A Sharding Blockchain Model for Scalable Trust Management in Social IoT

Amin Rouzbahani, Fattaneh Taghiyareh*
School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Iran; rouzbahani@ut.ac.ir, ftaghiyar@ut.ac.ir

ABSTRACT

Today, the Internet of Things is a widely recognized phenomenon that generates a significant amount of data and connects many devices. Many products are incorporating electronic components to facilitate their integration and interaction with the Internet. Scalable and efficient trust management systems are required to maintain network reliability, considering the increasing number of IoT devices and generated data. In order to enable scalable trust management in social IoT, this paper presents a sharding-based scalable trust management approach that combines social interactions with smart contract functionality. Through the division of transaction state into smaller segments and the enhancement of trust value propagation among connected devices, sharding techniques in blockchain can offer scalable trust management protocols. When implementing the model on the Hyperledger Fabric platform, we carried out a thorough evaluation. The model calculates trust in terms of trust convergence and success rate efficiently. We have conducted several tests to evaluate the scalability of the model. To boost it, we have also implemented the state sharding. We also conducted a study to highlight the advantages of the sharding strategy on the scalability of the model. The results demonstrate that using shards significantly improves trust management capacity on the blockchain. The proposed method demonstrates the potential application of sharding in blockchain-based Trust Management (TM) for scalable trust management in SIoT.

Keywords— Sharding; Trust Management; Internet of Things; Blockchain.

1. Introduction

The number of devices used on the Internet has significantly increased. The Internet of Things is generating a substantial amount of data. Researchers have paid substantial attention to the Internet of Things (IoT), a rapidly growing technology that finds use in diverse applications. A network of connected devices makes up the Internet of Things (IoT), using various protocols, topologies, and software to collect and exchange data [1]. While the Internet of Things (IoT) offers numerous benefits to humans, it also faces challenges that threaten the advancement of IoT applications. Implementing procedures for this technology is critical to ensuring the dependability and accuracy of data produced and sent within its network. Furthermore, the creation of interconnected networks facilitates remote access to IoT devices, thereby establishing connections between all individuals and objects. This gives rise to potential threats from adversaries, increasing concerns over the safeguarding of security and privacy [2]. IoT applications need procedures to safeguard the data from potential malicious activities.

Adversaries increase their own advantage by specifically targeting and interrupting other entities' operations. These nodes have the potential to harm the reputation of honest nodes or enhance the credibility of other malicious nodes, compromising the functionality of the network. In order to ensure a secure and safe environment, it is crucial to limit the presence of malicious nodes and promote trust among the honest nodes. As confidence between reliable nodes grows, they work together and interact with their trustworthy counterparts in the system to offer improved services. These difficulties highlight the necessity of implementing Trust Management in the IoT environment to enhance system accuracy, ensure high-quality services, and enhance information security.

Current trust management systems are not efficient in meeting this need due to constraints in storage capacity and computational resources. Many

http://dx.doi.org/10.22133/ijwr.2024.451270.1231

Citation A. Rouzbahani, F. Taghiyareh, "A Sharding Blockchain Model for Scalable Trust Management in Social IoT", *International Journal of Web Research*, vol.7, no.3,pp.29-39, 2024, doi: http://dx.doi.org/10.22133/ijwr.2024.451270.1231.

*Coressponding Author
Article History: Received: 22 March 2024 ; Revised: 14 June 2024; Accepted: 19 June 2024.

owners may act maliciously to advance their own benefits by carrying out attacks.

Users with strong social connections may collude to damage the image of rival devices by engaging in reputation attacks, while simultaneously enhancing their own reputation. An IoT trust management protocol needs to be resistant to such attacks in order to be sustainable. Moreover, IoT devices are primarily carried or handled by humans. Trust management should consider the social links between device owners to optimize protocol performance.

Blockchain technology has been employed in various approaches to maintain trust in the Internet of Things (IoT). Various studies have examined Blockchain-based TM methods, including those by I. R. Chen et al. [3], D. Chen et al. [4], X. Chen et al. [5], and Lin et al. [6]. However, blockchain-based techniques for TM have significant obstacles, including scalability issues and a requirement to manage real-world constraints such as limited resources and low computing capabilities on resource-limited devices. There are still unresolved issues with trust management systems. The goal of this work is to create and verify a scalable and optimal protocol for SIoT systems in order to address the aforementioned issues. We have introduced a trust management protocol called SCoTMan. This protocol operates on a permission-based blockchain that is scalable and has high throughput. It aims to maximize the usage of trust resources and compute by employing sharding techniques, hence enhancing its scalability.

Blockchain sharding is employed to address the scalability challenges mentioned above. This process entails dividing the ledger into distinct shards, each of which is maintained by specific groups of nodes [7]. By dividing tasks across different shards, system performance increases proportionally with the number of shards. In order to carry out cross-shard transactions, which involve transactions that operate on different shards, sharding blockchains need to utilize a cross-shard commit protocol. This protocol is necessary to guarantee the atomicity of transactions, but it has a major influence on the performance of the system.

The goal of the study is to develop and test a scalable and resource-efficient protocol for SIoT systems to address the identified problems. We will achieve this by integrating blockchain technology with weighted sum approaches to trust management. The contribution includes the incorporation of social properties within the SIoT and the global viewpoint provided by blockchain, which boosts both direct and indirect trust evaluations. We have subjected the TM protocol to multiple assessments to validate its functional and non-functional performance within the HLF environment. We assess and improve the model's scalability through state sharding. We have developed a trust management system on HLF to support IoT applications.

2. Literature Review

2.1. Social IoT and trust

Liu's survey [8] discusses the classification of blockchain-based trust management systems based on the need to compute trust values. Trust is conveyed by specific trust values in the initial category, referred to as trust value-based. Trust value-free refers to the second category, characterized by the absence of a predetermined technique for assessing trust and the lack of a nuanced differentiation between trust relationships. The research we conducted falls into the first category. Therefore, we examine a variety of works from this specific genre in the field of literature.

The Social IoT concept (SIoT) considers IoT objects as socially connected nodes, combining the characteristics of social networks with IoT networks [9]. The social structure, in conjunction with feedback derived from previous actions of the nodes, can serve as a valuable resource for evaluating a device's reliability. An effective approach to managing trust in Internet of Things systems involves analyzing the social connections among interconnected devices [10].

Furthermore, alongside blockchain-based solutions, there are studies that employ social ties within the context of the Social Internet of Things (SIoT). Trust is established by aggregating both indirect and direct trust between two nodes [3]. This goal is achieved by utilizing the dynamic weighted summation technique. Weights are modified in accordance with the provider's performance. The average of the beta distribution is used to calculate direct trust. After each transaction, users evaluate the performance of the service provider and assess direct trust. Indirect trust is established based on the opinions of individuals who provide recommendations, under the assumption that these individuals and the ones who rely on their recommendations have similarities such as friendship, social interaction, and shared interests. Indirect trust is determined by aggregating the evaluations of recommenders and assigning weights based on their similarities.

2.2. Blockchain and Sharding

Blockchain technology is a decentralized and transparent system for recording transactions, allowing for efficient and secure transfers of ownership between parties. This transaction employs public-key cryptography and consensus protocols to guarantee security. A cryptographic hash connects the blocks to previous blocks, storing the transaction data in each block. After a consensus is achieved, valid blocks become unchangeable and stay on the chain, maintaining the unchangeability of transactions. The transactional data is stored in an endless sequence of blocks interconnected by cryptographic methods. A decentralized time-stamping technique organizes the blocks, enabling peers to collaboratively evaluate the accuracy of database updates and ultimately reach a consensus on the correct block order and a mutually accepted system state. Hence, the individuals involved in a blockchain network can interact with each other without depending on a central governing body to resolve disagreements regarding the accurate order and specifics of transactions. Blockchain-based systems aim to eliminate any single point of failure and guarantee the existence of comprehensive, transparent, and verified transaction records. These characteristics enable economical interactions, simplify contracts [11], and enhance the accessibility of information sharing. In addition, blockchain offers conflict resolution by creating an unchangeable record of transactions that is accessible to the public.

Blockchain technology has found applications in various sectors, such as healthcare, electronics, manufacturing, education, economics, social networking, and more. The adoption of blockchain technology in different industries has garnered significant attention from researchers, who are exploring its potential not just for enhancing security but also for enabling smart contracts, promoting transparency, and developing decentralized solutions, among other applications. A strategy for automated smart contract management enables safe hierarchical communications by leveraging shared secrets [12].

The SIoT devices communicate with each other to exchange information or offer services. As stated in [13], edge nodes can provide support to nodes that have weak communication or processing skills. Examples of edge nodes include smartphones, roadside gadgets, and gateway routers. SIoT applications, including Smart City, Smart Healthcare, Smart Energy, Smart Vehicles, and Smart Industry solutions, can link with the edge layer, which includes blockchain nodes. We can receive and process the transactions on the main blockchain or through other channels, known as off-chain, to enhance scalability. Edge nodes might choose to establish contact with the cloud layer, which has greater computing power and data storage capacity. The cloud layer enables the execution of complex tasks such as consensus protocols, smart contract execution, and the exploitation of artificial intelligence algorithms. In Wu et al. [14] authors proposed a privacy-preserving trust management architecture named PPTMA for IIoT systems that uses a game-theory based incentive mechanism to reward or punish the devices. The solution is evaluated in an HLF environment without assessing non-functional performance of the model.

In Table 1, we assessed trust management methodologies in the SIoT that utilize blockchain technology. The evaluation criteria cover the consensus mechanism employed, the extent to which scalability was examined, the performance indicators taken into account, and the implementation platform or simulation used. While the majority of the research was to demonstrate their algorithm's scalability, there are concerns about its capacity to scale. One possible explanation is that most of them do not employ a scalable consensus mechanism. Furthermore, our research reveals that the evaluation of blockchain-based techniques neglects several crucial performance aspects like connection overhead, storage, and computing expenses. Implementing Trust Management (TM) in the Internet of Things (IoT) using platforms like Ethereum or Bitcoin is not practical due to the scalability issues associated with the Proof of Work (PoW) methods. In order to fill this need, we attempted to create a blockchain-powered Trust Management system for the SIoT. This system uses a scalable consensus mechanism to evaluate the platform's expenses. Moreover, as we addressed these shortcomings, we significantly improved the TM itself.

Originally proposed for conventional distributed databases, sharding is a thoroughly studied strategy to improve scalability. Well-known databases such as MongoDB, MySQL, and Bigtable [20] employ sharding as a technique to alleviate the burden on the central server. Sharding divides the data into discrete pieces and stores it on different servers, resulting in enhanced performance. However, the unique adversary models of blockchain prevent direct application of the technology. The malevolent node within the database has limited capabilities, as it can only withhold messages or become disconnected from the network. However, hostile nodes within the blockchain have the ability to carry out arbitrary assaults. Therefore, to improve its robustness, we can incorporate additional procedures into sharding. More precisely, the methods that use sharding involves dividing nodes into several segments. Afterwards, the system segregates related actions and

Table 1. Comparison of Some BC-based TM in IoT

Study	***Features***		
	Consensus	***Scalability***	***Implementation***
Putra [15]	POW	×	×
Moinet et al. [16]	×	✓	Sim.
Azad et al [17]	×	×	Sim.
Lin et al. [18]	✓	✓	Sim.
Latif et al. [19]	×	×	×
Wu et al. [14]	✓	×	HLF
this study	✓	✓	HLF

a Sim.: Simulation × Not Addressed ✓ Addressed HLF: Hyperledger Fabric objects into several autonomous divisions. Objects encompass several components, such as transactions, blocks, and ledgers. The activities included in this process are verifying transaction, broadcasting data, and updating storage. Several shards can concurrently process transactions, and enhance the blockchain's scalability. Separate shard nodes exclusively handle, execute, and retain transactions associated with their own shards, lowering the computing cost and also causes reduction in storage and network costs. The primary concept underlying sharding is the notion of divide and conquer. Hence it can enhance the scalability across storage, computation, and networking [21, 22]. Hence, the sharding blockchain can be categorized into some distinct types: network, transaction, and state sharding. Network sharding is the process of partitioning the entire blockchain network into smaller shards, which are also referred to as committees. Nodes on public blockchains like Bitcoin do not need to download and validate all records. Instead, they are provided merely with the specific transactions that are assigned to them. This decreases the amount of network bandwidth required, which is particularly important in these blockchains where network overhead is a concern.

In public blockchains, network sharding is commonly regarded as the foundation for the two types of sharding, as stated by Huang et al. [23]. However, in permissioned blockchains, this is not always true. Transaction sharding is the process of separating legitimate transactions into distinct groups and allocating them to various validators. The consensus within each set of validators is limited to the present shard. Sharding decreases the computational burden on individual nodes and minimizes the minimum requirement for node computing capacity. State sharding, often referred to as storage sharding, involves the storage of the ledger associated with each shard independently by each committee, rather than having all nodes store the full history ledger.

3. SCoTMan model

It is assumed that all nodes in P2P or distributed solutions for trust management in the IoT have the ability to perform computations, store data, and disseminate data using trust management methods. However, this is not achievable. Our technique implements trust management procedures using a smaller number of reliable blockchain nodes. Internet of Things (IoT) devices with limited communication and resource capabilities only need to connect to the blockchain layer, requiring fewer and less resource-intensive operations. Blockchain is a decentralized system that utilizes robust encryption to ensure the confidentiality and anonymity of transactions. In addition, smart contracts based on blockchain technology have the capability to automate trust management. The architecture enables IoT nodes to collaborate and maintain trust values by utilizing feedback and prior actions to manage trust. This method entails storing data for the trust management technique and automating it through the use of smart contracts and blockchain technology. This method uses consensus mechanisms within the blockchain to establish distributed agreements over the status of the ledger, which stores trust-related data, instead of relying on a centralized trusted authority. We can classify nodes into two categories: devices and users (or owners). The user possesses multiple devices. In our trust management system, a user becomes the trustor and another user's device becomes the trustee. Each node can be uniquely identified by its own address.

Figure 1 presents a brief description of the provided framework. This strategy involves using Internet of Things (IoT) devices to establish communication with the blockchain network in order to request services or gather information. The framework utilizes the Hyperledger Fabric (HLF) technology, which has the ability to store transactions and execute smart contracts that control trust management on the blockchain. Validators on the blockchain perform vital tasks and store critical data on it. There are features available that make it easier to request services or provide feedback to devices on the blockchain. In addition, nodes can choose to act as validators or *orderers*, and the network will select them based on their reputation or other criteria. This selection of *orderer* or validator is beyond the scope of this research.

The approach for assessing trust in Social IoT through the utilization of blockchain technology is illustrated in Figure 1. The proposed method functions as follows: The trustee sends a request to the blockchain, which includes parameters containing its list of friends. The blockchain verifies the trustee's identity, records the transaction request, and generates a transaction ID. Additionally, it executes the smart contract to determine the trust value of the trustor. Once the trustee requests a service from the trustor, it informs the blockchain by supplying the transaction feedback value. The smart contract employs a trust evaluation mechanism to authenticate and record its past activities. The smart contract initiates the process of upgrading trust at previously determined intervals. Subsequently, it carries out the protocol and modifies the trust values accordingly.

We store each feedback record on a blockchain until it significantly impacts the trust calculation, at which point we use it to calculate trust for more nodes. Direct interactions and experiences between the trustee and the trustor, along with indirect information about the trustee from other system nodes, primarily determine the trust value [24]. In order to accomplish this objective, it is necessary for

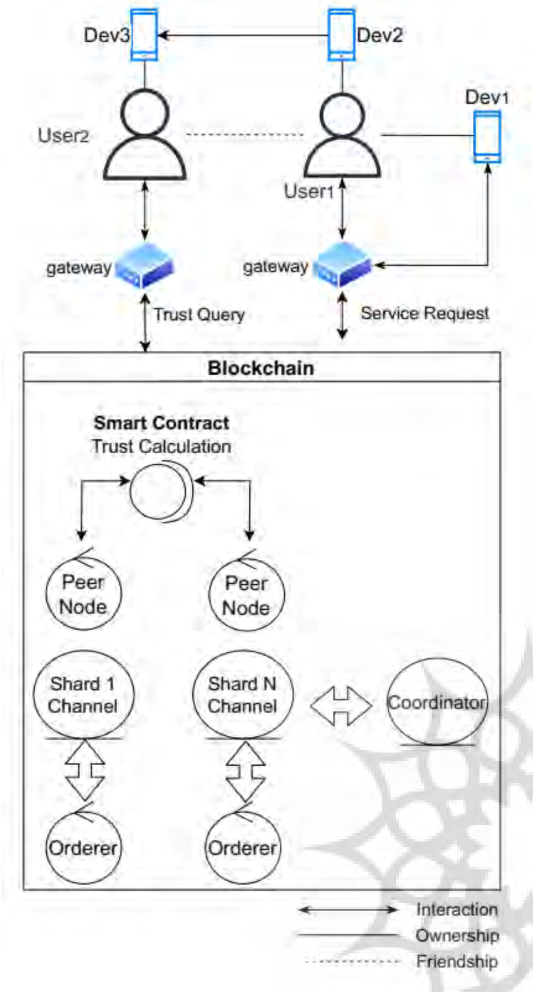


Figure. 1. Overall Architecture of SCoTMan

each device to maintain an updated list of recommenders, considering the social connections among IoT devices. The device then derives its recommendations from the previous transactions recorded in this list. This strategy relies on collecting exclusive suggestions from specific devices that have a closer relationship with the trustor.

The SCoTMan system [25–27] comprises three fundamental procedures: *QueryTrust*, *RequestService*, and *SubmitFeedback*. The trust value of a device can be acquired using the *QueryTrust* method, which serves as an interface. This allows users to determine the level of trustworthiness attributed to a specific device within the system. Nevertheless, when a device endeavors to establish communication with another device on the network in order to access a specific service, it employs the *RequestService* function. This technology facilitates the coordination and communication amongst devices, enabling the provision of vital services. The "*SubmitFeedback*" mechanism is utilized to transmit high-quality feedback regarding the services provided by a device, hence facilitating ongoing enhancements within smart contracts.

We used a Two-Phase Commit (2PC) protocol for cross-shard transaction handling. Transactions whose state spans more than one shard will follow a two-phase protocol. In the first phase, known as the prepare phase, a transaction is proposed to the endorsing peers of each shard, and they will execute the validation of the transaction itself without committing the transaction and return an acknowledgment to state if they are ready. In the second phase, if all shards are prepared-that is, the transaction is ordered and broadcast-the state in each shard is updated atomically. If any shard fails during the prepare phase, the transaction is aborted; that is, no shard commits the transaction. This mechanism ensures that the network operates under the consistency rule: either all the shards commit a transaction or none does [28]. To reduce cross-shard overhead, we utilize the interaction list to relocate the user's account from one shard to the shard containing those with the highest interactions.

3.1. Direct Trust

We applied the widely recognized and well-established Bayesian technique for measuring direct trust from user satisfaction experiences in trust and reputation systems [29]. Following a direct encounter, a requester may give a provider feedback on the quality of the provided services. Feedback criteria could include failures, costs, reaction time as assessed by the requester, and more. A value, $f_{i,j}$, indicates the current satisfaction experience of node U_i toward device D_j. We explore the straightforward scenario where the direct user satisfaction experience $f_{i,j}$ is represented by a 1 where denotes satisfaction and 0 denotes discontent. Afterwards, we can conceive of $f_{i,j}$ as the outcome of Bernoulli trials, where $\theta_{i,j}$, the likelihood of success variable, has a beta distribution. The predicted value of $\theta_{i,j}$, that is, the direct trust between node U_i and device D_j, is computed (Equ(1)). As is common in the literature, we initialize both α and β to 1, which results in a neutral trust value of 0.5.

$$DT_{i,j}(t) = \frac{\alpha_{i,j}(t)}{\alpha_{i,j}(t) + \beta_{i,j}(t)} \qquad (1)$$

3.2. Indirect Trust

Users can directly engage with other devices by transmitting their requests to the SCotMan. Subsequently, the blockchain collects their comments. Blockchain can utilize your comment as a suggestion for future transactions. If necessary, a device can also request trust recommendations from another device belonging to a friend. The recommendation approach will ensure privacy by maintaining the confidentiality of user and device identities.

In Figure 2, we propose a two-phase recommendation selection process to determine the most appropriate recommenders. When there are limited interactions with a device, less information is gathered through direct observation. Instead, we select recommenders based on our understanding of relationships. In Social IoT, the similarity between two nodes is measured using similarity metrics to assess their relationship. By leveraging the connections within the system, we can make more effective and useful recommendations. The following variables, which represent the relationships between devices, are used to calculate this parameter. For similarities we use Jaccard index as the following (M is the similarity metric base on F sets) (Equ(2)):

$$M_F = \frac{|F_i \cap F_j|}{|F_i \cup F_j|} \qquad (2)$$

Owner's Social Network: Each device stores a list of its owner's friends and sends it, together with any supplementary data, to the blockchain. The device's owner has provided this list. By leveraging the Jaccard similarity coefficient, this friendship list can be used to compute the friendship metric between two devices.

Interest Metric: Users who share similar interests are likely to have comparable social interests when using a service supplied by the same device. Moreover, there is a high probability that a community of interest will be established between two customers who have previously utilized the services of the same IoT device. The blockchain can compute this metric using the same approach as the friendship metric, as it already records the interactions between devices.

Contact similarity: This is a metric that measures the degree of similarity between two nodes by assessing their physical connection. It functions as an estimate of the nodes' perspectives on devices that offer the identical service. The operational area can be partitioned into smaller grids. The user keeps a log of locations and allows their devices to utilize it when sending a transaction to the blockchain. The list is employed to compute the Contact Metric for devices by a comparable calculation.

3.3. Trust Calculation in SCoTMan

As shown in Algorithm 1, the function *Submit_Feedback* receives the feedback record from node *i* to node *j*, along with optional values of last *location* and *friendlist*, which are sent only when they are modified for node *i*. The procedure identifies the most recent interaction record of the nodes and thereafter verifies whether the feedback is derived from prior interaction requests or not. Subsequently, it verifies and prevents any attempts of self-promotion attacks. After receiving the values, it stores them and logs the feedback in the *PENDING_FEEDBACK* list for future updates.

In Algorithm 2, the process of updating trust includes all outstanding feedback in both direct and indirect trust calculations. After updating the global trust values, the lists are sorted based on these values, and only the top *k* entries are kept while the rest are discarded. If the quantity of these recommenders is lower than half of the value of *k*, the algorithm

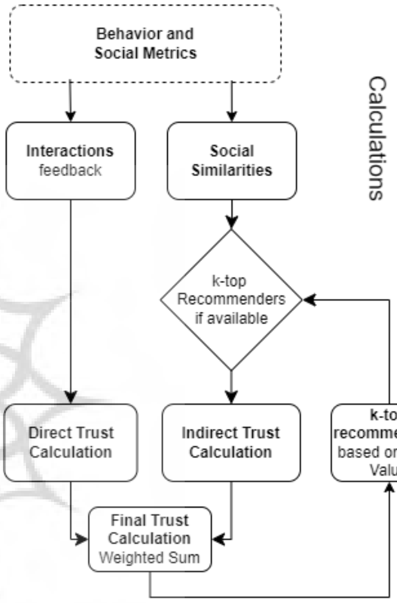
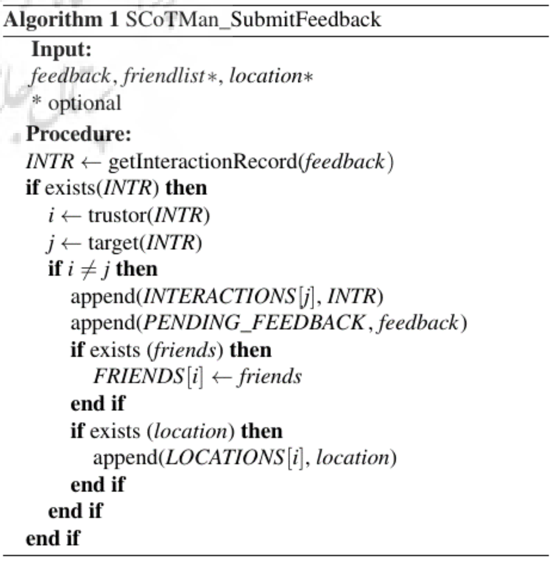

Behavior and Social Metrics
Calculations
Interactions feedback
Social Similarities
k-top Recommenders if available
Direct Trust Calculation
Indirect Trust Calculation
k-top recommenders based on Trust Value
Final Trust Calculation Weighted Sum

Figure. 2. SCoTMan calculations



```
Algorithm 1 SCoTMan_SubmitFeedback
 Input:
 feedback, friendlist*, location*
  * optional
 Procedure:
 INTR ← getInteractionRecord(feedback)
 if exists(INTR) then
   i ← trustor(INTR)
   j ← target(INTR)
   if $i \neq j$ then
     append(INTERACTIONS[j], INTR)
     append(PENDING_FEEDBACK, feedback)
     if exists (friends) then
       FRIENDS[i] ← friends
     end if
     if exists (location) then
       append(LOCATIONS[i], location)
     end if
   end if
 end if
```

Algorithm 1 *SubmitFeedback* procedure in SCoTMan

calculates social similarities and utilizes them for indirect suggestions.

4. Evaluation and Results

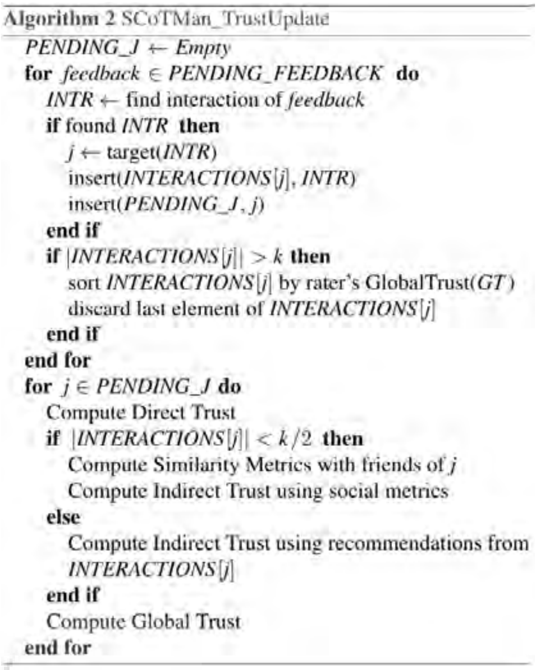
In this section, we describe the simulation process and its outcomes obtained by implementing our proposed trust management protocol on IoT devices. The objective of our study was to develop a streamlined method that achieves a success rate similar to earlier research, while efficiently managing trust of nodes with low computational and communication cost.

In order to replicate a more authentic real-world situation, we have opted for the open-source Hyperledger Fabric (HLF) [30] as the blockchain framework for our ecosystem. HLF is the right choice for this study because of its scalability features, minimal processing complexity, and flexibility to easily integrate different consensus mechanisms. In addition, many validators have the ability to process the transactions concurrently. Furthermore, the processing efficiency is improved by employing the rapid RAFT consensus approach in the ordering service [31]. RAFT is simple to configure, possesses the capacity to distribute governance, and is capable of implementing Byzantine Fault Tolerance (BFT). By employing the RAFT algorithm, HLF is able to achieve a significant throughput of transactions per second and guarantee a high level of scalability. HLF has been utilized as a blockchain platform for IoT systems in many research studies [32, 33]. Furthermore, all HLF applications are executed within Docker containers [34]. This setup ensures a clear separation between the application program and the underlying physical resources. The purpose of isolating the containers is to guarantee the security of the application.

A comprehensive assessment of the suggested model's performance has been conducted using a testbed built on the HLF platform. HLF introduces the concept of *ChainCode*, which represents smart contracts developed using the *Javascript* programming language. The testbed is equipped with a 100 Mbps Ethernet connection and an Intel Core i9-9750H CPU with 16GB of RAM. HLF nodes are implemented with the Docker platform, which operates on Ubuntu 22.04. To be more precise, the RAFT consensus algorithm is implemented using the latest version of HLF.

4.1. Trust Evaluation

Initially, we assess the performance of our system using one shard (a single channel) and then progressively increase the transactions per second (TPS) to determine its capacity. In our tests we use the parameters in Table 2.



```
Algorithm 2 SCoTMan_TrustUpdate
  PENDING_J ← Empty
  for feedback ∈ PENDING_FEEDBACK do
    INTR ← find interaction of feedback
    if found INTR then
      j ← target(INTR)
      insert(INTERACTIONS[j], INTR)
      insert(PENDING_J, j)
    end if
    if |INTERACTIONS[j]| > k then
      sort INTERACTIONS[j] by rater's GlobalTrust(GT)
      discard last element of INTERACTIONS[j]
    end if
  end for
  for j ∈ PENDING_J do
    Compute Direct Trust
    if |INTERACTIONS[j]| < k/2 then
      Compute Similarity Metrics with friends of j
      Compute Indirect Trust using social metrics
    else
      Compute Indirect Trust using recommendations from
      INTERACTIONS[j]
    end if
    Compute Global Trust
  end for
```

Algorithm 2 Trust Update in SCoTMan

Table 2. Parameters for Evaluation

Parameter	*Value*	*Description*
PM	0.1, 0.3, 0.5	Percent of Malicious
N_User	40	Number of Users
N_Dev	400	Number of Devices
k	10, 20, ∞	Storage capacity of each user
TPS	10-600	Transactions per seconds
Orderers	2	Orderer nodes
Peers	2	Peer Nodes
Block Timeout	2 secs.	Max. wait time to close a BC block
Block Size	50	Max. number of transactions in a block
Shards	1-5	Number of Shards

As the rate at which transactions are sent grows, Figure 3a demonstrates that the model is capable of processing a maximum of 240 transactions per second before encountering failure or canceling the excess transactions. Based on our evaluation, we conclude that our testbed has the capacity to process a maximum of 240 transactions per second (TPS). However, by using shards we try to increase this limit. In Figure 3b illustrates the latency of transactions by contrasting the SCoTMan with the base blockchain, which is an empty *chaincode*. The latency is increasing after TPS = 150, but the amount of increase is such that the latency is still below 1 second before the hardline of 240 TPS.

To test the model for its functionality, we conducted experiments involving ballot-stuffing and badmouthing assaults to evaluate the resilience of our trust protocol against various sorts of nodes, with PM (Percent of Malicious) values set at 0.1, 0.3, and 0.5. An honest node in our tests continuously adheres to providing honest feedback. Conversely, an adversary node participates in deliberate actions such as ballot-stuffing and badmouthing to obtain an unfair advantage by delivering misleading feedback. The trust levels for a randomly selected honest node are depicted in Figure 3 for three different situations involving different PMs. To enhance comparability, the ground truth value was selected as 0.82. The graph illustrates the gradual convergence of the trust score of a trustworthy node over time.

At time 100, when the PM is set to 0.1, the trust value hits 0.79. The chart subsequently expands until it reaches the ultimate convergence value. In the PM=0.3 situation, the trust value is currently 0.77, while in the PM=0.5 scenario, it is 0.74. At time 400, the trust values recorded are 0.81, 0.77, and 0.76, respectively. These values demonstrate that, regardless of bias, SCoTMan has the ability to identify genuine behavior in these situations.

We have implemented a restriction on the maximum number of entries that each user can have for keeping the interaction data. This constraint enables us to restrict the overall storage required for executing the trust management procedures. We provide three options for cases: 10, 20, and an unlimited case. In these three scenarios, Figure 4 depicts SCoTMan's trust convergence. In all scenarios, it is evident that the average trust value for an honest node is approaching the ground truth level. However, under restricted conditions, the speed of convergence is slower compared to an unrestricted scenario. Figure 5 shows the success rate under various storage limit conditions. The findings indicate that even with a restriction of 10 entries, the success rate continues to increase, though it does not approach 100%. Under more relaxed circumstances, we are able to achieve a success rate of 100%.

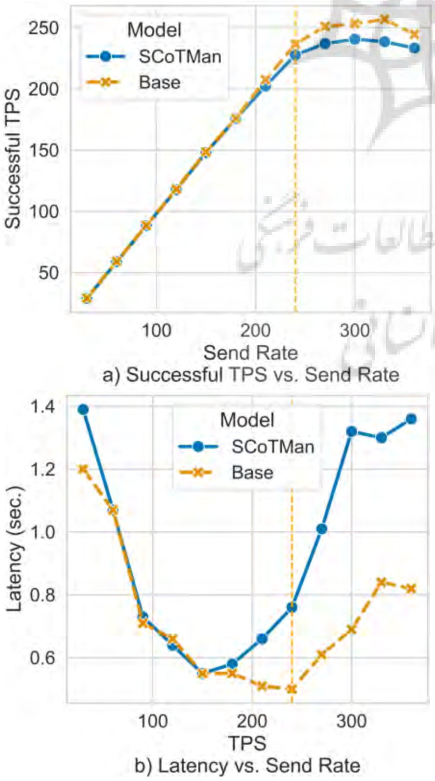


Figure. 3. SCoTMan capacity without Sharding

4.2. Sharding Results

By implementing state sharding in SCoTMan, we have evaluated the performance of our system using this technique. We have divided the node transactions into multiple channels (state storage in HLF). In our test, we have assessed the SCoTMan performance with 2 to 5 shards. In Figure 6, when there is no sharding (shard=1), the maximum number of accepted transactions per second (TPS) is limited to 270. This situation is similar to what we encountered when measuring the capacity of SCoTMan. With 2 shards, the capacity of the model increases to a maximum of 320 TPS. When testing SCoTMan with

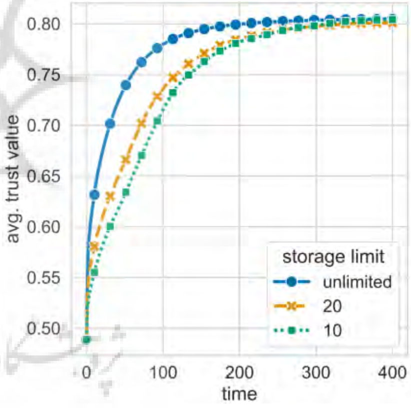


Figure. 4. Trust Convergence of a good node with limited storage

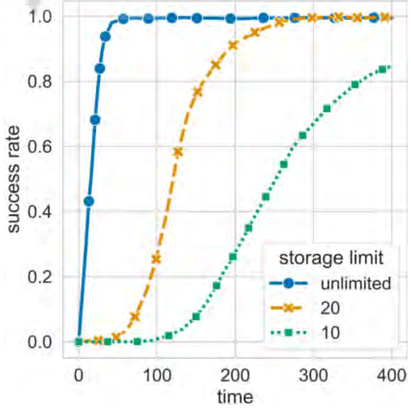


Figure. 5. Success Rate under storage limit conditions

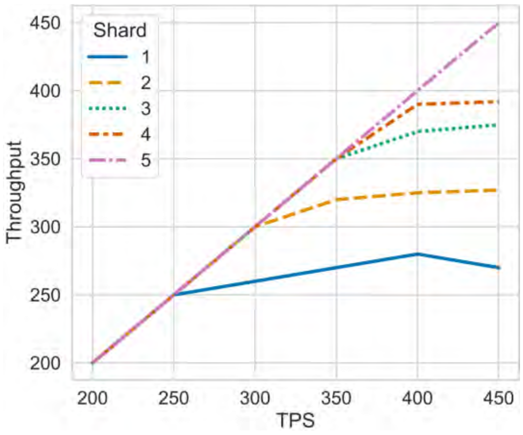


Figure. 6. SCoTMan throughput (TPS) based on the number of state Shards

3 state shards, the model reaches a maximum of 370 TPS, but the breaking point is around 350. As shown in the figure, with 4-state sharding, the maximum TPS reaches around 400. Furthermore, when utilizing 5 shards, SCoTMan can handle all the transactions imposed on it.

5. Conclusions

Our suggested solution, SCoTMan, is a scalable technique that utilizes blockchain technology for trust management in the SIoT. It has demonstrated acceptable performance among the trust management methods discussed in the literature. When there is little interaction history, the method uses social metrics among SIoT devices to identify inappropriate behavior, particularly for recommendations.

When a sufficient number of suitable recommenders were present, we determined the best recommenders by picking the top nodes that had prior interactions with the device and possessed greater levels of trust. In addition, by implementing state sharding in SCoTMan, we significantly enhanced the model's scalability by a near-linear scalability proportion.

We conducted a series of tests to assess SCoTMan's performance in terms of both blockchain and trust computation. To demonstrate the method's efficacy, the studies discovered multiple design factors. In addition, we carried out a thorough evaluation of the blockchain network's performance, showcasing its suitability for implementation in the Hyperledger Fabric (HLF). We conducted studies that demonstrate that SCoTMan can effectively achieve trust convergence speed and bias, even when confronted with real-world constraints on memory and CPU usage. This study reveals the successful deployment of scaling improvements in SCoTMan. The utilization of smart contracts and state sharding enabled these achievements. Furthermore, our experiments demonstrated that we enhance the performance of trust management methods.

In future research, we can concentrate on implementing more complex sharding techniques, such as network sharding or full sharding, by using the social connections between nodes. In addition, we can employ a punishing approach [35] to accelerate the decrease of trust values for malicious nodes. Adopting this method can enhance the model's resilience against complex attack scenarios and accelerate the trust convergence stage.

Declarations

Funding

This research did not receive any grant from funding agencies in the public, commercial, or non-profit sectors.

Conflict of interest

The authors declare that no conflicts of interest exist.

References

[1] A. Khanna and S. Kaur, “Internet of Things (IoT), applications and challenges: A comprehensive review,” *Wirel. Pers. Commun.*, vol. 114, pp. 1687–1762, 2020. https://doi.org/10.1007/s11277-020-07446-4

[2] I. Lee and K. Lee, “The Internet of Things (IoT): Applications, investments, and challenges for enterprises,” *Bus. Horiz.*, vol. 58, no. 4, pp. 431–440, 2015. https://doi.org/10.1016/j.bushor.2015.03.008

[3] I. R. Chen, J. Guo, and F. Bao, “Trust Management for SOA-Based IoT and Its Application to Service Composition,” *IEEE Trans. Serv. Comput.*, vol. 9, no. 3, pp. 482–495, 2016, doi: https://doi.org/10.1109/TSC.2014.2365797

[4] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, “TRM-IoT: A trust management model based on fuzzy reputation for internet of things,” *Comput. Sci. Inf. Syst.*, vol. 8, no. 4, pp. 1207–1228, 2011. https://doi.org/10.2298/CSIS110303056C

[5] X. Chen, J. Ding, and Z. Lu, “A decentralized trust management system for intelligent transportation environments,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 558–571, 2020. https://doi.org/10.1109/TITS.2020.3013279

[6] Z. Lin and L. Dong, “Clarifying trust in social internet of things (extended Abstract),” *IEEE 34th Int. Conf. Data Eng. (ICDE) 2018*, Paris, France, 2018, pp. 1825–1826. https://doi.org/10.1109/ICDE.2018.00270

[7] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, “A Secure Sharding Protocol For Open Blockchains,” in *the 2016 ACM SIGSAC Conference*, Oct. 2016, pp. 17–30. https://doi.org/10.1145/2976749.2978389

[8] Y. Liu, J. Wang, Z. Yan, Z. Wan, and R. Jantti, “A Survey on Blockchain-Based Trust Management for Internet of Things,” *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5898–5922, 2023. https://doi.org/10.1109/JIOT.2023.3237893

[9] P. Kasnesis, C. Z. Patrikakis, D. Kogias, L. Toumanidis, and I. S. Venieris, “Cognitive friendship and goal management for the social IoT,” *Comput. & Electr. Eng.*, vol. 58, pp. 412–428, 2017.

[10] L. Atzori, A. Iera, and G. Morabito, “From" smart objects" to" social objects": The next evolutionary step of the internet of things,” *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 97–105,

2014. https://doi.org/10.1016/j.compeleceng.2016.09.024

[11] C. Catalini and J. S. Gans, “Some simple economics of the blockchain,” 2016.

[12] C. Wright, “Sustainable Blockchain-Enabled Services : Smart Contracts,” in *IEEE International Conference on Big Data (BIGDATA)*, Boston, MA, USA, 2017, pp. 4255-4264, doi: https://doi.org/10.1109/BigData.2017.8258452

[13] E. Baccarelli, M. Scarpiniti, P. G. V. Naranjo, and L. Vaca-Cardenas, “Fog of social IoT: When the fog becomes social,” *IEEE Netw.*, vol. 32, no. 4, pp. 68–80, 2018. https://doi.org/10.1109/MNET.2018.1700031

[14] X. Wu, Y. Liu, J. Tian, and Y. Li, “Privacy-preserving trust management method based on blockchain for cross-domain industrial IoT,” *Knowledge-Based Syst.*, vol. 283, p. 111166, 2024. https://doi.org/10.1016/j.knosys.2023.111166

[15] G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, “Trust Management in Decentralized IoT Access Control System,” *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Toronto, ON, Canada, 2020, pp. 1-9,https://doi.org/10.1109/ICBC48266.2020.9169481.

[16] A. Moinet, B. Darties, and J.-L. Baril, “Blockchain based trust \& authentication for decentralized sensor networks,” *arXiv Prepr. arXiv1706.01730*, 2017. https://doi.org/10.48550/arXiv.1706.01730

[17] M. A. Azad, S. Bag, F. Hao, and A. Shalaginov, “Decentralized Self-Enforcing Trust Management System for Social Internet of Things,” *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2690–2703, 2020. https://doi.org/10.1109/JIOT.2019.2962282

[18] J. Lin, Z. Shen, and C. Miao, “Using blockchain technology to build trust in sharing LoRaWAN IoT,” *Proceedings of the 2nd International Conference on Crowd Science and Engineering*, 2017, pp. 38–43. https://doi.org/10.1145/3126973.3126980

[19] A. A. A. El-latif, B. Abd-el-atty, I. Mehmood, K. Muhammad, S. E. Venegas-andraca, and J. Peng, “Quantum-Inspired Blockchain-Based Cybersecurity : Securing Smart Edge Utilities in IoT-Based Smart Cities,” *Inf. Process. Manag.*, vol. 58, no. 4, p. 102549, 2021. https://doi.org/10.1016/j.ipm.2021.102549

[20] K. Croman *et al.*, “On Scaling Decentralized Blockchains: (A Position Paper),” in *International conference on financial cryptography and data security*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 106–125. https://doi.org/10.1007/978-3-662-53357-4_8

[21] Z. Avarikioti, A. Desjardins, L. Kokoris-Kogias, and R. Wattenhofer, “Divide & scale: Formalization and roadmap to robust sharding,” in *International Colloquium on Structural Information and Communication Complexity*, Cham: Springer Nature Switzerland, 2023, pp. 199–245. https://doi.org/10.1007/978-3-031-32733-9_10

[22] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, “A survey on the scalability of blockchain systems,” *IEEE Netw.*, vol. 33, no. 5, pp. 166–173, 2019. https://doi.org/10.1109/MNET.001.1800290

[23] H. Huang *et al.*, “Brokerchain: A cross-shard blockchain protocol for account/balance-based state sharding,” in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, London, United Kingdom, 2022, pp. 1968–1977. https://doi.org/10.1109/INFOCOM48880.2022.9796859

[24] M. Nitti, R. Girau, and L. Atzori, “Trustworthiness management in the social internet of things,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1253–1266, 2013. https://doi.org/10.1109/TKDE.2013.105

[25] A. Rouzbahani and F. Taghiyareh, “SCoTMan: a Scalable Smart Contract for Trust Management in Social IoT with Real-World Constraints,” *IEEE Access*, vol. 12, pp. 137836-137850, 2024. https://doi.org/10.1109/ACCESS.2024.3411581

[26] A. Rouzbahani and F. Taghiyareh, “A Trust-Aware Task Allocation Method Based on Blockchain for The Internet of Things,” in *2022 8th International Conference on Web Research (ICWR)*, Tehran, Islamic Republic of Iran, 2022, pp. 156-161. https://doi.org/10.1109/ICWR54782.2022.9786257

[27] A. Rouzbahani and F. Taghiyareh, “Trust-independent Blockchain for self-interested Multiagent Systems through Smart Contracts,” *10th Information and Knowledge Technology Conference (ICIKT2019)*, Tehran, Iran, 2019. https://civilica.com/doc/982314/.

[28] C. Zhang and H.-A. Jacobsen, “Towards Dependable, Scalable, and Pervasive Distributed Ledgers with Blockchains,” in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 1337–1346. 10.1109/ICDCS.2018.00134

[29] A. Josang and R. Ismail, “The beta reputation system,” in *Proceedings of the 15th bled electronic commerce conference*, 2002, vol. 5, pp. 2502–2511.

[30] E. Androulaki *et al.*, “Hyperledger fabric: a distributed operating system for permissioned blockchains,” in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15. https://doi.org/10.1145/3190508.3190538

[31] C. Copeland and H. Zhong, “Tangaroa: a byzantine fault tolerant raft.” Tech. rep, 2016.

[32] S. Pešić, M. Radovanović, M. Ivanović, M. Tošić, O. Iković, and D. Bošković, “Hyperledger Fabric Blockchain as a Service for the IoT: Proof of Concept,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11815 LNCS, pp. 172–183, 2019. https://doi.org/10.1007/978-3-030-32065-2_12

[33] H. Liu, D. Han, and D. Li, “Fabric-iot: A Blockchain-Based Access Control System in IoT,” *IEEE Access*, vol. 8, pp. 18207–18218, 2020. https://doi.org/10.1109/ACCESS.2020.2968492

[34] D. Merkel and others, “Docker: lightweight linux containers for consistent development and deployment,” *Linux j*, vol. 239, no. 2, p. 2, 2014.

[35] M. Ebrahimi, M. H. Tadayon, M. S. Haghighi, and A. Jolfaei, “A quantitative comparative study of data-oriented trust management schemes in Internet of Things,” *ACM Trans. Manag. Inf. Syst.*, vol. 13, no. 3, pp. 1–30, 2022. https://doi.org/10.1145/3476248

Amin Rouzbahani received his B.S. and M.S. degrees in Computer Engineering from Sharif University of Technology. He has obtained his Ph.D. from the School of Electrical and Computer Engineering, College of Engineering, at the University of Tehran, with a focus on Trust Management in IoT ecosystems. His Ph.D. thesis addressed Trust Management in Internet of Things through the use of Blockchain technology, one of the many research projects he has been working on in recent years.

Fattaneh Taghiyareh obtained her B.Sc. and M.Sc. degrees from Sharif University of Technology in 1990 and 1994. In 2000, she obtained her Ph.D. in computer engineering from Tokyo Institute of Technology. She has been employed at the University of Tehran's School of Electrical and Computer Engineering since 2001 and is currently an Associate Professor of computer engineering, software, and information technology. She presently works as a member of the Software Engineering and Information Technology Department and is in charge of the Multi-Agent System Laboratory, which she founded in 2005, and the eLearning Laboratory, which she founded in 2012. She was the previous director of the University of Tehran's Technology Incubator as well as the former manager of the department of information technology. She founded The IT Foundation Laboratory in 2006, and she is currently in charge of it. She is the author of many publications, including journal and conference articles. Her areas of interest in research are multi-agent systems and collaborative environments for e-learning. Her current work applies ontology to semantic web and examines the junction of Web-Services, customization, and adaptive learning management systems. She is on the International Journal of Information and Communication Technology's Editorial Board.