security improves. Thus, the greater the value of γ, the greater the impact of repute value on the consensus.

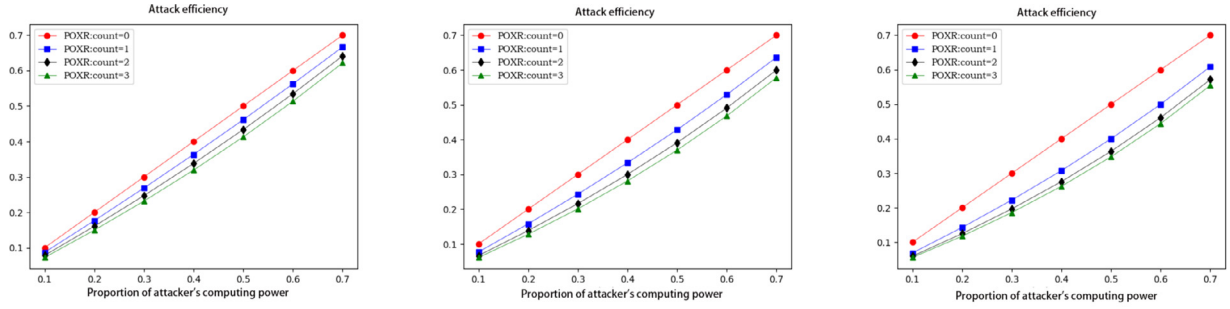
Assuming that the repute values of both the honest node and the attacker are the initial repute value of $R_{init} = 1000$, when an attacker with the X-attribute proportion of α produces *count* blocks in a competitive cycle, the relationship between attack efficiency $\mathbb{A}_s(\alpha)$ and α, *count* is shown in Fig. 2. As evident in the figure, with the increase of *count*, the attacker's attack efficiency decreases, and the attacker's probability of continuous block output also decreases, increasing the security. Thus, the larger the value of β, the greater the attenuation range of the node's effective repute value, the lower the probability of continuous block output, and the higher the security.

Compared to the original protocol, our algorithm based on the repute module mainly improves the attack performance dimension: an attacker with proportion α has a block probability lower



(a) γ=0.5 (b) γ=0.7 (c) γ=1.0

Fig. 1. γ: Relationship between the efficiency of different attacks and the proportion of computing power.



(a) β=1.2 (b) β=1.5 (c) β=2.0

Fig. 2. β: Relationship between the efficiency of different attacks and the proportion of computing power.

than α. When the attacker is producing blocks, there are three situations:

(1) $P_a^{'} = P_a,\ r = r_a$. The implication of this situation is that when the repute value of the attacker is r_a, the block-out rate of the attacker in this round will not be affected by the other nodes. At this point, the block-out rate of the attacker can be expressed by formula (12), indicating that in a competition cycle, the block-out probability of the attacker $P_a^{'}$ gradually decreases as the number of blocks out increases, which can remarkably enhance the attack. This feature is also applicable to (2) and (3).

(2) $P_a^{'} < P_a,\ r = r_a^{'}\ r_a^{'} < r_a$. In this case, the attacker is evidently affected by the high repute value of the honest node, and its probability of block-out is reduced.

(3) $P_a^{'} > P_a,\ r = r_a^{''}$ and at this point $r_a^{'} > r_a$. In this situation, when the attacker finishes a certain number of blocks, $\frac{r}{\beta^i}$ decreases, leading to situation (1) or (2).

In summary, our algorithm can ensure that the attack difficulty increases, and the probability of block out decreases when the attacker continuously blocks out, effectively resisting the attacker's double-flower attack.

6. Experiments and evaluation

Hypotheses and environment To demonstrate the applicability of the repute module in the IoT, we used the current mainstream blockchain project Ethereum to develop and apply the repute module. We ran multiple miner nodes on a single machine, on which each node independently operated the Ethereum protocol to mine and ran the protocol on Docker. To conduct a simple test of basic data, miner registration contracts were not used here; we manually set the miner's address and repute value in the Genesis Block and maintained the static list of miner addresses in the Ethereum agreement. The server had a quadcore Core i5-3470 CPU and 8 GB of memory, and its OS was Linux Mint 18.2. For the convenience of observation, all parameters were set to their ideal values, given in Table 1.

Table 1
Experimental parameters.

$R_{\max}$	$R_{\min}$	R_{init}	T_{cmp}	T_{dec}	β	γ	α	θ	δ
2000	0	1000	3	20	1.2	1	0.01	0.01	0.2

Simulation experiment Normal repute growth includes the repute reward decrease and repute penalty from many blocks, and repute growth performance differs according to the module parameter settings. We set up five miners with respective mining calculation capacities of three times, two times, one time, one time, and one time and respective initial repute values of 1000, 1000, 1000, 1400, and 1800. All five miners can operate well. The experimental parameters were set up in a more differentiated way, and several groups of comparative experiments were made on γ. We observed the results and recorded the growth of the repute values and the proportion of blocks.

Set γ to 0.25, 0.5, 0.75, and 1.0 for the four comparative experiments. Fig. 3 shows the growth of the repute value of each node in this simulation experiment, where the *X*-axis represents the block height, the *Y*-axis represents the node repute value, and each point in the figure represents a block.

Given the same initial repute value, the nodes with high computing power generate blocks frequently, and their repute growth frequency is high, but its overall growth is slow. Given the same computing power, a node's probability of generating blocks is directly proportional to its repute value. The experimental data is provided in Table 2.

Fig. 4 shows the effective repute value of each node in the γ = 1.0 experiment. We can observe in the figure the application of node repute values in the consensus. When the initial repute is the same for all miners, the effective repute value of miner 1, with a

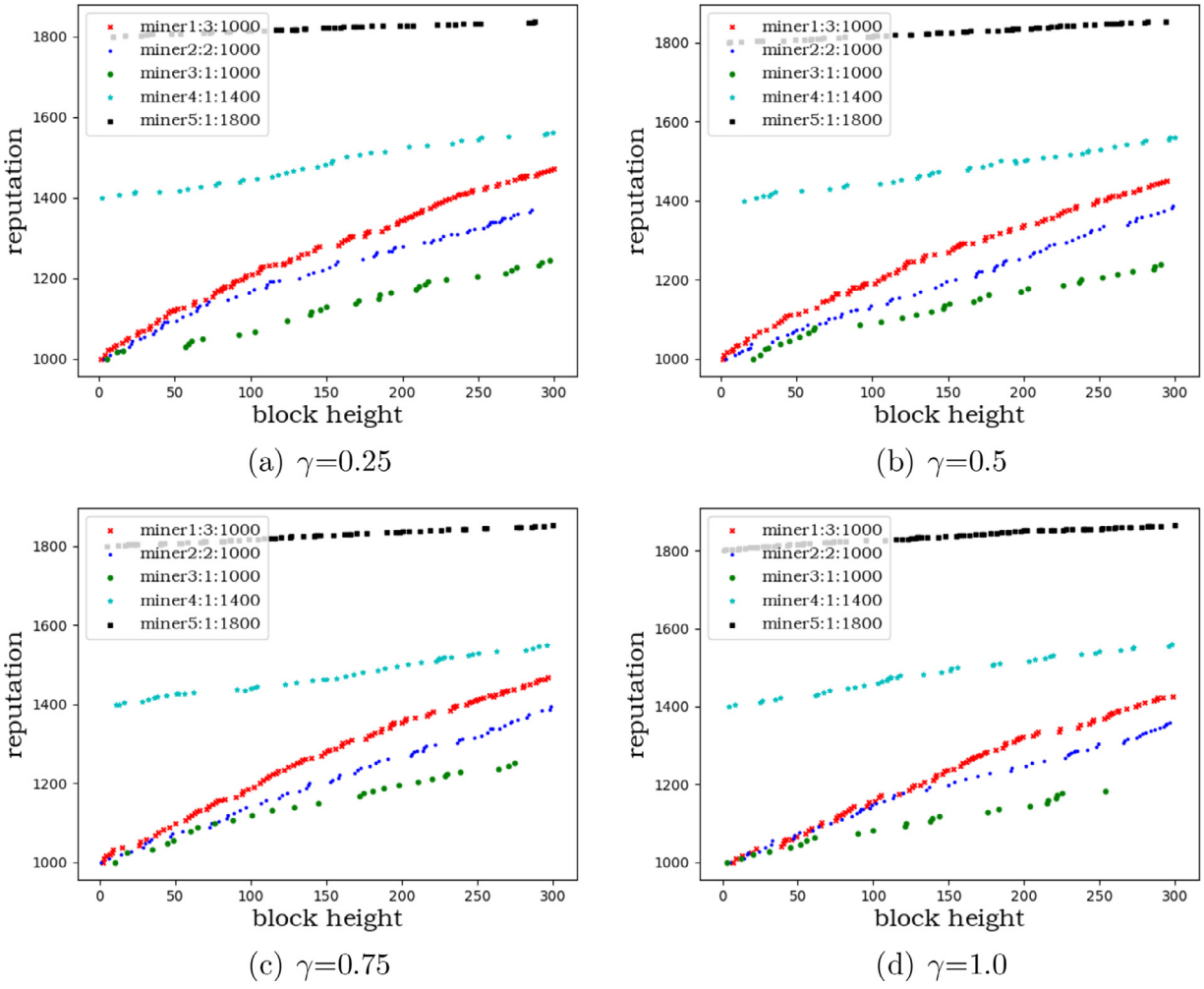


Fig. 3. Simulation experiment statistics with different γ values.

Table 2
Simulation experiment statistics with different γ values.

γ	Miner computing power	Credit	1–50	51–100	101-150	151–200	201–250	251-300	Total
0.25	3	1000	21	19	17	17	22	19	115
0.25	2	1000	14	11	10	12	10	16	73
0.25	1	1000	3	5	6	5	6	6	31
0.25	1	1400	5	7	8	6	5	5	36
0.25	1	1800	7	8	9	10	7	4	45
0.5	3	1000	18	18	17	16	13	16	98
0.5	2	1000	13	13	9	12	14	11	72
0.5	1	1000	6	5	6	5	4	6	32
0.5	1	1400	6	4	8	7	8	8	41
0.5	1	1800	7	10	10	10	11	9	57
0.75	3	1000	15	17	18	18	15	18	101
0.75	2	1000	10	9	9	11	12	14	65
0.75	1	1000	5	4	4	5	5	3	26
0.75	1	1400	8	7	7	7	10	6	45
0.75	1	1800	12	13	12	9	8	9	63
1.0	3	1000	11	15	15	21	10	16	88
1.0	2	1000	10	12	8	6	13	14	63
1.0	1	1000	5	5	5	2	6	1	24
1.0	1	1400	6	7	10	6	8	5	42
1.0	1	1800	18	11	12	15	13	14	83

three-time calculation force, decays twice consecutively, while that of miner 2, with a two-time calculation force, does not decay twice consecutively. For miner 3 with a one-time calculation force, the repute rarely decays; given the same calculation force and a high repute value, the repute of miner 5 decays twice consecutively.

When comparing miner 3 and miner 4, we observe that node 3 has more instances of decay in its effective repute value. Through the decay of its effective repute value, it can therefore be inferred that even when a three-time calculation force occupies 3/8 of the entire network, it is difficult to generate multiple blocks in succes-

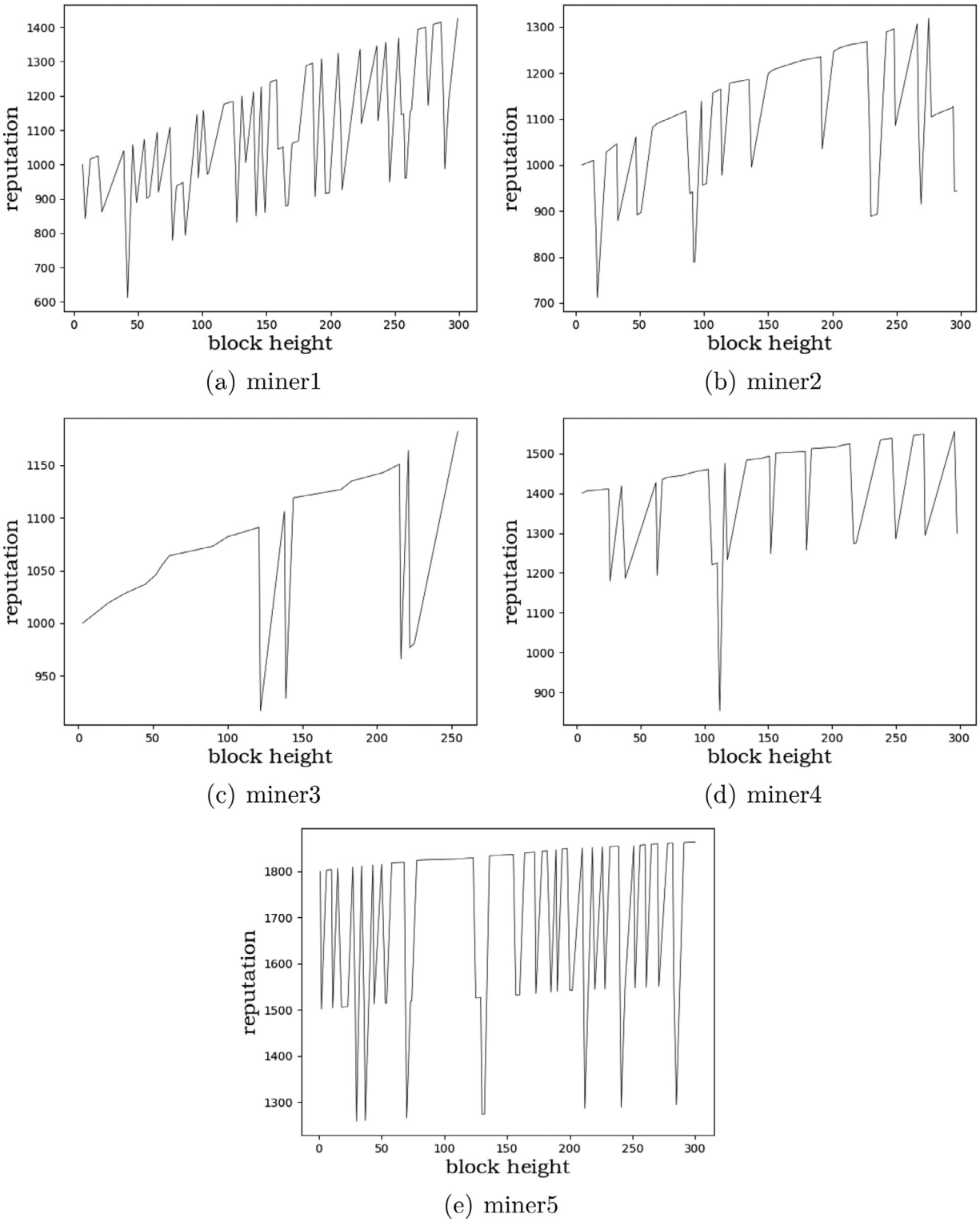


Fig. 4. Effective repute value of miners with different $\gamma = 1.0$.

sion. This proves that the new protocol based on the repute module can effectively improve system security.

Attack simulation experiment This experiment simulates a 51% attack of PoW consensus. An attacker who controls a certain amount of computing power can create its own private chain at the beginning and does not mine on the real chain. Because miners who do not block on the real chain for an extended time receive a repute punishment, the attacker must complete the block out of the real chain once in each decay period. To facilitate the experiment and observation, this experiment assumes that the attacker produces a real block in the attenuation period to avoid the repute penalty. The experimental parameters are set up in a more differentiated way. We set up four miners, one of whom is the attacker who has been forging its own blockchain since the genesis block, while the other three are honest miners. The initial repute of the four miners is set to 1000, and a comparative experiment is con-

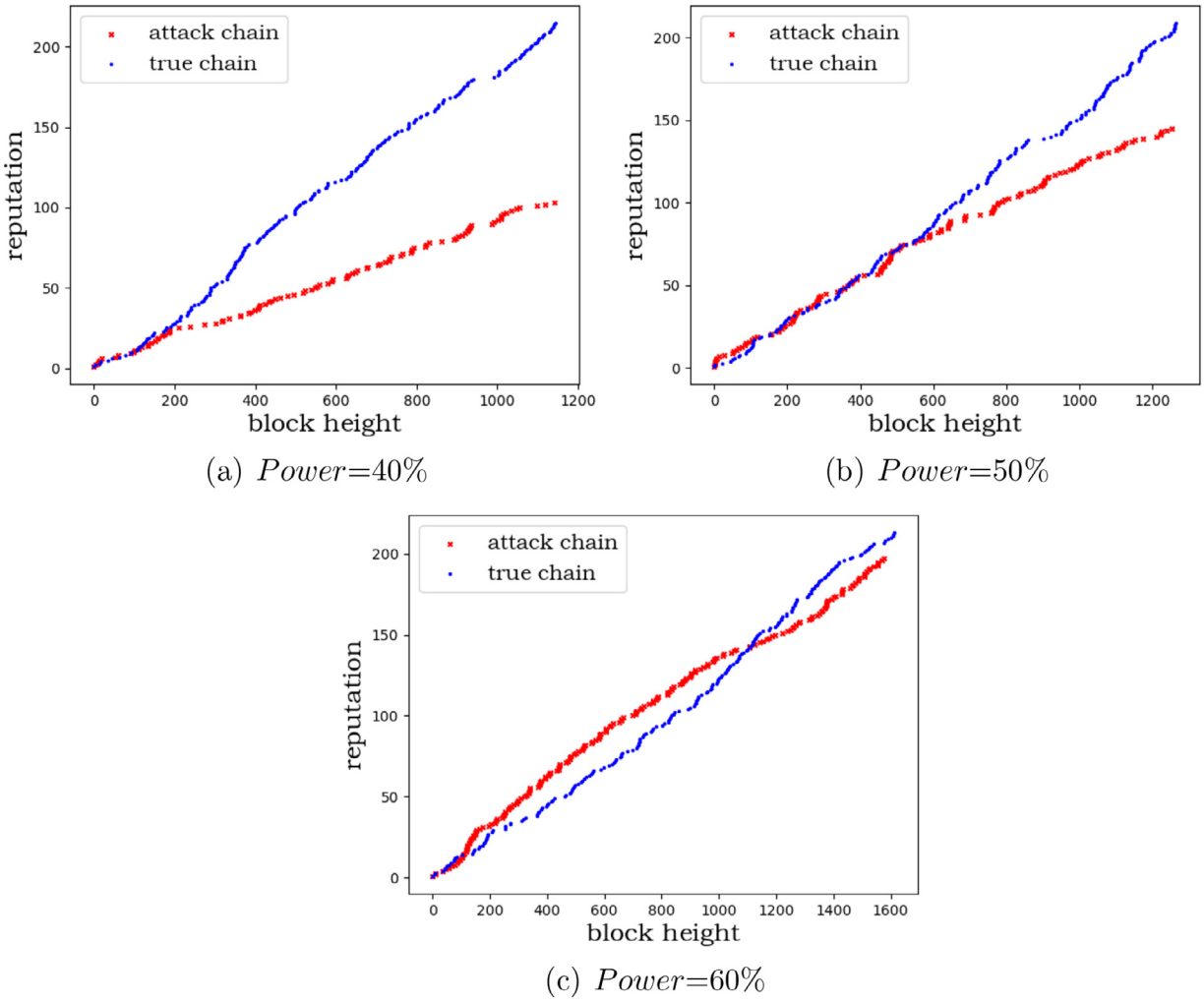


Fig. 5. Attack experiments with different values of *power*.

ducted by giving the attacker different computing power ratios of *power*. Further, we assign different values to the repute conversion rate parameter γ, which makes the experiment more convincing.

Fig. 5 shows attack scenarios in which the attacker has different values of *power*. The *x*-axis is time, and the *y*-axis is block height. The entire figure represents the growth of the real chain and attack chain length over time. The experimental parameters are listed in Table 1.

Fig. 5(a) shows that the blockchain forged by attackers with 40% of the computing power of the entire network has been unable to catch up in length with the real chain. There is a trend of a gradually widening gap, making it difficult to successfully launch an attack.

Fig. 5(b) shows that in the initial stage, the blockchain forged by attackers with 50% of the computing power of the entire network tends to be of similar length as the real chain. However, over time, the repute value of the honest nodes in the real chain increases continuously, leading to the increase of their block rate. The blocker in the attack chain is always the attacker, and it fails to increase its repute value. The decrease in effective repute value makes it difficult for the attacker to block out. Finally, the length of the real chain disposes the attack chain. Therefore, when the repute value of an attacker is similar to that of an honest node, there is a small probability of a successful attack in the initial stage. When the repute value of an honest node is much higher than that of an attacker, the probability of a successful attack is remarkably low.

Fig. 5(c) shows that in the initial stage, the length of the blockchain forged by attackers with 60% of the computing power of the entire network exceeds that of the real chain. As time goes on, the length of the real chain becomes similar to that of the attack chain. Therefore, when the repute value of the attacker is similar to that of the honest node, there is a high probability of attack success in the initial stage. When the repute value of the honest node is much higher than that of the attacker, the probability of attack success is reduced.

Fig. 6 shows attack scenarios with a *power* of 50% computational power. The *x*-axis is time, and the *y*-axis is block height. The entire figure shows the growth of the real chain and the attack chain with time. In addition to the γ parameter, the experimental parameters are listed in Table 1.

Through comparison of Fig. 6(a)–(d), we find that the larger the γ value is, the sooner a gap appears between the real chain and the attack chain. This is because the γ parameter represents the degree of computing power given by node repute conversion. Hence, the larger γ is, the more computing power can be converted by credit. Moreover, the friendlier the consensus is to honest nodes, the smaller the gap between honest nodes and large computing attackers, with the total computing power of the real chain ultimately exceeding that of the attack chain.

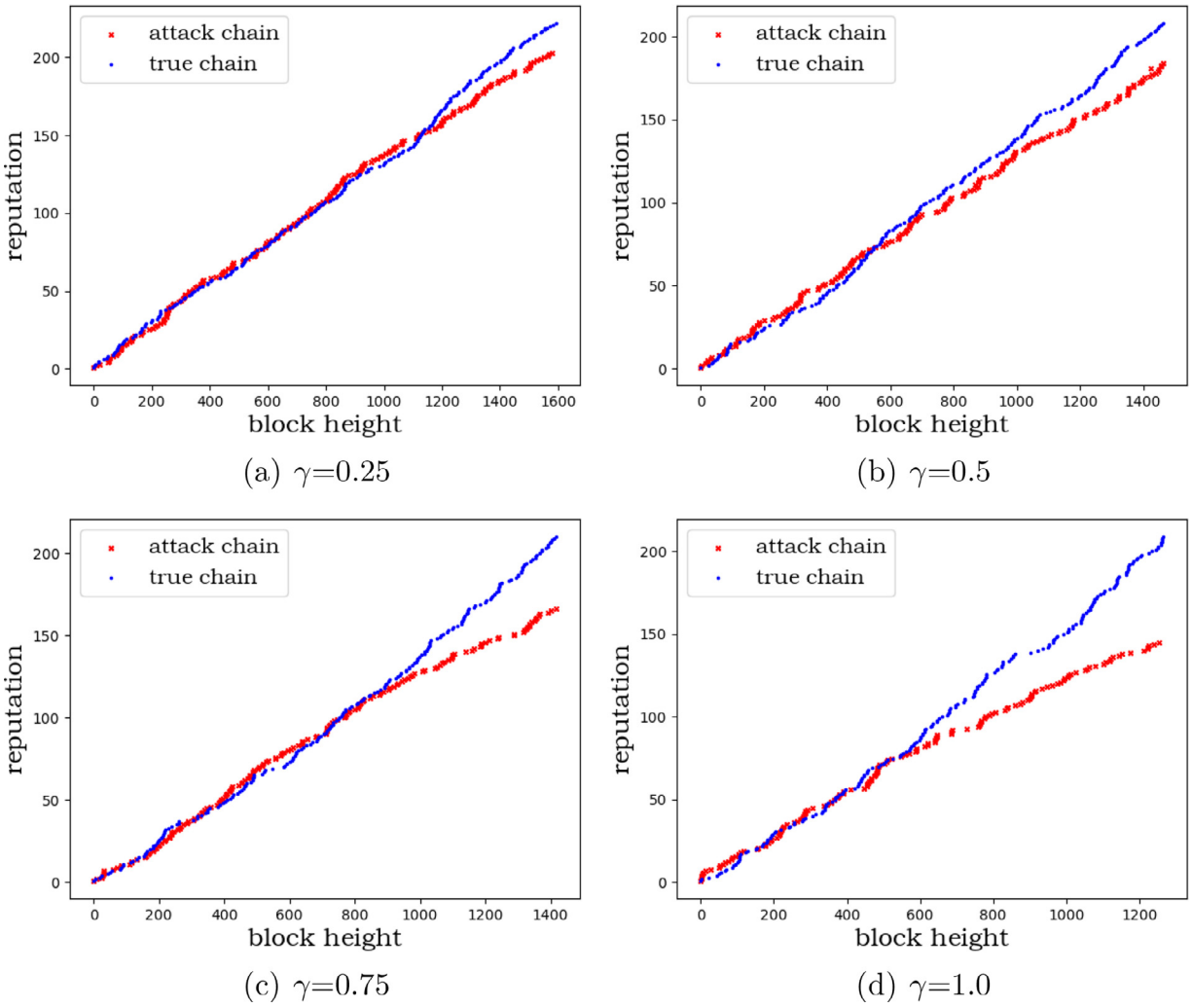


Fig. 6. Attack experiment with different values of γ.

Original protocol comparison experiment To verify whether the new protocol based on the repute module can work as expected, the parameters were set up with greater differentiation between miners. We have four miners; their mining calculation forces are set to three times, one time, one time, and one time, and their initial repute values were set to 1000, 1000, 1400, and 1800, respectively. The protocol parameters are shown in Table 1. All four miners can operate well. We recorded the growth of their repute values and their proportion of blocks.

Fig. 7 shows the growth of the repute value of each node in this simulation experiment. Under normal circumstances, the larger the repute value is, the more difficult it is to obtain credit, with the repute value eventually converging to the maximum value.

Table 3 presents the experimental results for the original protocol and for the new protocol, with each protocol producing 1000 blocks. It shows that under the condition of having the same computing power, the proportion of out blocks for nodes in the original protocol is almost the same, whereas when the computing power is different, the node with the largest computing power has a very high rate of out blocks. The new protocol with the repute module affects the block rate according to the node repute value. For a given level of computing power, the higher the node repute value is, the higher the block rate will be. When the computing power is different and the repute is the same, a higher computing power produces a higher block rate. Finally, we find that the node with the highest repute value can narrow the gap in the block rate with the node with the highest computing power. Thus, malicious attacks from the nodes with high computing power and low repute can be effectively resisted.

Table 3
Comparison of the block proportion of each miner.

Protocol	Miner power	Repute	Blocks generated	Block rate
PoXR	3	1000	422	42.20%
PoXR	1	1000	115	11.50%
PoXR	1	1400	206	20.60%
PoXR	1	1800	257	25.70%
PoW	3	null	522	52.20%
PoW	1	null	158	15.80%
PoW	1	null	168	16.80%
PoW	1	null	152	15.20%

To conduct a comparative experiment on protocol security, we used the same configuration for the original protocol and the new protocol. Each experiment was run with four nodes, one of which was an attacker. By adjusting the proportion of the attacker's computing power to the total computing power, we simulated attackers with different computing power. If the attacker implements a 51% attack, the attacker can produce blocks on its own, while the other three nodes are honest. In the new protocol, the initial repute value of all nodes was set to the initial repute value of *Rinit*. The parameters used in the protocol are shown in Table 1.

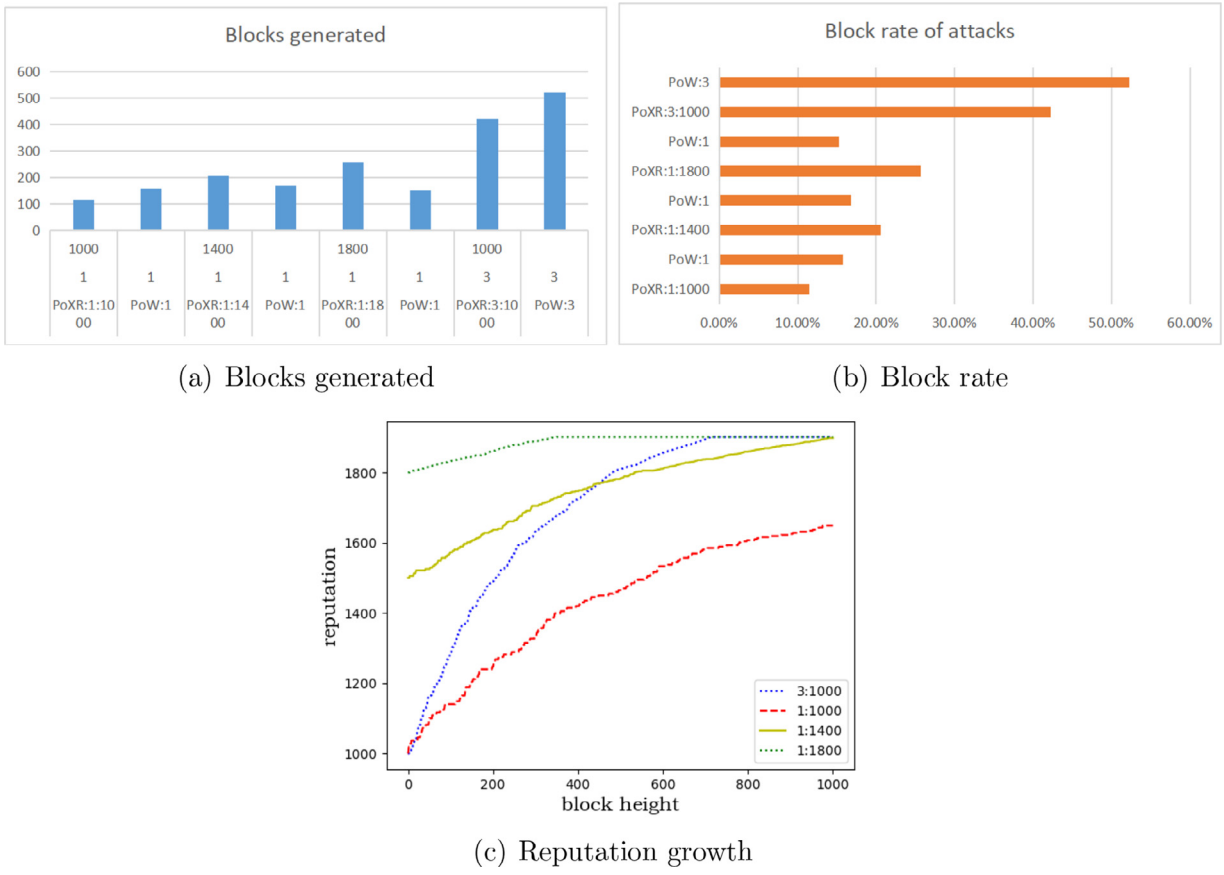


Fig. 7. Single node experiment.

In each experiment, the honest nodes produced 200 blocks. The number of blocks produced by the attacker depends on its proportion of computing power. Suppose the honest nodes produce c_h blocks and the attacker produces c_a blocks. To intuitively reveal the contrast, we consider the attack efficiency $\eth$ of Section 4.2 as the protocol security measurement index; $\eth$ is calculated as $\eth = \frac{c_a}{c_h+c_a}$. Each group of experiments was repeated 10 times and the average value was obtained. The attack efficiency of the new protocol PoXR was lower than that of the original protocol PoW, and the experiment verified the statements in Section 4.2. The security analysis shows that PoXR can improve the security of the original protocol.

Summary According to the above experiments, when the repute values of the nodes in the same computing environment are similar, their blocking rates and repute growth rates are almost the same. When the repute values of the nodes in the same computing environment are different, they fall into a cycle: the blocking rate of nodes with high repute values is higher, but not too high (depending on the parameters), but their repute increases slowly. However, the repute of the nodes with low credit is difficult to increase. When the repute value of nodes with high repute increases to a certain point, their blocking rate continues to improve, but increasing their credit becomes difficult. The blocking rate of the nodes with low repute decreases, but their repute increases steadily. When the repute values of the nodes in different computing environments are the same, the nodes with high computing power have a higher outgoing rate, but their repute value increases slowly. Conversely, the outgoing rate of nodes with low computing power is lower, but their repute value can accumulate rapidly, and the computing power gap between the two nodes narrows.The establishment of a repute system can improve the consensus protocol, resisting stronger attacks, and providing some users with lower computing power a better opportunity to participate in the consensus.

7. Conclusions

In this study, we designed a repute model for consensus protocols for IoT systems. We aimed to provide a new module of the repute method and to show the potential of repute for managing trust in a consensus protocol. In the repute module, satisfactory behavior is encouraged and bad behavior is punished, which can improve the security of the protocol. We implemented a prototype of our method, and the experimental results show that our method has satisfactory performance in terms of efficiency and safety.

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.

CRediT authorship contribution statement

Eric Ke Wang: Conceptualization. **RuiPei Sun:** Data curation, Formal analysis, Writing - original draft. **Chien-Ming Chen:** Data curation, Formal analysis, Writing - original draft. **Zuodong Liang:** Data curation, Formal analysis, Writing - original draft. **Saru Kumari:** Investigation. **Muhammad Khurram Khan:** Project administration, Supervision, Writing - review & editing.

Acknowledgment

This paper is supported by Researchers Supporting Project number (RSP-2019/12), King Saud University, Riyadh, Saudi Arabia.

References

Drijvers, M., Edalatnejad, K., Ford, B., Kiltz, E., Loss, J., Neven, G., Stepanovs, I., 2019. On the security of two-round multi-signatures. In: Proceedings of 2019 IEEE Symposium on Security and Privacy (SP' 2019), pp. 1084–1101. 05

Eyal, I., Sirer, E.G., 2014. Majority is not enough: bitcoin mining is vulnerable. In: Proceedings of the Financial Cryptography and Data Security. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 436–454.

Gervais, A., Karame, G.O., Wüst, K., Glykantzis, V., Capkun, S., 2016. On the security and performance of proof of work blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference.

Gilad, Y., Leung, D., Suhl, A., 2019. On the security of two-round multi-signatures. In: Proceedings of 26th Annual Network and Distributed System Security Symposium (NDSS 2019), 02, pp. 24–27.

Gramoli, V., 2017. From blockchain consensus back to byzantine consensus. Future Gener. Comput. Syst..

Huang, J., Kong, L., Chen, G., Wu, M.Y., Liu, X., Zeng, P., 2019. Towards secure industrial IoT: blockchain system with credit-based consensus mechanism. IEEE Trans. Ind. Inf..

Kang, J., Xiong, Z., Niyato, D., Ye, D., Kim, D.I., Zhao, J., 2019. Toward secure blockchain-enabled internet of vehicles: optimizing consensus management using reputation and contract theory. IEEE Trans. Veh. Technol. 68 (3), 2906–2920.

Karame, G.O., Androulaki, E., Capkun, S., 2012. Double-spending fast payments in bitcoin. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. ACM, New York, NY, USA, pp. 906–917.

Kokoris-Kogias E., Jovanovic P., Gasser L., Gailly N., Syta E., Ford B.. OmniLedger: a secure, scale-out, decentralized ledger via sharding. 2018. 583–598, 05.

Liu, D., Alahmadi, A., Ni, J., Lin, X., Shen, X., 2019. Anonymous reputation system for IIoT-enabled retail marketing atop pos blockchain. IEEE Trans. Ind. Inf. 15 (6), 3527–3537.

Liu, J., Li, W., Karame, G.O., Asokan, N., 2018. Scalable byzantine consensus via hardware-assisted secret sharing. IEEE Trans. Comput. 68 (1), 139–151.

Liu, X., Wang, W., Niyato, D., Zhao, N., Wang, P., 2018. Evolutionary game for mining pool selection in blockchain networks. IEEE Wirel. Commun. Lett. 7 (5), 760–763.

Nayak, K., Kumar, S., Miller, A., Shi, E., 2016. Stubborn mining: generalizing selfish mining and combining with an eclipse attack. In: Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS P), March, pp. 305–320.

Pass, R., Seeman, L., Shelat, A., 2017. Analysis of the blockchain protocol in asynchronous networks. In: Proceedings of the Advances in Cryptology – EUROCRYPT 2017. Springer International Publishing, Cham, pp. 643–673.

Su, Z., Wang, Y., Xu, Q., Fei, M., Tian, Y.C., Zhang, N., 2018. A secure charging scheme for electric vehicles with smart communities in energy blockchain. IEEE Internet Things J..

Wang, E.K., Liang, Z., Chen, C.M., Kumari, S., Khan, M.K., 2020. Porx: A reputation incentive scheme for blockchain consensus of IIoT. Future Generation Computer Systems 140–151. 102

Wang, Y., Su, Z., Zhang, N., 2019. Bsis: blockchain based secure incentive scheme for energy delivery in vehicular energy network. IEEE Trans. Ind. Inf..

Yang, Z., Yang, K., Lei, L., Zheng, K., Leung, V.C.M., 2018. Blockchain-based decentralized trust management in vehicular networks. IEEE Internet Things J. 6 (2), 1495–1505.

Yin J., Wang C., Zhang Z., Liu J.. Revisiting the incentive mechanism of bitcoin-NG2018.

Zamani, M., Movahedi, M., Raykova, M., 2018. Rapidchain: Scaling blockchain via full sharding. In: Acm Sigsac Conference.

Zhang, F., Eyal, I., Escriva, R., Juels, A., Renesse, R.V., 2017. REM: Resource-efficient mining for blockchains. In: 26th USENIX Security Symposium (USENIX Security 17), August. USENIX Association, Vancouver, BC, pp. 1427–1444.

Zhang, R., Preneel, B., 2019. Lay down the common metrics: evaluating proof-of–work consensus protocols' security. In: Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), May, pp. 175–192.

Zohar, A., 2017. Securing and scaling cryptocurrencies. In: Twenty-Sixth International Joint Conference on Artificial Intelligence.

Dr. Eric Ke Wang is an associate professor of Harbin Institute of Technology (HIT), China. Currently, he work as a senior researcher at Key Laboratory of Shenzhen Internet Information Collaborative Technology and Application of HIT. He received a Phd from department of computer science, the University of Hong Kong in 2009. His main research interests include blockchain technology and data security. He has obtained two granted projects from National Science Funding (NSFC) of China. Besides,he has developed two software platforms for blockchain tools and obtained two authorized related patents.

Mr. Ruipei Sun is a post graduate student in Harbin Institute of Technology, Shenzhen, China. He received his bachelor degree from Nanjin PolyTech University, China. Currently, he work as a assistant researcher at Key Laboratory of Shenzhen Internet Information Collaborative Technology and Application of HIT. His main research interests include blockchain and data security.

Dr. Chien-Ming Chen received the Ph.D. degree from National Tsing Hua University, Taiwan. He is currently an Associate Professor of Shandong University of Science and Technology, Shandong 266590, China. He also serves as an Associate Editor of the IEEE ACCESS. His current research interests include network security, blockchain, the mobile Internet, the IoT, and cryptography. He serves as an Executive Editor of the International Journal of Information Computer Security.

Mr. Zuodong Liang is a post graduate student in Harbin Institute of Technology, Shenzhen, China. He received his bachelor degree from Shandong Science and Technology University, China. Currently, he work as a assistant researcher at Key Laboratory of Shenzhen Internet Information Collaborative Technology and Application of HIT. His main research interests include blockchain and data security.

Dr. Saru Kumari is currently an assistant professor with the Department of Mathematics, Ch. Charan Singh University, Meerut, Uttar Pradesh, India. She received her Ph.D. degree in Mathematics in 2012 from CCS University, Meerut, UP, India. She has published more than 136 research papers in reputed International journals and conferences, including 119 publications in SCI-Indexed Journals. She is on the Editorial board of AEÜ - International Journal of Electronics and Communications, Elsevier (SCI); International Journal of Communication Systems, Wiley (SCI-E); Telecommunication Systems, Springer (SCI); Human Centric Computing and Information sciences, Springer (SCI-E); Transactions on Emerging Telecommunications Technologies; Wiley (SCI-E), KSII Transactions on Internet and Information Systems (SCI-E), published from Taiwan; Information Security: A Global Perspective, Taylor & Francis (ESCI, Scopus); International Journal of Wireless Information Networks (ESCI, Scopus), Springer; Security and Privacy, Wiley; Journal of Computing and Information Technology, (Scopus); Iran Journal of Computer Science, Springer; Azerbaijan Journal of High Performance Computing, published by Azerbaijan State Oil and Industry University, Azerbaijan. She served as GuestEditor of the Special Issue "Big-data and IoT in e-Healthcare" for Computers and Electrical Engineering, Elsevier (SCI-E), Elsevier. She is Technical Program Committee Member for more than a dozen of International conferences. She is a reviewer of more than 50 reputed Journals including SCI-Indexed Journals of IEEE, Elsevier, Springer, Wiley etc. Her current research interests include information security and applied cryptography.

Dr. Muhammad Khurram Khan is currently working as a Professor of Cybersecurity at the Center of Excellence in Information Assurance (CoEIA), King Saud University, Kingdom of Saudi Arabia. He is one of the founding members of CoEIA and has served as Manager R&D from March 2009 until March 2012. He, along with his team, developed and successfully managed Cybersecurity research program at CoEIA, which turned the center as one of the best centers of excellence in Saudi Arabia and in the region. He is founder and CEO of the 'Global Foundation for Cyber Studies and Research' (http://www.gfcyber.org), an independent, non-profit, and non-partisan cybersecurity think-tank registered in Washington D.C. Prof. Khurram is the Editor-in-Chief of a well-reputed International journal 'Telecommunication Systems' published by Springer-Nature for over 26 years with its recent impact factor of 1.707 (JCR 2019). He is the Founding Editor of 'Bahria University Journal of Information & Communication Technology (BUJICT)'. Furthermore, he is the editor of several international journals, including, IEEE Communications Surveys & Tutorials, IEEE Communications Magazine, IEEE Internet of Things Journal, IEEE Transactions on Consumer Electronics, Journal of Network & Computer Applications (Elsevier), IEEE Access, IEEE Consumer Electronics Magazine, PLOS ONE, Electronic Commerce Research, IET Wireless Sensor Systems, Journal of Information Hiding and Multimedia Signal Processing, and International Journal of Biometrics, etc. He has also played role of the guest editor of several international journals of IEEE, Springer, Wiley, Elsevier Science, and Hindawi. Moreover, he is one of the organizing chairs of more than 5 dozen international conferences and member of technical committees of more than 10 dozen international conferences. In addition, he is an active reviewer of many international journals as well as research foundations of Switzerland, Italy, Saudi Arabia and Czech Republic. Prof. Khurram is an honorary Professor at IIIRC, Shenzhen Graduate School, China and an adjunct professor at Fujian University of Technology, China. He has secured an outstanding leadership award at IEEE international conference on Networks and Systems Security 2009, Australia. He has been included in the Marquis Who's Who in the World 2010 edition. Besides, he has received certificate of appreciation for outstanding

contributions in 'Biometrics & Information Security Research' at AIT international Conference, June 2010 at Japan. He has been awarded a Gold Medal for the 'Best Invention & Innovation Award' at 10th Malaysian Technology Expo 2011, Malaysia. Moreover, in April 2013, his invention has got a Bronze Medal at '41st International Exhibition of Inventions' at Geneva, Switzerland. In addition, he was awarded best paper award from the Journal of Network & Computer Applications (Elsevier) in Dec. 2015. Prof. Khurram is the recipient of King Saud University Award for Scientific Excellence (Research Productivity) in May 2015. He is also a recipient of King Saud University Award for Scientific Excellence (Inventions, Innovations, and Technology Licensing) in May 2016. He has published more than 360 papers in the journals and conferences of international repute. In addition, he is an inventor of 10 US/PCT patents. He has edited 9 books/proceedings published by Springer-Verlag, Taylor & Francis and IEEE. He has secured several national and international competitive research grants with an amount of over USD 3 Million in the domain of Cybersecurity. Prof. Khurram has played a leading role in developing 'BS Cybersecurity Degree Program' and 'Higher Diploma in Cybersecurity' at King Saud University. In 2019, he has played an instrumental role as a cybersecurity subject expert for a USD 6 Million series B investment in a South Korean startup 'SecuLetter', which has received a corporate valuation of USD 38 Million. His research areas of interest are Cybersecurity, digital authentication, IoT security, biometrics, multimedia security, cloud computing security, cyber policy, and technological innovation management. He is a fellow of the IET (UK), fellow of the BCS (UK), fellow of the FTRA (Korea), senior member of the IEEE (USA), senior member of the IACSIT (Singapore), member of the IEEE Consumer Electronics Society, member of the IEEE Communications Society, member of the IEEE Computers Society, member of the IEEE Technical Committee on Security & Privacy, member of the IEEE IoT Community, member of the IEEE Smart Cities Community, and member of the IEEE Cybersecurity Community. He is also the Vice Chair of IEEE Communications Society Saudi Chapter. He is a distinguished Lecturer of the IEEE.