Jayasinghe, U, Lee, GM, MacDermott, Á and Rhee, WS

TrustChain: A Privacy Preserving Blockchain with Edge Computing

http://researchonline.ljmu.ac.uk/id/eprint/10912/

Article

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 2014697, 17 pages
https://doi.org/10.1155/2019/2014697

Research Article

TrustChain: A Privacy Preserving Blockchain with Edge Computing

Upul Jayasinghe,[1] **Gyu Myoung Lee**,[1] **Áine MacDermott**,[1] **and Woo Seop Rhee**[2]

[1] *Department of Computer Science, Liverpool John Moores University, Liverpool, UK*
[2] *Department of Information and Communication Eng., Hanbat National University, Daejeon, Republic of Korea*

Correspondence should be addressed to Gyu Myoung Lee; g.m.lee@ljmu.ac.uk

Received 15 March 2019; Accepted 16 June 2019; Published 8 July 2019

Guest Editor: Khalid Elgazzar

Recent advancements in the Internet of Things (IoT) has enabled the collection, processing, and analysis of various forms of data including the personal data from billions of objects to generate valuable knowledge, making more innovative services for its stakeholders. Yet, this paradigm continuously suffers from numerous security and privacy concerns mainly due to its massive scale, distributed nature, and scarcity of resources towards the edge of IoT networks. Interestingly, blockchain based techniques offer strong countermeasures to protect data from tampering while supporting the distributed nature of the IoT. However, the enormous amount of energy consumption required to verify each block of data make it difficult to use with resource-constrained IoT devices and with real-time IoT applications. Nevertheless, it can expose the privacy of the stakeholders due to its public ledger system even though it secures data from alterations. Edge computing approaches suggest a potential alternative to centralized processing in order to populate real-time applications at the edge and to reduce privacy concerns associated with cloud computing. Hence, this paper suggests the novel privacy preserving blockchain called TrustChain which combines the power of blockchains with trust concepts to eliminate issues associated with traditional blockchain architectures. This work investigates how TrustChain can be deployed in the edge computing environment with different levels of absorptions to eliminate delays and privacy concerns associated with centralized processing and to preserve the resources in IoT networks.

1. Introduction

Internet of Things (IoT) data is becoming one of the most valuable assets in today's data-driven digital economy as it leads to developing many business models providing numerous ubiquitous and intelligent services [1]. However, these data contain sensitive personal information and can reveal the identity of the associated stakeholders if a proper privacy preserving mechanism is not in place [2]. For example, a malicious actor who has access to someone's personal information (such as their address, date of birth, nationality, bank account number, or private e-mail address) can use his identity to gain a financial advantage or obtain other benefits in the other person's name. The person whose identity has been assumed may suffer adverse consequences, especially if they are responsible for the perpetrator's actions. Hence enforcing strong privacy preserving techniques and regulations is necessary. From a regulatory point of view, a new data protection act called General Data Protection Regulation (GDPR) was introduced by the European Union (EU) in May 2018 to control the unnecessary usage of data, empowering users' right on their data [3]. It rigorously discusses the importance of a "privacy by design" concept which essentially calls for privacy to be considered throughout the whole engineering process.

Worryingly, features of the IoT networks such as their distributed architectures, massive scale, and scarcity of the resources with respect to processing power, storage capacity, bandwidth, etc., do not provide a safe platform for privacy preserving applications. Further, traditional applications of IoT networks are designed in such a way that data management functions, i.e., data collection, data storage, data processing, data sharing, and data destruction, are executed in a centralized fashion, neglecting the distributed nature of IoT devices. This approach has proved to lead to significant delays and traffic congestion when used for delay sensitive applications and thus cannot satisfy the requirements of ultra-low delay sensitive IoT applications, such as a real-time computer vision for smart city security. It not only heightens the issues associated with scalability and latency, but also makes IoT nodes more vulnerable for privacy and security threats, including lack of control over personal data hence unauthorized user profiling and identity theft, fake knowledge propagation, network eavesdropping, illegal invasion, and denial of service (DoS) attacks.

On the other hand, a hierarchical edge computing architecture can resolve the issues associated with centralized architectures by pushing data pipeline functions towards the edge of the IoT networks depending on the resource availability and application requirements [4, 5]. Deployment of such an architecture is beneficial in building an ecosystem involving content providers, application developers, network equipment vendors, third-party partners, and middleware providers and thereby improving the end-user experience dramatically due to powerful and energy efficient computing power at hand, low latency, mobility, location, and context-aware support for IoT applications [6–8]. However, edge computing alone cannot support the safeguarding of the privacy of the stakeholders, as it introduces a new set of vulnerabilities due to multiple attack surfaces, closeness to sensitive data generators, heterogeneity of the device resources, the scale of the network, and difficulty of assessing the trustworthiness of participating stakeholders [9].

The fundamental concept behind blockchain technology provides a promising approach to establish a healthy interaction among untrustworthy and unknown entities, while supporting the distributed nature of IoT, eliminating the need of a central authority as in cloud computing architectures [10, 11]. The main technology of the blockchain lies behind the use of an immutable public record of data called "*public ledger,*" shared among all participants. The ledger consists of blocks of data which are linked with each other by a cryptographic hash key and the process of linking is termed "*Proof of Work*" (PoW). However, adopting the blockchain technology as it is in the IoT environment is quite challenging due to the exhaustive computation power required to solve PoW puzzles with resource-constrained IoT devices. The delay associated with the mining process is not suitable for real-time IoT applications, in addition to scalability issues associated with blockchain, and further overhead created by blockchain consensus algorithms.

Motivated by the facts stated above, we first propose a novel variation of blockchain called TrustChain which removes the need for PoW by utilizing trust evaluation concepts discussed in our previous work [12–14]. In general, the concept of trust can be seen as a metric that is used to evaluate stakeholders in consideration of mutual benefits, coordination, and cooperation. Perception of trust is often achieved with direct observations, through experience and based on the beliefs and opinions of others who are around it as we discussed in [14]. This paper also discusses a privacy perceiving edge computing architecture based on the concepts of ROOF (ROOF: Real-Time Onsite Operations Facilitations), Fog, and Cloud computing concepts [15]. Then, we further extend the idea of TrustChain based on the permissioned blockchain concepts to match with the requirements at each layer of the proposed distributed architecture. We use a smart city use case to demonstrate the proposed ideas in all three cases above.

The structure of this paper is as follows: Section 2 presents the concepts and underlying principles of traditional blockchain technologies and trust evaluation in general to support the proposed ideas in the following sections. Section 3 focuses on the details of TrustChain and underlying mechanisms compared to typical blockchains. Section 4 discusses the importance of the proposed TrustChain when realizing privacy preserving and trustworthy services in a distributed environment, and Section 5 concludes the paper.

2. Preliminaries

In this section, we briefly introduce the concepts and underlying technology associated with traditional blockchains and trust evaluation. In the context of TrustChain, we utilize these concepts to develop a novel blockchain technology to remove the issues related to privacy and efficient use of resources in a decentralized setting like the IoT.

2.1. Blockchain Overview. Blockchain technology is one of the highly researched topics in the recent years, due to its distributed and immutable data storage mechanism enabling applications in almost any area including banking, supply chain, and other transaction networks like IoT. The concept of blockchain was first introduced by Satoshi Nakamoto [16] as the fundamental technology of the digital cryptocurrency called Bitcoin. The use of blockchain in public and distributed ledgers for Bitcoin transactions made it the first cryptocurrency not only to transact digital money in a securely and inexpensive manner but also to resolve the long-standing problem of "double spending" without the need for a trusted and powerful third-party. By nature, blockchain technology is inherently resistant to data modification due to its public ledger and the consensus mechanism called PoW. Once recorded, data in any given block cannot be altered retroactively as this would invalidate all hashes in the previous blocks in a blockchain and break the consensus agreed among nodes voiding the blockchain.

2.1.1. Blockchain Architecture. Blockchain can be basically considered as a chain shaped data structure in which a chain of blocks are connected with each other through an address pointer based on a hash value; i.e., blockchain is a shared, decentralized, and distributed state machine. This means that all nodes independently hold their own copy of the blockchain, and the current known "state" is calculated by processing each transaction in order as it appears in the blockchain. Each block of a blockchain typically contains six parts; hash of the previous block, nonce ("number used once"), the hash of the current block, Merkle root (hash of multiple transactions), timestamp, and transaction data as shown in Figure 1.

In the context of Bitcoin, transactions generally consist of the sender's address, recipients address, and the value. However, depending on the application this can vary. The

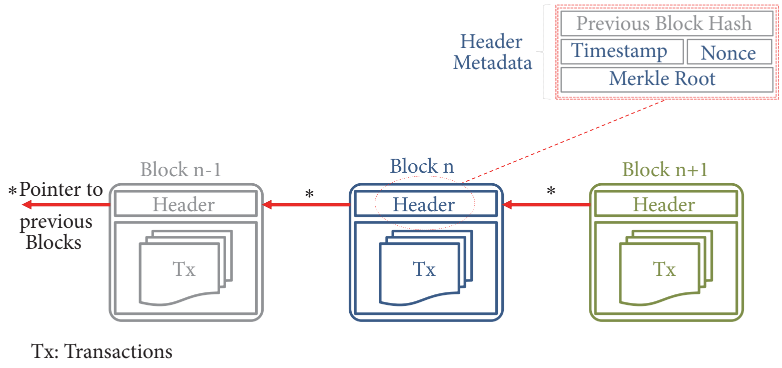


Figure 1: A typical structure of a blockchain.

header of each block contains a set of metadata that helps to validate each block and link to previous blocks in the public ledger. Having a public ledger means that the data and access to the system is available to anyone who is willing to participate (e.g., Bitcoin, Ethereum, and Litecoin blockchain systems).

However, depending on the application requirements, the structure of the blockchain can be designed in either a more centralized or a more decentralized manner. In this regard, private blockchain architectures are more centralized as they are controlled by a centralized authority that controls the access to the blockchain network. Similar to private blockchains, consortium blockchains are controlled by a set of selected nodes rather than one specific organization hence a suitable candidate for IoT applications.

2.1.2. Consensus and Mining. Mining is the process that validates the blocks created by the blockchain nodes and attaches them to the genesis blockchain. However, the process is a computationally intensive procedure due to the cryptographic puzzle that needs to be solved in order to validate the block. In Bitcoin networks, this process is called PoW in which miners need to find a suitable nonce that gives a unique hash key for each block. Usually, this key is 256 bits long; hence breaking the key is extremely hard without controlling 51% of the total computing power available among miners. Miners receive a reward when they solve the complex mathematical problem. There are two types of rewards: new bitcoins or transaction fees. The overall process of bitcoin mining is shown in Figure 2.

The consensus is the process that allows every node in the blockchain network to agree upon connecting the new block to the chain. This involves the mining process as well as other rules including the maximum allowable mining time, how to treat blockchain divergence, signing transactions into blockchain block, rewarding miners, and choosing miners. Other than famous PoW consensus protocols, there are other alternatives including the Byzantine Fault Tolerance algorithm (BFT) [17], the Proof-of-Stake algorithm (PoS) [18], and Delegated Proof-of-Stake algorithm (DPoS [19]). [20]. Each of these algorithms is designed to achieve certain agreement among nodes depending on the application area of use while enabling unique features like minimum resource requirements, immunity of the protocol, easiness of access control, and privacy. A comparison of typical consensus algorithms is shown in Table 1.

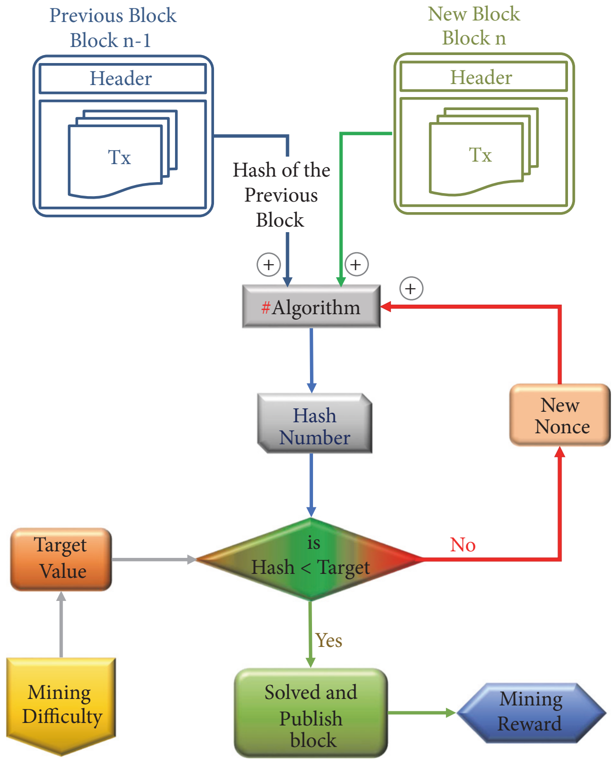


Figure 2: Bitcoin mining process.

2.1.3. Smart Contracts. One of the significant parts of a blockchain is the smart contract entity as it bridges the gap between prosumers in terms of executing their preagreed rules and conditions without a centralized authority; i.e., it connects service providers with respectable consumers or connects blockchain service applications. The smart contract codes are otherwise known as executing contracts, blockchain contracts, or digital contracts that are stored in the ledger. Transactions can invoke smart contract functions. They not only define the rules and penalties around an agreement in the same way that a traditional contract does, but also automatically enforce those obligations. The concept

Table 1: A comparison of typical consensus algorithms [21].

Property	PoW	PoS	BFT	DPoS
Identity Management	Open	Open	Permissioned	Open
Energy Saving	No	Partial	Yes	Partial
Immunity	25% of the Computing power	51% of Stake	33.3% of faulty replicas	51% of Validators
Example	Bitcoin [16]	Peercoin [22]	Hyperledger Fabric [23]	Bitshares [24]

of smart contract was first introduced by Nick Szabo [25] and is widely used in many popular blockchain versions, like in Ethereum [26].

2.2. Trust Evaluation Overview. In general, trust represents a measure of confidence that indicates how much an actor or entity will behave in an expected manner in a situation, and how much an actor is willing to rely on the actions of another actor or party in the future. The concept of trust is an abstract notion, with different meanings depending on both participants and scenarios and influenced by both measurable and nonmeasurable factors. Hence, inconsistency in trust definitions might lead to difficulty in establishing a common, general notation that holds regardless of personal dispositions or differing situations. To avoid such ambiguity, we define trust as "*a qualitative or quantitative property of a trustee measured by a trustor for a given task in a specific context and in a specific time period*" as in [14]. In the context of IoT, trust can be identified as a feature that affects the appetite of an object to consume a particular service or product offered by another. This can be observed in everyday life, where trust decisions are made. When purchasing a specific product, we may favor certain brands due to our trust that these brands will provide excellent quality compared to unknown brands. Trust in these brands may come from our knowledge, experience of using their products, or their reputations, which are perceived by other people who bought items and left their opinions about those products.

Although the significance of trust in our physical world is as important as it is in the digital environment, building trust and confidence in the latter is much more difficult. This is due to our inability to have a physical view of an object, unlike in our physical world, where we can view the building of the bank, observe its safe deposits, meet the bank personnel, etc. Another issue with trust is that it is difficult to quantify the exact trustworthiness value of an object. This is even harder when each object has different interpretations and perceptions of the term "trustworthy." Therefore, they may assign different trustworthiness values to the same provider or the service. For example, a service consumer assigns "very trustworthy" to the provider for a transaction that he has performed. However, another consumer assigns "untrustworthy" for a similar transaction from the same provider. These differences further increase the difficulty in determining the exact trustworthiness of an object.

Despite difficulties in trust evaluation, it shows quite a significant potential towards eliminating risks related to privacy and preserving the integrity of the interactions. Hence, we borrow a definition for trust and the trust model from our previous work in [12–14] to formulate the foundation in this work. We identify three Trust Metrics (TMs) to identify, evaluate, and create trust relationships among objects in the IoT endowment, namely, knowledge, experience, and reputation. Each TM is a collective representation of several Trust Attributes (TAs) and each TA represents the trustworthiness feature of a trustee as shown in Figure 3.

The knowledge TM covers all aspects of direct trust evaluations, which provide a perception about a trustee before and during an interaction. To make this possible, it must provide relevant data to the trustor for its assessment. If a data feature can be represented using a quantitative measurement, then the result is a numerical value in a certain range. As an example, social relationships like colocation and cowork, credibility factors like cooperativeness, time-dependent features like the frequency and duration of interactions, and spatial distribution of relevant trustees compared to the trustor can be used as direct trust measurements. The main purposes of trust assessments are to facilitate more intelligent decision-making and task delegation. In this regard, we further elaborate two further metrics, which come under the knowledge TM as nonsocial TMs and social TMs. In nonsocial trust, the idea is to find whether the trustor can rely on physical or cyber objects, and social trust determines whether a trustor can depend on other social objects.

After acquiring enough evidence about trustees through the knowledge TM, the trustor can initiate collaborations with selected trustees based on the perception that the trustor has already obtained. However, the result of these interactions might differ from the perception and hence it is critical to keep a record of each individual experience to be used in future interactions. For instance, the experience might be feedback from consumers after each transaction (as used in many e-commerce systems), just a Boolean value (0/1) indicating whether a service transaction successfully operates (as in some reputation-based trust systems), etc. Then, by accumulating these experiences over time in relation to the corresponding contexts, tasks, and times, the trustor can build up additional intelligence compared to the knowledge TM. To further enhance the perception of the trustor, other objects can share their experience in using the trustee, upon a request by the trustor, which we identify as reputation or the global opinion of the trustee. As an example, we have created a nonbias PageRank based model to calculate the reputation values of trustees in a distributed network

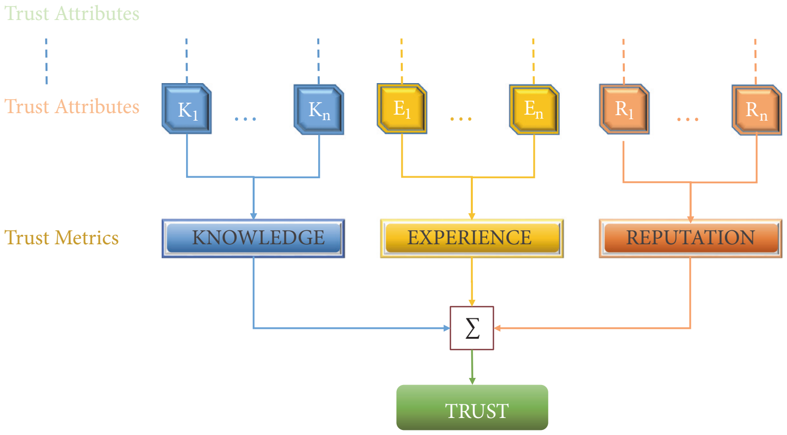


Figure 3: Accumulation of Trust Attributes (TAs) towards Trust Metrics (TMs) and formation of a trust value.

as in [27]. In summary, the experience TM is a personal observation considering only interactions from a trustor to a trustee, whereas the reputation TM reflects the global opinion of the trustee.

According to the model in Figure 3, the first step of the trust evaluation is to estimate relevant TMs depending on the application. However, this information is not readily available and therefore attributes which define these TMs must be obtained. There are numerous methods available to estimate these TAs ranging from numerical methods, probabilistic methods, belief theory, to machine learning (ML) methods. In simple terms, the mathematical approach to find the trust value trustor "i" and trustee "j" can be represented as in

$$\begin{aligned} K_{ij} &= \alpha_1 K_1 + \alpha_2 K_2 + \ldots + \alpha_n K_n \\ E_{ij} &= \beta_1 E_1 + \beta_2 E_2 + \ldots + \beta_n E_n \\ R_{ij} &= \gamma_1 R_1 + \gamma_2 R_2 + \ldots + \gamma_n R_n \\ Trust_{ij} &= \theta_1 K_{ij} + \theta_2 E_{ij} + \theta_n R_{ij} \end{aligned} \tag{1}$$

where α, β, γ, and θ are weighting factors that normalize each metric in between 0 and 1. K_x, $E_{x,}$, and R_x represent the TMs: knowledge, experience, and reputation, respectively.

3. TrustChain Platform

In this section, we present the underlying mechanisms involved with the TrustChain platform which is, an alternative initiation of traditional blockchain in terms of preserving privacy, distributed nature, and the computational resources. In contrast to many issues associated with traditional blockchains as discussed in Section 2.1, the TrustChain is powered with several advantages including

(i) efficient mining scheme that does not depend on computing resources but mutual agreements and trust among them;

(ii) significantly small mining delay compared to PoW;

(iii) enhanced scalability due to associated distributed storage mechanism;

(iv) improved privacy due to intelligent encryption algorithm running inside the TrustChain to hide/minimize the personal data exposure;

(v) compatibility with IoT business models due to permissioned nature of TrustChain, while maintaining the distributed and autonomous decision-making capability without relying on a central validator who has control over each node;

(vi) interoperability among several TrustChains as well as with external traditional blockchain due to the application of smart contracts inside TrustChains.

In order to provide such competency over traditional rivalries like Bitcoin, Ethereum, and Hyperledger, TrustChain is equipped with unique services which provide the aforementioned properties. Hence, we propose several services that must be combined with traditional blockchain service as shown in Figure 4.

Furthermore, to tackle the issues related to privacy in traditional blockchain networks, our proposed TrustChain service should be designed in compliance with data and privacy regulation standards like GDPR [3]. While the principles of data accountability and transparency have previously implicit requirements of data protection law, GDPR further extends the requirements of authorities by introducing explicit provisions that promote data accountability and governance to protect the privacy of personal data. GDPR defines three participant roles, namely, Data Subject (DS), Data Controller (DC), and Data Processor (DP) and specifies their associated obligations under EU data protection law. In this regard, we adopt the permissioned blockchain concept in order to design the TrustChain platform as it allows the prosumers to control the data visibility through access control and consensus mechanisms. Also, the use of smart contracts enables DSs, DCs, and DPs to impose and to negotiate consent rules for finalizing an agreement on data usage as legal grounds by means of a usage control language. We believe our proposed TrustChain platform is a promising approach for developing a secure and trusted data management in Smart Cities that fully comply with GDPR.

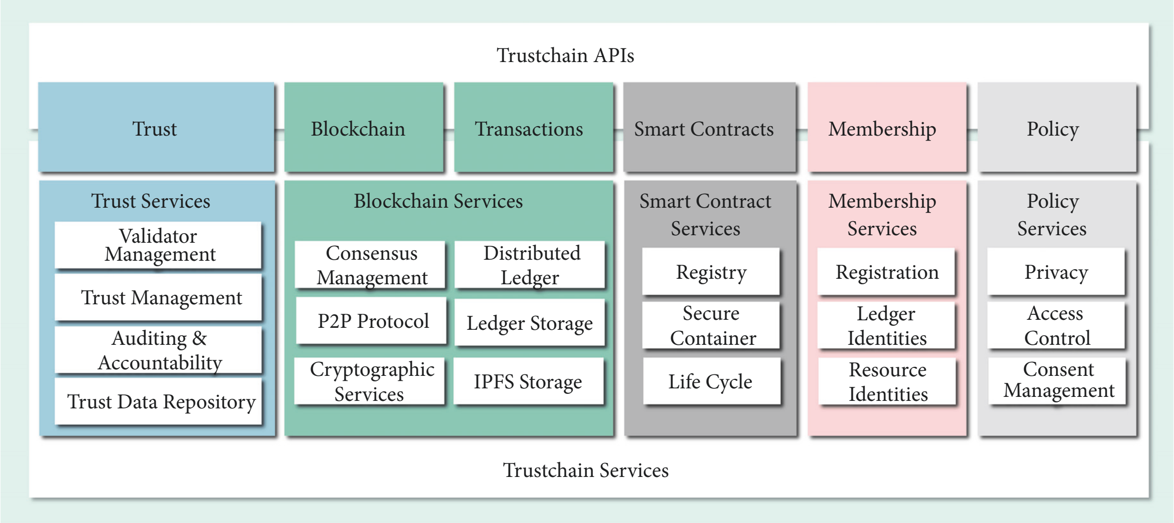


Figure 4: Core services of TrustChain platform.

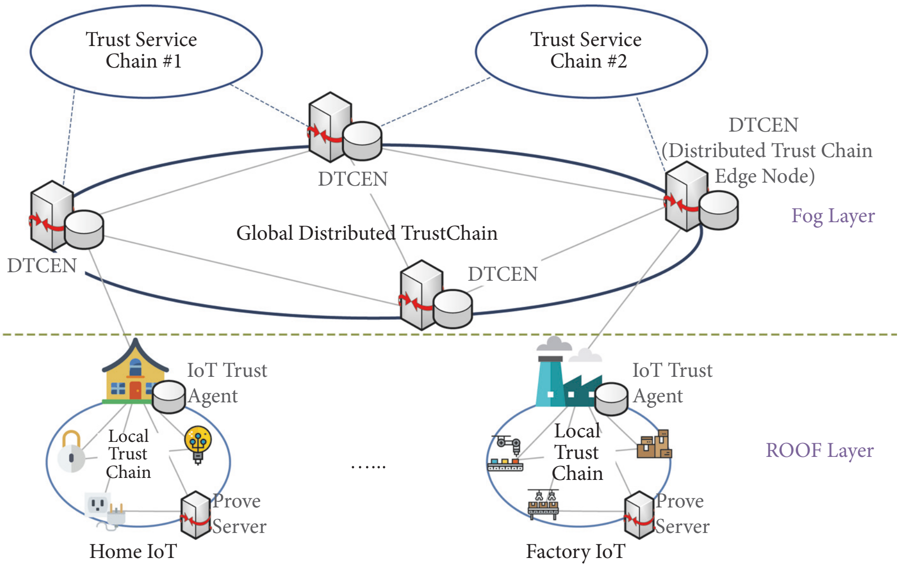


Figure 5: TrustChain Services in the global distributed TrustChain and local TrustChain Networks.

3.1. Trust Service. Formally, we define trust as in Section 2.2 and use it as a service in this section to empower each aspect of TrustChain to design a lightweight consensus algorithm as investigated in Section 3.2, administer smart contracts as described in Section 3.3, manage membership and policies as in Section 3.4, and, importantly, protect the privacy of the users in compliance with GDPR legislation as described in this section. The trust service in a Distributed TrustChain Edge Node (DTCEN) network is implemented as a logical Trust Service Chain (TSC) as shown in Figure 5. Then based on the requirements of TrustChain platform, trust service is provided on demand to both local TrustChain stored in ROOF node/IoT Trust agents (such as a home router) and global distributed TrustChain stored in the DTCEN network. Additionally, the DTCEN network is allowed to keep a public ledger alongside private ledger depending on the application scenarios, user consents, and privacy policies of participating nodes. When there is a requirement to validate a block, engage with smart contract service, or manage the membership of the stakeholders, TSC will be triggered among the relevant actors to evaluate trust based on the methods discussed below and thereby assist the obligation in place.

Essentially, the trust service is a result of various trust evaluation models and management functionalities. It is important to note that the equations in (1) represent the overall processes of TM/TA segmentation, evaluation, and aggregation which is needed to establish trust service irrespective of the area of application and only the TAs which represent the features of a given situation are different to each other. For example, the trust evaluation model based on features of an entity, integrity of data, and associate privacy requirements are shown in Figure 6.

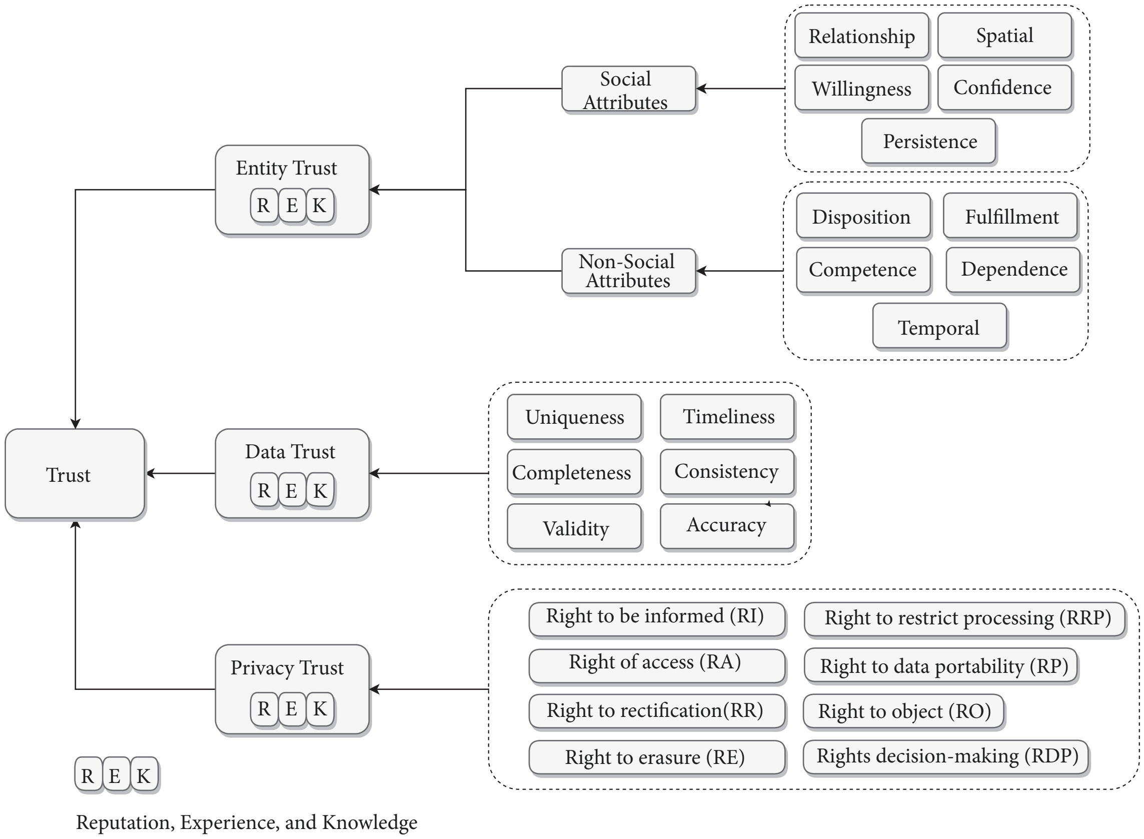


Figure 6: Trust evaluation based on object trust model, data trust model, and privacy trust model.

However, having an appropriate trust model is not enough in evaluating trust and hence a Trust Management and Evaluation (TME) module addresses the whole process from data collection to trust evaluation as shown in Figure 7. To support such processes, the TME module is equipped with several important submodules, Trust Agent (TAG), Trust Data Access Object (TDAO), IPFS enabled Data Repository (DR), TA Extractor (TAE), Trust Information Analysis (TIA), AI Engine (AIE), Trust Modelling Algorithm (TMA), and Trust Lifecycle Management module (TLM). These modules will perform one or more tasks at a time to evaluate trust.

TAG basically collects appropriate trust data from all the sources, including DC, DP, and DS objects that provide designated services, a trust broker who coordinates with external reputation systems, and TME API that provides an interface to extract data from external objects, environment, and other repositories in order to determine relevant TAs for trust evaluation. Generally, TAG works similarly to the client-server application, in which objects and the TME module change their role depending on the direction of data flow. The data could be either information obtained directly from relevant parties, experience, or opinions of objects as reputation or feedbacks from/to other objects, applications, or services. Once the TAG inside the TME module acquires the data, it will be stored in the local repository to be used by other submodules inside the TME module. To facilitate easy access to data in a distrusted setting, an Inter-Planetary File System (IPFS) based data repository is used in addition to local storage at each node.

Having obtained the necessary information by TAG, the next step is to assess the TA with help from TAE and AIE. Once TAs are assessed, they will be processed by TIA in combination with TMA to aggregate all the TAs based on the techniques like numerical, statistical, ML, or ensemble approaches as discussed in [12–14]. The TMA should work together with the TAE and TIA to find the best possible metrics for each model. Some models will combine the metrics according to pre-defined rules or policies, while others will generate these rules and policies dynamically to suit the situation in the best possible way. Finally, estimated trust information is transmitted towards the relevant party for appropriate decision-making processes.

3.2. Blockchain Service. The blockchain service enables prosumers to interact with each other without a centralized authority but with a community of peers in the form of a peer-to-peer (p2p) network, in which trust is not placed in an individual, but rather distributed across the entire population. Hence, no one can unilaterally take actions on behalf of the community and change the interactions. Moreover, the distributed nature of blockchains enables both horizontal and vertical service provision depending on the application requirement. As shown in Figure 8, different layers of TrustChains are established by their responsible computing authority. A permissioned TrustChain at the Fog

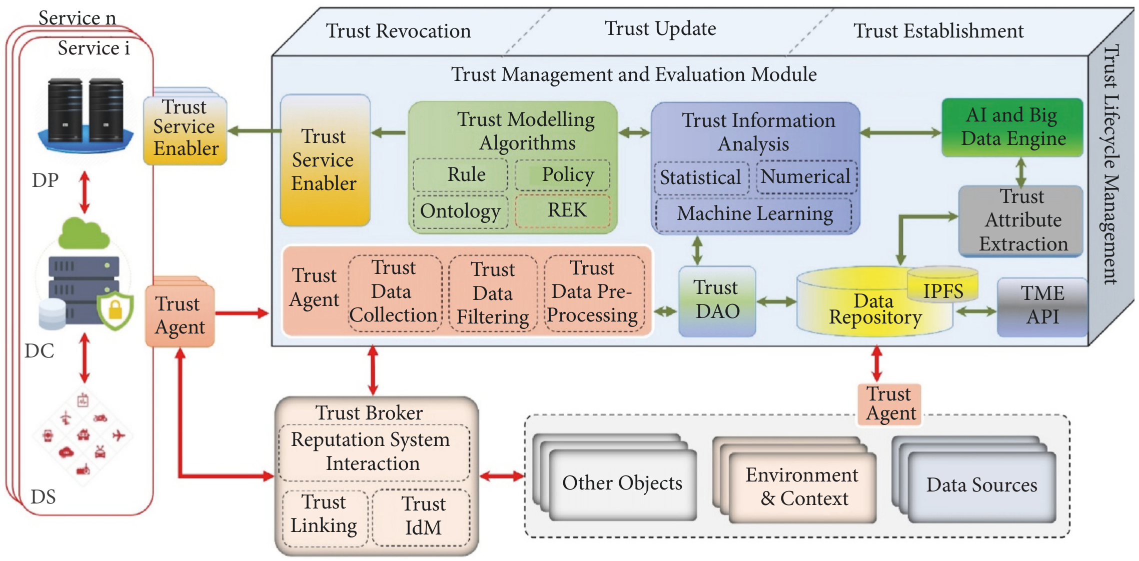


Figure 7: Trust management and evaluation module.

and ROOF layers are implemented, as service providers at the Fog and ROOF layers have better control over the network. However, there are no TrustChains occupied in the cloud layer due to the p2p nature of the TrustChains and privacy issues related to centralized processing. To explain the applicability of TrustChain in a real-world scenario, a smart city use case has been utilised. In this scenario, the bottom layers represent IoT nodes like sensors, smartphones, and tablets and ROOF agents such as hubs, routers, and home servers. To facilitate localized services which demand real-time decision-making capabilities like in emergency services and energy management services. TrustChains in ROOF will work together through smart contracts among them. Due to the lightweight attribute of the consensus protocol suggested below, even a ROOF node has the ability to validate new blocks and add them to the genesis TrustChain improving the overall performance of the IoT local network.

At the area level, a collection of distributed servers at the Fog layer can establish a permissioned blockchain based on the data coming from ROOF nodes, as well as data stored in the DTCENs to facilitate near real-time services. Further, depending on the computational power available to Fog nodes, it is possible to deploy artificial intelligence and data mining techniques to obtain extra knowledge about a situation to respond to it correctly. Global TrustChain at Fog layer will act as a control layer in such cases to orchestrate the services required by area level applications in a smart city. In order to improve the interoperability among several TrustChains, decentralized exchange (DEX) will act as a broker as shown in Figure 8. A detailed implementation of DEX based on the smart contract service is discussed in Section 3.3.

With respect to data management, it is not required to send all the data through the blockchain and depending on the user consent some of the data can be directly transmitted towards the upper layer for immediate processing as denoted with "Direct Communication" in Figure 8. Furthermore, smart contracts can be used to negotiate between actors to trigger certain services stored in the TrustChain or to find the services stored in a separate database and generate a combined result. The scenario applies to Fog and cloud level in a similar manner and the concept of smart contract can also be used as a DEX to communicate in between layers when taking higher level decisions such as at the cloud layer. Additionally, it is beneficial to identify different types of nodes depending on their capabilities on forming a TrustChain in terms of storage and verification as shown in Table 2. Full nodes are capable of storing both the full TrustChain and the verification by triggering the consensus protocol discussed later in this section. Heavy nodes are mostly confined with limited storage and hence only stores TrustChain below the Fog layer, and if further processing is required it will initiate a smart contract with cloud-based nodes to establish full TrustChain depending on the application. Light nodes are only capable of storing TrustChain headers and often not able to add any blocks to the chain as their resources are limited for performing verification. Data issuers are simply the IoT devices and sensors. They can choose to store data in TrustChains which are handled by upper layers or directly send the data towards upper layers without depositing the data on TrustChain for fast processing after evaluating the privacy requirements and consents from users via the trust service. It is important to establish smart contracts between different types of TrustChains to ensure privacy by design concept. Hence, different types of distributed ledgers are proposed to store data, smart contracts, and cryptocurrency separately and connected by DEXs.

Further, to limit the usage of TrustChain for data storage, it is possible to integrate TrustChain with parallel distributed architecture [28] that separates the data layer from the control layer using TrustChain. Then the data generated by each node at each layer is stored in a local database and TrustChain only holds the reference to data that serve as the identifier to verify the correctness and integrity of data. This enables TrustChain

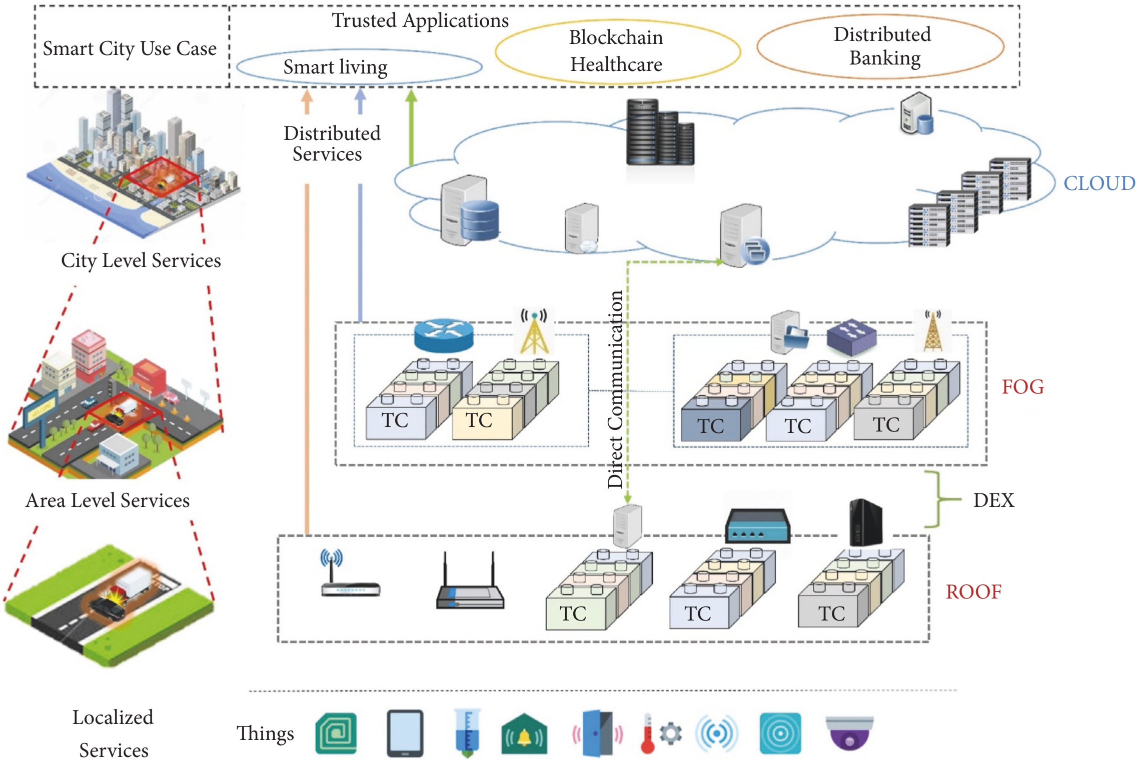


Figure 8: Composition of services via TrustChains in a smart city.

Table 2: Categorization of Node Types in TrustChain Networks.

Node Type	Storage	Validation
Full Node (e.g. Cloud nodes)	Full TrustChain	Yes
Heavy Node (e.g. Fog Nodes)	TrustChain below Fog	Possible
Light Node (e.g. ROOF Nodes)	Mostly store block headers only	Rarely
Data Issuers (e.g. Sensors, IoT nodes)	None	No

to act as a control channel to record the overall state of the IoT system including resources and service availability. Hence, the removal of redundant data storage will enable storing more information into a single block with the same block size, which can significantly improve the transaction processing speed. Moreover, cooperation between nodes can effectively improve the efficiency of the entire IoT system.

3.2.1. Trust-Based Consensus Management Protocol. As we highlighted in Section 2.1, many consensus management protocols have been developed in both permissioned and permissionless blockchain networks to minimize the computing resources required for mining, minimize mining delay, and improve robustness against a large number of distributed nodes. However, all these techniques were in the assumption of the trustless nature of blockchain, even though the trust factor places a major role in the blockchain ecosystem; ranging from choosing the right business partner for trustworthy service provisioning, to creating a trustworthy mining process. Therefore, we identify trust as an important property on validating the blocks in the proposed TrustChain and provide comprehensive details on how to achieve it in this section. In contrast to well-known consensus mining techniques like PoW, PoS, and PoA,, the algorithm proposed here is uniquely identified as Proof-of-Trust (PoT) throughout this paper.

In the TrustChain or even in other blockchain versions, miners' task is to verify individual data blocks and connect them with the genesis chain. However, to be a miner in conventional mining schemes, miners must have either computational power, wealth, authority, or similar type of advantage over others. Similarly, in the PoT method, a group of nodes who have maintained higher-level trustworthiness are chosen as the governing property to be selected as miners or "*Trust Bloggers*" (TB) in the TrustChain network. Furthermore, in the consensus process in PoW, anyone with a mining rig can participate in consensus and miners can join and leave the network without impacting consensus. On the other hand, Byzantine Fault Tolerance (BFT) mining-based methods use the centralized validator list chosen by a central authority. Even though BFT based methods show good performance against the use of efficient resources as the underlying mechanism is based on a voting system, it still lacks the decentralized nature of mining as the selection of

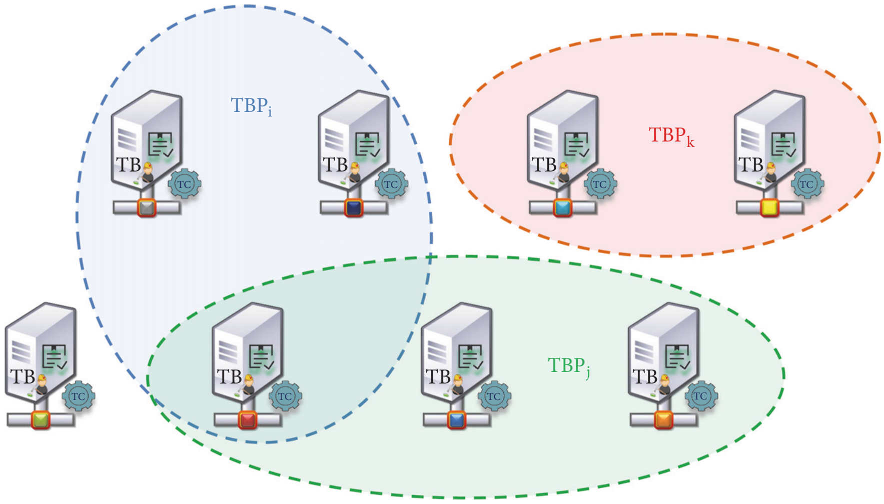


Figure 9: Consortium of Trust Bloggers (TBs) and Trust Blogger Pools (TBPs) in the TrustChain network.

validators is controlled by the centralized manner in permissioned blockchain networks [29]. In such cases, anyone can spin up a validator, but it can only participate in consensus if the authority adds the new node to the validator list. This requirement for a recommended validator list means that BFT follows a closed membership system.

Motivated by the BFT based methods, a voting system relying on the trust service is used in the TrustChain network for the consensus mining process. However, the selection of TBs in the TrustChain network is not controlled by a central authority and allows any node who has enough trustworthiness to be selected as a blogger. In TrustChain, each TB decides which other TB they trust using the trust management services given under the Trust Service Component as disuse in Section 3.1. The list of TBs is called the Trust Blogger Pool (TBP) in the context of TrustChain. Depending on the network size, service distribution, type of TrustChain, and availability of Trust Attributes to calculate trust, multiple segments of TBPs with various pool sizes, and overlapping sections create the global TBP which covers the whole IoT network, strengthening the distributed nature of IoT further as shown in Figure 9. Due to the open membership nature of TBP, anyone can spin up a validator and participate in consensus because there is no single master authority deciding which nodes get to participate in the consensus process. Inherently, it allows for growing decentralization unlike BFT as more and more nodes are added to the network and form new pools around them making it difficult for a malicious miner to interfere with the voting process. For example, unlike PoW or BFT based systems, no single party can own 51% of a global trust network in TrustChain technology, as the selection of TBs based on trustworthiness in contrast to factors like computing power, authority and wealth, which can be controlled to gain unfair advantages.

However, when pools are not overlapping with each other, there is a possibility that different pools may maintain a different copy of the TrustChains and it will put the overall consensus mechanism in jeopardy. In such cases, a smart contract between the disjoint TBPs can be created to continue the voting process while preserving their pool sizes as they are. On the other hand, a limited number of trustworthy authorities like government control bodies can be selected as the TBs to overcome any disjoints, assuming their inherited trustworthiness. Yet, the later solution might reduce the power of the decentralized architecture of the TrustChain to some extent. Therefore, it is important to consider a minimum level of trustworthiness when it comes to the selection of TBs to avoid disjoint TBPs.

In contrast to BFT based methods, we propose to use the REK model, discussed in Section 2.2, based on our previous work [14] to assist the TB selection process. We follow the same procedure mentioned in (11) to calculate the Experience TM and Reputation TM. However, TAs which represent the Knowledge TM must be redefined to grasp the true features of TBs including its social properties and dependable properties. TAs which define social properties like Colocation, Cowork, Cooperativeness, Frequency, Length of Interactions, Mutuality, Centrality, and Community of Interest assist to catch how well the TBs have behaved in the past to uphold the moral standards of the TrustChain network while dependable TAs like reliability, availability, safety, integrity, maintainability, and confidentially [30]. There are many models that can be seen in the literature to quantify dependable properties like in [30–32]. Therefore, this paper formulates numerical models to analyze social TAs as described in Section 3.1 based on our previous work in [14]. Once, both social and dependable TAs are calculated, a cumulative score for Knowledge TM can be obtained as in (2). Further, combining it with (11), the trust value of each candidate TB can be modeled as in (3).

$$K_{ij}(t) = \rho.K_{ij}^{Social} + \sigma.K_{ij}^{Dependable} \tag{2}$$

$$Trust_{TB} = \alpha.E_{TB} + \beta.R_{TB} + \gamma.K_{TB} \tag{3}$$

Where $Trust_{TB}$ represents the trustworthiness of TB and K_{TB}, E_{TB}, and R_{TB} denote the TMs based on knowledge, experience, and reputation. Based on the trust value of TBs, one who has the highest trustworthiness is chosen as the

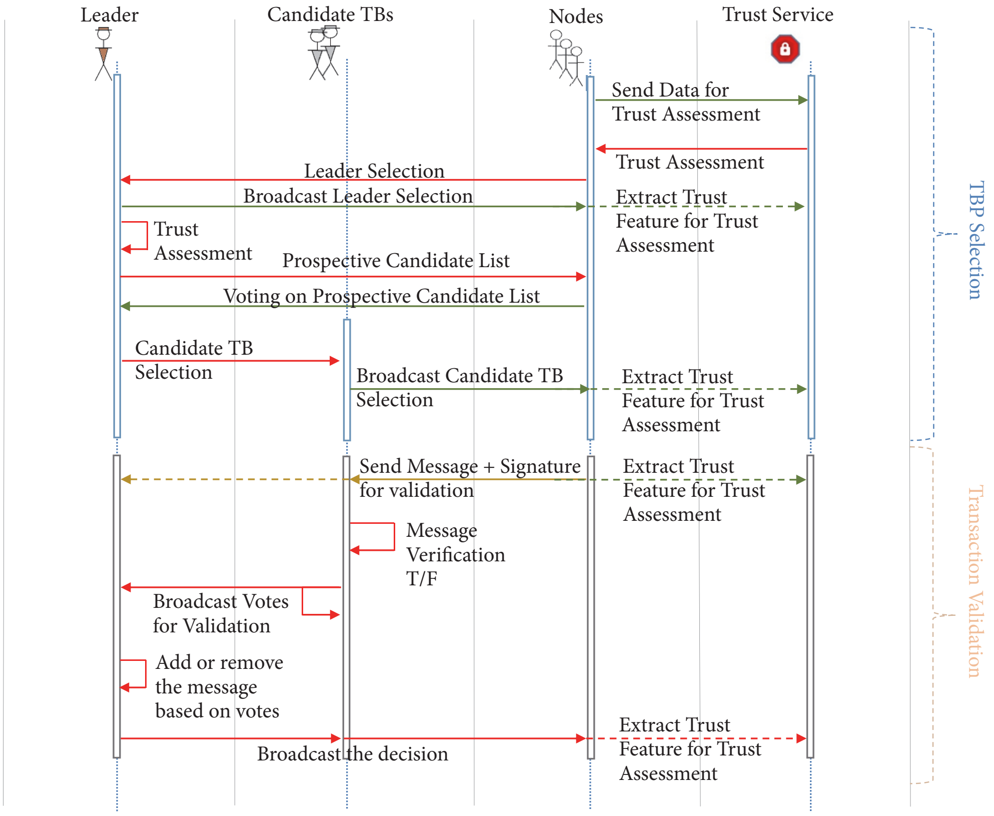

Leader
Candidate TBs
Nodes
Trust Service
Send Data for Trust Assessment
Trust Assessment
Leader Selection
Broadcast Leader Selection
Extract Trust Feature for Trust Assessment
Trust Assessment
Prospective Candidate List
Voting on Prospective Candidate List
Candidate TB Selection
Broadcast Candidate TB Selection
Extract Trust Feature for Trust Assessment
Send Message + Signature for validation
Extract Trust Feature for Trust Assessment
Message Verification T/F
Broadcast Votes for Validation
Add or remove the message based on votes
Extract Trust Feature for Trust Assessment
Broadcast the decision
TBP Selection
Transaction Validation

FIGURE 10: Trust-based consensus management protocol.

leader for a specific TBP and broadcast it to other nodes in the pool with the *leader's* digital signature. Upon receiving the signature from the leader, other nodes can verify it and then acknowledge it with their own signatures. The leader is mainly responsible for managing the consensus process until its term period has expired. Once the term of a leader has expired, a new leader must be chosen based on the highest trustworthiness value of a node.

After a leader is elected, he chooses a list of deputy *candidates* for the *blogging* or in other words the validating process. In order to select such TB *candidate list*, the *leader* TB evaluates his trust relationships with other prospective candidates for TBs and chooses the ones that have the highest trust relationships with him. To determine the satisfactory margin of trustworthiness, a threshold value or machine learning approach is used, as described in [14]. Then the leader TB sends the list to other nodes to cross-check their relationship with the selected TB list. If the other nodes are happy with the trust level, they can respond back to the leader with their approval or simply ignore it to show the disproval. Then based on the votes from nodes, the ones who have higher votes will be selected as the final *candidate list* for the blogging process and will be broadcast back again to inform the other nodes in the pool. The overall process of selecting a leader and candidate TB list is shown in Figure 10.

Once the *leader* and *candidate* TB list are decided, nodes can initiate transactions and broadcast messages to the TBP with their signatures which can be generated by taking the hash of both the message and their Secret Key (SK) as in (4). Furthermore, freedom is given to the message generator to encrypt messages using a suitable encryption mechanism or to anonymize the data depending on the privacy requirement in contrast to conventional blockchains. Upon receiving these blocks by TBs, they can first validate the message using verification function denoted in (5) and the senders Public Key (PK).

$$Sign\left(Message, SK\right) = Signature \tag{4}$$

$$Verify\left(Message, Signature, PK\right) = True/False \tag{5}$$

Then, based on the result of the verification function, each TB can vote for adding this specific message to a block in the genesis chain. If the leader receives votes to include the block in the genesis chain, it generates a header for the message and the resulting block is added to the existing chain. If he receives contradictory votes or votes do not have a majority then the message is ignored as it can tamper with malicious intentions. The process of adding each new block to a genesis chain is shown in Figure 10.

3.3. Smart Contract Service. Smart contracts are simply self-executing and immutable codes that reside in a blockchain. They essentially remove the need for a central broker due to their self-executing nature and can be triggered upon receiving a request from IoT nodes, other TrustChains, or by the smart contract itself based on the self-triggering rules inside the contract. In the TrustChain service, smart contracts are stored in a separate chain to improve the efficiency associated process like creating, storing, executing, and terminating. The registry component in the smart contract service basically register newly deployed contracts by issuing an address, name, and version and help to track the outcome of the contract once it is initiated by storing the hash of the result. A secure container includes a secure operating system, smart contract language, runtime environment, and a software development kit to create and run the smart contracts within the TrustChain service.

Even though smart contracts work in a trustless manner after signing the contract, it is beneficial to have a trust evaluation mechanism in place before the contract is signed to implement proactive measures to avoid contract violations. Hence, smart contract service discussion is coupled with the trust service to identify accountabilities and enforce them autonomously. For example, let us consider a distributed market place in a smart city use case in which stakeholder exchange goods for cryptocurrencies. In a normal condition, the seller must ship the items as soon as he received the transaction from the customer. On the other hand, in case of the seller deliberately delaying the shipment, the smart contract will be triggered, and money will be refunded to the customer's account. However, this process might involve some transaction fees and a waste of effort in the perspective of the customer. Hence, to avoid such outcome, the trust service is used to evaluate the trustworthiness of the prospective sellers based on the REK model, discussed in [14], and recommend appropriate stakeholders before establishing the smart contract.

Let us assume Alice (a) the IoT user needs to find a good banker, say Bob (b), and they need to establish a smart contract for their mutual interactions. For the generality, let us take the trust level between any IoT nodes "*a*" and "*b*", with respect to *a*'s preference being denoted by Trust_{ab}, similar to concepts discussed in Section 3.2 and our previous work [12–14]. Then the trust level between "*a*" and "*b*" is calculated as in

$$Trust_{ab} = \alpha . E_{ab} + \beta . R_{ab} \tag{6}$$

where E_{ab} denotes the experience between "*a*" and "*b*", R_{ab} calculates the reputation of "*b*", and α and β are normalizing coefficients to bring the final scores in between 0 and 1. Note that we have deliberately omitted calculating Knowledge TM as it needed personal information to evaluate the trust based on knowledge. To calculate the Experience TM and Reputation TM, we recall the concept of directed graph theory from Section IV of [27] as in Figure 11. Note that, Figure 11 only shows the relative interaction distance among each other and it does not represent the actual physical distance between them.

Let us take TA evaluation of the j^{th} transaction with node x as $\text{v}_x(\text{j})$, successfulness of the j^{th} transaction with node x as $\text{a}_x(\text{j})$, and time attenuation function between current time and transaction time w.r.t. node x as $\text{t}_x(\text{i}, \Delta\text{t})$. Then the experience between "*a*" and "*b*" can be calculated as in

$$E_{ab} = T_{ab}^{H} . E_b = T_{ab}^{H} . \frac{\sum_{j=1}^{m} v_b(j)\, a_b(j)\, t_b(j, \Delta t)}{\sum_{j=1}^{m} a_b(j)\, t_b(j, \Delta t)} \tag{7}$$

where H denotes the hop distance to the trustee "*b*"; T_{ab}^{H} represents the transition matrix between "*a*" and "*b*"; and unique value at (a,b) of the T_{ab}^{H} represents the connection between "*a*" and "*b*"; and $\text{t}_x(\text{i}, \Delta\text{t})$ with respect to arbitrary node x can be calculated as in (8). However, the evaluation of $\text{v}_x(j)$ is context and content dependent and TAs that governs the particular transaction must be intelligently selected in order to evaluate the $\text{v}_x(j)$. For example, TAs like response time, timely delivery, quality of the product/service, and fees associated can be taken into consideration in this example. Moreover, successfulness of the j^{th} transaction denoted by $\text{a}_x(j)$ can be defined based on the criticalness of the application. For example, it can be either 0 or 1 for strict applications while it can be varied in between for other cases.

$$t_x(j, \Delta t) = \begin{cases} 1 & \Delta t < \alpha \\ e^{-\Delta t} & \alpha < \Delta t < \beta \\ 0 & \beta < t \end{cases} \tag{8}$$

where α and β represent the threshold values that adjust the importance of the transaction in consideration relative to the current time. If the application demands more recent transaction in order to evaluate the trust, then β must be set very close to the origin. Following the same procedure as above and the model discussed in Section IV of [27], the reputation of "*b*" can be calculated as in

$$\begin{aligned} R_{ab} &= \sum_{n=1}^{N=6} T_{nb}^{n} . \text{d}(\text{n}) . R_b \\ &= \sum_{n=1}^{N=6} T_{nb}^{n} . \text{d}(\text{n}) . \frac{\sum_{j=1}^{m} v_n(j)\, a_n(j)\, t_n(j, \Delta t)}{\sum_{j=1}^{m} a_n(j)\, t_n(j, \Delta t)} \end{aligned} \tag{9}$$

where $\text{v}_n(j)$, $\text{a}_n(j)$, and $\text{t}_n(i, \Delta\text{t})$ take their useful meaning as discussed above but with respect to node "*n*" or else they represent the reputation value of "*b*" with respect to node "n". Note that the hop count from node "*b*" is limited to six here based on the "small world problem" discussed in [33], in which authors argue that any node is reachable within the range of six hopes. Further, based on the n^{th} value, the reputation is attenuated by a factor of d(n) as in (10) to reduce the effect of reputation that far away nodes hold on to node "*b*".

$$d(n) = \frac{7 - n}{6} \tag{10}$$

After that, the final trust score of the node "*b*" with respect to "*a*" can be obtained by substituting (7) and (9) into (6) as shown in

$$\begin{aligned} Trust_{ab} &= \alpha . E_{ab} + \beta . R_{ab} \\ &= \alpha . T_{ab}^{H} . \frac{\sum_{j=1}^{m} v_b(j)\, a_b(j)\, t_b(j, \Delta t)}{\sum_{j=1}^{m} a_b(j)\, t_b(j, \Delta t)} \\ &+ \beta . \sum_{n=1}^{N=6} T_{nb}^{n} . \text{d}(\text{n}) . \frac{\sum_{j=1}^{m} v_n(j)\, a_n(j)\, t_n(j, \Delta t)}{\sum_{j=1}^{m} a_n(j)\, t_n(j, \Delta t)} \end{aligned} \tag{11}$$

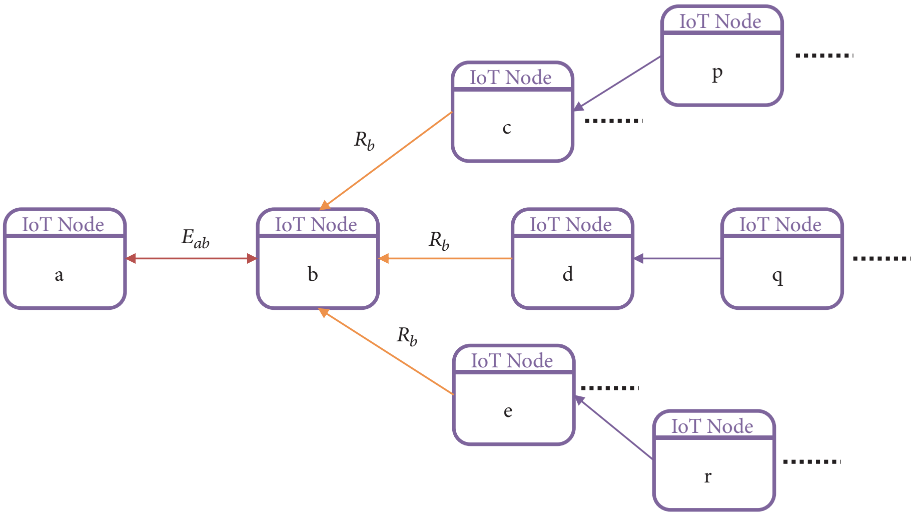


Figure 11: Experience and Reputation flow among IoT nodes.

Further, TrustChain is designed with the requirement of interoperability in mind to enable DEX. Due to the autonomous nature of executing prearranged rules and also its ability to become embedded inside TrustChain networks, smart contracts provide a promising approach to implement DEX in the TrustChain network. Having DEX capability not only assists in maintaining a different ledger in TrustChain for different services like Data, Transactions, Trust, Policy/Membership, and Smart Contracts itself but also allows communication with external blockchains whenever necessary, minimizing content exposure preserving privacy of the users as well as their data. Nevertheless, this type of smart contract based DEX enables the deployment of complex use case scenarios like a smart city with millions of IoT nodes possessing different resources and service subscriptions. For example, Figure 12 illustrates a scenario where several distributed ledgers interact together through the smart contract ledger. In such a framework, a node can deposit its data in "Data Ledger" transactions based on cryptocurrencies in "Transactions Ledger" and create, execute, or terminate smart contracts via the "Smart Contract Ledger." Consequently, associated managing functions related to the above example will be triggered through the smart contract ledger by the "Trust Service Ledger" and "Policy and Membership Ledger."

TrustChain must ensure the privacy of the stakeholders in both internal and external interactions whenever possible as per the GDPR. Hence, we propose to use the concept of Zero Knowledge Proof (ZKP) to hide the user information when interacting with service providers through smart contracts [34]. Let us assume an IoT node belonging to Alice (Prover) needs to retrieve confidential documents from Bob (Verifier). For that, Alice needs to prove her identity to Bob by providing her Name, Date of Birth, and Social Security Number (SSN). However, if she provided this information, Bob can use this data for other purposes like user profiling or share it with third parties for monetary gain. To avoid that ZKP can be used to prove the identity of Alice without sending the actual information her. In order to improve the robustness of the algorithm against hiding identity information, this paper proposes to combine the ZKP with Diffie–Hellman key exchange algorithm [35] to create a novel consensus protocol for smart contracts on hiding personal data as in the Algorithm 1.

Note that Bob and Alice share only g, p, and their public keys (k_1, k_2).There is no way that an eavesdropper can interfere with the identification process without knowing the secret keys of Alice and Bob. Moreover, what is inside the two hash functions C and C' must be similar in order to satisfy the condition C= C'. Hence, essentially Alices' commitment t must be equal to t' as shown in

$$\begin{aligned} t' &= g^{r}.k_1^{C} + k_2^{r}.s_k^{C} \\ &= g^{(v.x-C.x)}.\left(g^{x}\right)^{C} + \left(g^{y}\right)^{(v.x-C.x)}.\left(g^{xy}\right)^{C} \\ &= g^{(v.x)} + g^{(v.x.y)} = k_1^{v} + s_k^{v} = t \end{aligned} \tag{12}$$

3.4. Membership and Policy Service. By design, TrustChain platform is developed as a permissioned type of blockchain to control the privacy matters associated with a public blockchain and support business requirements demands by the service providers. Therefore, all nodes who need a TrustChain service are required to register with the membership services in order to obtain an identity to access the distributed services in TrustChain such as carrying out transactions, obtaining services offered by service providers, interacting with smart contracts, etc. However, as there are no centralized authorities in a TrustChain network to manage such credentials, the responsibility relies on the TB with the support from the trust service discussed in Section 3.1. The selection of TBs is discussed in Section 3.2 under the consensus management protocol. Fundamentally, when assigning an identity by TB to a prospective node, it will first evaluate the trustworthiness of the node with respect to the services he is demanding. The trust evaluation process follows a similar approach as discussed in previous sections and if the trust level is at the expectation level, the node is granted to enter the TrustChain network with restricted

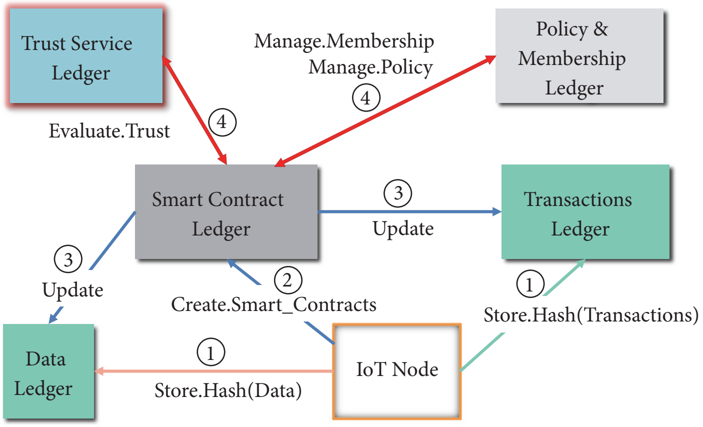


Figure 12: TrustChain distributed and interoperable ledgers.

```
1: Trust Service selects two large random prime numbers p (Prime) and g (Generator) s.t.
   256<p,g and communicate to both Alice and Bob.
2: Alice generates hash of her information and store in x=SHA256(Name, DoB, SSN).
3: Alice calculates $k_1=g^x$ mod p and sends it to Bob.
4: Bob takes the hash of his request and store in y, s.t. 0<y<p.
5: Bob calculates $k_2=g^y$ mod p and sends it to Alice.
6: Alice calculates the shared secret key, $s_k= (k_2)^x mod\ p$.
7: Bob calculates the shared secret key, $s_k= (k_1)^y mod\ p$.
8: Alice generates another random oracle v, s.t. 0<v<p.
9: Alice calculates her commitment, $t=(k_1)^v +(s_k)^v$.
10: Alice calculates the challenge, $C$=SHA256(g, $s_k$, $t$).
11: Alice sends the response, $r=(v.x - C.x)\ mode\ p$ and challenge $C$ to Bob.
12: Bob calculate the commitment, $t'= g^r. (k_1)^C +(k_2)^r. (s_k)^C$ and challenge $C'$=SHA256(g, $s_k$, $t'$).
13: If $C=C'$, Bob can satisfy that Alice has provided the requested information.
```

Algorithm 1: ZKP with DH to Suppress Personal Information Exposure.

permission based on his trust level. It is up to the node to behave and collaborate in a good manner to improve his reputation over the time if he needs to access more advanced services including becoming a TB.

Policy services mainly manage areas such as preserving the privacy of the prosumers, monitoring consents, meeting consensus rules and ensuring accountability in case of policy violations. In this regard, the trust service can support to track such rules to detect violations beforehand and take necessary countermeasures in case of an incident already occurring. A more detailed version of implementing such a system is discussed in our previous work [13], in compliance with GDPR legislation when it comes to privacy matters.

4. Discussion

4.1. TrustChain vs. Privacy Compliance. The initial versions of blockchains like Bitcoin [16] and Ethereum [26] were designed and developed to facilitate trustless data management, transparency, and promote the decentralization aspects. However, they lack the fundamental requirement of privacy when it comes to managing personal and proprietary information in an IoT ecosystem. Several blockchain technologies were invested later to rectify these privacy issues based on the permissioned and private blockchain concepts like Hyperledger Fabric [23]. However, due to interference from centralized authorities in managing membership and consensus protocols, they also had to compromise the initial properties of blockchains like decentralization capabilities and scalability in a setting like the IoT.

In contrast, TrustChain is designed in such a way that it only stores the information allowed by the users. Techniques like ZKP, encryption, and anonymization are used to hide the sensitive data while communicating amongst relevant stakeholders and evaluate trust accordingly without affecting the privacy. The ZKP algorithm suggested in the text is one of the promising techniques used in the TrustChain to communicate amongst parties without revealing their personal information. This property satisfies the GDPR requirements under the *right to object processing and right to control profiling* using personal information. Further, it allows encryption and other anonymization techniques within TrustChain to hide delicate information as it is not required to maintain a public ledger system like in a Bitcoin blockchain. Hence the user is given the *right to restrict processing* as per the GDPR legislation. On other hand, an application of parallel distributed architectures enables the storing of sensitive data in a local database and linking the data to TrustChain through a cryptographic link, which in fact allows the user to remove data from the TrustChain whenever necessary, thus satisfying GDPR conditions under the *right to erasure* in addition to the *right to data portability.* Moreover, the underlying technology of the TrustChain service is built upon a trust management system and hence every interaction is monitored, and trust is evaluated accordingly to support future interactions. This fact essentially enables the accountability principle and allows the user to inform how its data is being used as demanded by the *right to be informed* condition in GDPR.

4.2. TrustChain vs. Edge Computing. Typically, an IoT ecosystem represents nodes ranging from small sensors (which only emit) data to massive complex data centers (which process billions of interactions per time). In the edge computing setup, these small sensors can be found at the bottom of the hierarchy. Nodes that have a comparatively higher processing power and storage lie in the middle of the network, and large data centers represent nodes at the cloud layer. It is challenging to implement conventional blockchain technologies towards the end of hierarchy due to associated resource restrictions to perform consensus algorithms and limited storage to store the massive public ledger. In contrast, the consensus protocol in TrustChain is simply a combination of trust and BFT and so computation power that can broadcast a set of messages is more than enough to implement the consensus algorithm. Therefore, TrustChain can be easily deployed towards the end of edge computing hierarchy. Furthermore, a lightweight protocol enables efficient processing of blocks while saving more energy compared to traditional blockchain technologies in which a rig of miners must coordinate with each other to solve a highly complex cryptographic puzzle. This, in fact, improves the performance of the overall system and prevents divergence of the ledger as nodes do not need to wait for much longer to identify the correct copy of the ledger in contrast to a traditional blockchain in which it can take at least ten minutes to add a new block to the genesis chain.

Moreover, TrustChain uses a permissioned identity management protocol when selecting suitable validators like in Hyperledger. Application of TBP in TrustChain enables us to expand the TrustChain network just like in Bitcoin blockchain networks, in contrast to Hyperledger in which centralized authority is used to select the validators. Further, the trust service in the TrustChain works collaboratively with membership and policy services to identify trustworthy TB and discourage malicious nodes entering the TrustChain network, contributing to the mining process. This makes the consensus protocol described in Section 3.2 nearly immutable as only trustworthy nodes are given permission to perform the mining process.

According to TrustChain architecture Trust Broker Pool (TBP) is a small portion of the global pool which consists of millions of overlapping TBPs. One TBP contains 51% of trust nodes means; more than half of the global IoT nodes trust each other. This scenario is practically impossible to occur as it is impossible to create such a large trust network by an individual or even by multiple organizations as trust depends on many factors as discussed in the paper. Further, let us assume 51% computing power is control by one organization. In such a situation, this organization can create a TBP consisting of all of their computers. However, it is highly unlikely that all other nodes external to this specific TBP would trust all nodes inside one TBP in any circumstance. Therefore, this specific TBP cannot interfere with the decision-making process as TrustChain consensus protocol is not based on the computer power but only on trustworthiness. Thus, having 51% computing power is also useless in TrustChain network to interfere with the consensus process.

In addition, the trust service is useful to check the integrity of the data generated by IoT nodes as described in Section 3.1. This feature not only improves the overall experience of the prosumers but also guarantees accountability in case of a policy or privacy violation. In comparison to traditional versions of blockchain, it is only available with the TrustChain platform as we discussed previously. Further, most of the numerical results based on the trust are carefully analyzed in our previous work [12–14, 36] and thus not repeated here. However, Table 3 summarizes the effectiveness of the proposed TrustChain architecture rationally over existing blockchain technologies as discussed in this work.

5. Conclusions

TrustChain is a permissioned blockchain network designed to enhance the privacy of its prosumers while improving the efficacy of TrustChain service compared to traditional blockchain technologies specifically in a distributed environment like the IoT. The major difference of TrustChain compared to traditional alternatives is the application of computational trust on realizing various functions inside the TrustChain service. Among them, evaluation of trust allowed (i) developing a novel lightweight consensus management protocol by combining with the BFT protocol; (ii) measuring the trustworthiness of participating prosumers before creating smart contracts and before initiating interactions among them; (iii) the membership and policy services to take intelligent decisions when adding new nodes to the network, selecting TBs, and realizing the accountability in case of consensus, privacy, or consent violation; and (iv) checking the integrity of each interaction before adding them to the genesis chain.

Moreover, TrustChain empowers the edge computing architecture of IoT due to its survivability with low computing and storage resources which is not the case with traditional approaches. Nevertheless, this prevents the need for centralized processing such as at the cloud allowing implementing innovative privacy preserving solutions at the Fog and the ROOF via TrustChain membership and policy services. However, it is important to allow vertical services in parallel with horizontal services to facilitate many intelligent solutions. To enable such services, a decentralized exchange is introduced based on the smart contract concept to negotiate among vertical layers to combine data and services in a

Table 3: TrustChain vs. traditional Blockchain technologies.

Property			Bitcoin	Ethereum	Hyperledger	TrustChain
Permission Restrictions			Permissionless	Permissionless	Permissioned	Permissioned
Consensus			PoW	PoW	PBFT	Trust+BFT
Energy Saving			No	Partial	Yes	Yes
Decentralized Regulation			Yes	Yes	Partial	Yes
Smart Contracts			No	Yes	Yes	Yes
Scalability		Node	High	High	Low	High
		Performance	Low	Low	High	High
Immunity			25% of the Computing power	51% of Stake	33.3% of faulty replicas	Nearly Immutable
Native Currency			Yes	Yes	No	Possible
Incentive			Mining Fee	Mining Fee	No	Trust
Blockchain as a control chain			No	No	No	Yes
Privacy			No	No	Partial	Yes (e.g. with ZKP)
GDPR Privacy Compliance	Right to	be informed	No support	No support	No support	Support
		access	Support	Support	Support	Support
		rectification	No support	No support	No support	Support
		erasure	No support	No support	No support	Support
		restrict processing	No support	No support	Support	Support
		data portability	No support	No support	Support	Support
		object processing	No support	No support	Partial	Support
		control profiling	No support	No support	Partial	Support

vertical manner via TrustChain and beyond that. Nevertheless, TrustChain acts in a control layer in such cases to minimize the ledger size and enable an efficient orchestration among different services. Further, TrustChain is embedded with unique techniques to improve privacy when dealing with delicate personal information and to comply with GDPR legislation, for example, using concepts like ZKP, encryption, and anonymization.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) [2018-0-00261, GDPR Compliant Personally Identifiable Information Management Technology for IoT Environment] and this study was financially supported by Hanbat National University financial accounting research fund, 2018 year.

References

[1] A. Opher, A. Onda, A. Chou, and K. Sounderrajan, *The Rise of the Data Economy: Driving Value through Internet of Things Data Monetization*, IBM Corporation, Somers, NY, USA, 2016.

[2] M. Seliem, K. Elgazzar, and K. Khalil, "Towards privacy preserving iot environments: a survey," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 1032761, 15 pages, 2018.

[3] European Union, "General data protection regulation (GDPR)," *Official Journal of the European Union*, vol. L119, pp. 1–88, 2016.

[4] H. Tianfield, "Towards edge-cloud computing," in *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*, pp. 4883–4885, Seattle, WA, USA, December 2018.

[5] W. Yu, F. Liang, X. He et al., "A survey on the edge computing for the internet of things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[6] V. Moysiadis, P. Sarigiannidis, and I. Moscholios, "Towards distributed data management in fog computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7597686, 14 pages, 2018.

[7] M. T. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy," in *Proceedings of the in Proceedings*

of the Sixth International Conference on Advances in Future Internet, pp. 48–55, 2014.

[8] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the in Proceedings of the 2015 workshop on mobile big data*, pp. 37–42, 2015.

[9] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[10] T. T. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: a data processing view of blockchain systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1366–1385, 2018.

[11] N. Rifi, N. Agoulmine, N. Chendeb Taher, and E. Rachkidi, "Blockchain technology: is it a good candidate for securing iot sensitive medical data?" *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–11, 2018.

[12] U. Jayasinghe, A. Otebolaku, T.-W. Um, and G. M. Lee, "Data centric trust evaluation and prediction framework for IOT," in *Proceedings of the 9th ITU Kaleidoscope Academic Conference: Challenges for a Data-Driven Society, ITU K 2017*, pp. 1–7, Nanjing, China, November 2017.

[13] U. Jayasinghe, G. M. Lee, and A. MacDermott, "Trust-based data controller for personal information management," in *Proceedings of the 2018 International Conference on Innovations in Information Technology (IIT)*, pp. 123–128, Al Ain, UAE, November 2018.

[14] U. Jayasinghe, G. M. Lee, T. W. Um, and Q. Shi, "Machine learning based trust computational model for IoT services," *IEEE Transactions on Sustainable Computing*, pp. 1-1, 2018.

[15] A. Meloni, S. Madanapalli, S. K. Divakaran et al., "Exploiting the IoT potential of blockchain in the IEEE P1931.1 ROOF standard," *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 38–44, 2018.

[16] S. Nakamoto, *Bitcoin: A Peer-To-Peer Electronic Cash System*, 2008.

[17] J. Sousa, A. Bessani, and M. Vukolic, "A byzantine Fault-Tolerant ordering service for the hyperledger fabric blockchain platform," in *Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2018*, pp. 51–58, June 2018.

[18] J. Kang, Z. Xiong, D. Niyato, P. Wang, D. Ye, and D. I. Kim, "Incentivizing consensus propagation in proof-of-stake based consortium blockchain networks," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 157–160, 2019.

[19] X. Fan and Q. Chai, "Roll-DPoS: a randomized delegated proof of stake scheme for scalable blockchain-based internet of things systems," in *Proceedings of the the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pp. 482–484, New York, NY, USA, November 2018.

[20] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018*, pp. 1545–1550, May 2018.

[21] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: architecture, consensus, and future trends," in *Proceedings of the 6th IEEE International Congress on Big Data, BigData Congress 2017*, pp. 557–564, June 2017.

[22] S. King and S. Nadal, *Ppcoin: Peer-To-Peer Crypto-Currency with Proof-Of-Stake*, vol. 19, self-published paper, 2012.

[23] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Proceedings of the Workshop on distributed cryptocurrencies and consensus ledgers*, 2016.

[24] F. Schuh and D. Larimer, "Bitshares 2.0: general overview," 2019, https://cryptorating.eu/whitepapers/BitShares/bitshares-general.pdf.

[25] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.

[26] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, 2014.

[27] U. Jayasinghe, N. B. Truong, G. M. Lee, and T.-W. Um, "RpR: a trust computation model for social internet of things," in *Proceedings of the Ubiquitous Intelligence &amp; Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Intl IEEE Conferences*, pp. 930–937, 2016.

[28] S. Liu, J. Wu, and C. Long, "IoT meets blockchain: parallel distributedarchitecture for data storage and sharing," in *Proceedings of the IEEE International Conference on Blockchain*, p. 1, Halifax, Canada, 2018.

[29] J. Kwon, "Tendermint: Consensus without mining," *Retrieved May*, p. 2017, 2014.

[30] M. Al-Kuwaiti, N. Kyriakopoulos, and S. Hussein, "A comparative analysis of network dependability, fault-tolerance, reliability, security, and survivability," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, pp. 106–124, 2009.

[31] A. Avižienis, J. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.

[32] G. Hatzivasilis, I. Papaefstathiou, and C. Manifavas, "Software security, privacy, and dependability: Metrics and measurement," *IEEE Software*, vol. 33, no. 4, pp. 46–54, 2016.

[33] J. Travers and S. Milgram, "The small world problem," *Phychology Today*, vol. 1, no. 1, pp. 61–67, 1967.

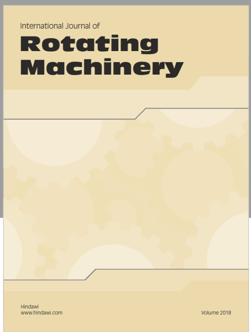
[34] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, STOC 1988*, pp. 103–112, May 1988.

[35] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

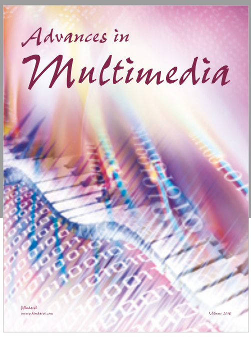
[36] U. Jayasinghe, H. W. Lee, and G. M. Lee, "A Computational Model to Evaluate Honesty in Social Internet of Things," in *Proceedings of the 32nd ACM SIGAPP Symposium On Applied Computing*, pp. 1830–1835, Marrakesh, Morocco, 2017.

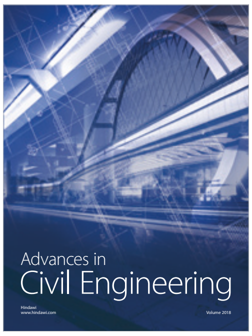
Journal of
Engineering

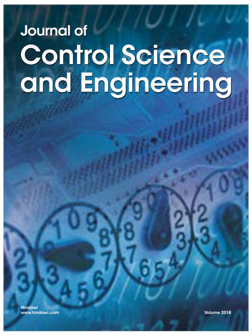
The Scientific World Journal


International Journal of
Rotating Machinery

Journal of
Sensors

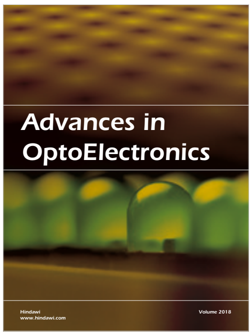

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

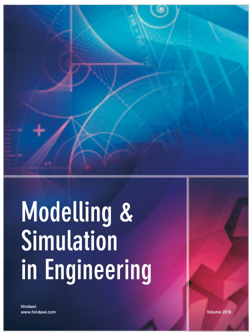
Journal of
Robotics

Journal of
Electrical and Computer Engineering

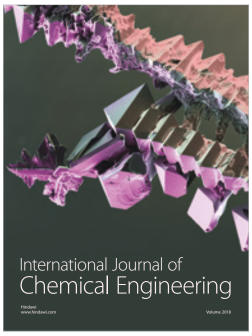

Advances in
OptoElectronics

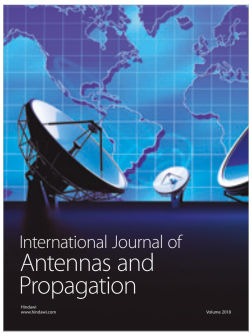
VLSI Design

International Journal of
Navigation and Observation

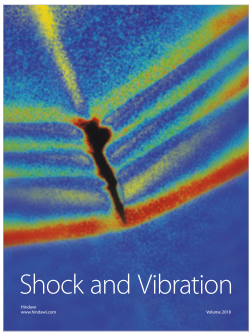

Modelling & Simulation in Engineering

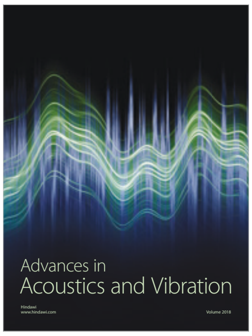
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration