Blockchain-based Trust and Reputation Management for Securing IoT

Blockchain-based Trust and Reputation Management for Securing IoT

Guntur Dharma Putra

A thesis in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Computer Science and Engineering
Faculty of Engineering

September 2022

Originality Statement

☑ I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Copyright Statement

☑ I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

Authenticity Statement

☑ I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.

Inclusion of Publications Statement

☑ The candidate has declared that some of the work described in their thesis has been published and has been documented in the relevant Chapters with acknowledgement.

A short statement on where this work appears in the thesis and how this work is acknowledged within chapter/s:

- The Chapter 2 of my thesis (Background and literature review) is partially comprised of a book chapter that I contributed to and is to appear in an edited book titled "Handbook of Blockchain" (Springer), editors: Duc Tran, My Thai and Bhaskar Krishnamachari.

- The contents of my technical chapter, i.e., Chapter 3, 4 and 5, are published in the IEEE TNSM Journal, IEEE Blockchain Conference and IEEE Network Magazine (partial), respectively.

- Acknowledgements of the work and the co-authors have been made accordingly in the Publications section.

Candidate's Declaration

☑	I declare that I have complied with the Thesis Examination Procedure.

To my family: past, present and future.

Abstract

The Internet of Things (IoT) brings connectivity to a large number of heterogeneous devices, many of which may not be trustworthy. Classical authorisation schemes can protect the network from adversaries. However, these schemes could not ascertain in situ reliability and trustworthiness of authorised nodes, as these schemes do not monitor nodes' behaviour over the operational period. IoT nodes can be compromised post-authentication, which could impede the resiliency of the network. Trust and Reputation Managements (TRM) have the potential to overcome these issues. However, conventional centralised TRM have poor transparency and suffer from single point of failures. In recent years, blockchains show promise in addressing these issues, due to the salient features, such as decentralisation, auditability and transparency. This thesis presents decentralised TRM frameworks to address specific trust issues and challenges in three core IoT functionalities.

First, a TRM framework for IoT access control is proposed to address issues in conventional authorisation schemes, in which static predefined access policies are continuously enforced. The enforcements of static access policies assume that the access requestors always exhibit benign behaviour. However, in practice some requestors may actually be malicious and attempt to deceive the access policies, which raises an urgency in building an adaptive access control. In this framework, the nodes' behaviour are progressively evaluated based on their adherence to the access control policies, and quantified into trust and reputation scores, which are then incorporated in the access control to achieve dynamic access control policies. The framework is implemented on a public Ethereum test-network interconnected with a private lab-scale network of Raspberry Pi computers. The experimental results show that the framework achieves consistent processing latencies and is feasible for implementing effective access control in decentralised IoT networks.

Second, a TRM framework for blockchain-based Collaborative Intrusion Detection Systems (CIDS) is presented with an emphasis on the importance of building end-to-end trust between CIDS nodes. In a CIDS, each node contributes detection rules aiming to build collective knowledge of new attacks. Here, the TRM framework assigns trust scores to each contribution from various nodes, using which the trustworthiness of each node is determined. These scores help protect the CIDS network from invalid detection rules, which may degrade the accuracy of attack detection. A proof-of-concept implementation of the framework is developed on a private lab-scale Ethereum network. The experimental results show that the solution is feasible and performs within the expected benchmarks of the Ethereum platform.

Third, a TRM framework for decentralised resource sharing in 6G-enabled IoT networks is proposed, aiming to remove the inherent risks of sharing scarce resources, especially when most nodes in the network are unknown or untrusted. The proposed TRM framework helps manage the matching of resource supply and demand; and evaluates the trustworthiness of each node after the completion of the resource sharing task. The experimental results on a lab-scale proof-of-concept implementation demonstrate the feasibility of the framework as it only incurs insignificant overheads with regards to gas consumption and overall latency.

Acknowledgements

In the name of God, the Most Gracious, the Most Merciful. All praise is due to God, Lord of all the worlds.

I would like to express my sincere gratitude to my supervisors, Prof Salil Kanhere, Prof Raja Jurdak and Dr Volkan Dedeoglu, for their endless support throughout my PhD journey. They always gave me the freedom to work which helped me explore my creativity and extend my network across multiple organisations and countries. I am grateful for the time and effort they have devoted and their invaluable guidelines and advice.

This journey would not have been possible without the support of my family. Special thanks to my Mom for her countless prayers and support; my wife, Icha, for the sacrifice and the unconditional love and compassion; and for my children, Naura and baby Raihan, for their patience and encouragement.

I would also like to thank my colleagues at UNSW, QUT and CSIRO, including Aleks, Ali, Regio, Nadeem, Wanli, Chuka, Sidra, Samuel, Ambrose, Pooja and Ahmad. I am also thankful to my external collaborators Salek, Edwards, James, Abhinav, Changhoon, Shantanu, Kealan, Zahra and Marcus for making my PhD journey even more enjoyable.

Contents

List of Figures

List of Tables

Abbreviations

AA Attribute Authority.

ABAC Attribute Based Access Control.

ACL Access Control List.

CapBAC Capability Based Access Control.

CIDS Collaborative Intrusion Detection System.

D2D Device-to-device.

DDS Dedicated Data Storage.

DL Deep Learning.

DRS Dynamic Resource Sharing.

DTRM Decentralised Trust and Reputation Management.

HIDS Host-based Intrusion Detection System.

IDS Intrusion Detection System.

IoT Internet of Things.

IoV Internet of Vehicles.

IPFS InterPlanetary File System.

MEC Mobile Edge Computing.

NIDS Network-based Intrusion Detection System.

PII Personally Identifiable Information.

PK Public Key.

PKI Public Key Infrastructure.

RSU Road Side Unit.

SC Service Consumer.

SDN Software Defined Network.

SK Secret Key.

SP Service Provider.

TEE Trusted Execution Environment.

TRM Trust and Reputation Management.

TTP Trusted Third Party.

VANET Vehicular Ad-Hoc Network.

XACML eXtensible Access Control Markup Language.

Publications

The main contributions of the thesis are based on the following publications:

Peer-reviewed Journal and Magazine

[1] **G. D. Putra**, V. Dedeoglu, S. S. Kanhere, R. Jurdak and A. Ignjatovic, "Trust-Based Blockchain Authorization for IoT," in *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1646-1658, June 2021. DOI: `https://doi.org/10.1109/TNSM.2021.3077276`. (Chapter 3)

[2] **G. D. Putra**, V. Dedeoglu, S. S. Kanhere and R. Jurdak, "Toward Blockchain-based Trust and Reputation Management for Trustworthy 6G Networks," to in *IEEE Network*, vol. 36, no. 4, pp. 112-119, July/August 2022. DOI: `https://doi.org/10.1109/MNET.011.2100746` (Chapter 5)

Book Chapter

[3] **G. D. Putra**, V. Dedeoglu, S. S. Kanhere and R. Jurdak, "Blockchain for Trust and Reputation Management in Cyber-physical Systems", in *Handbook on Blockchain.* Editors: Duc A. Tran, My T. Thai, and Bhaskar Krishnamachari. Springer Springer Optimization and Its Applications, vol 194. Springer, Cham. ISBN 978-3-031-07534-6. DOI: `https://doi.org/10.1007/978-3-031-07535-3_10`. (Chapter 2)

Refereed Conference Proceedings

[4] **G. D. Putra**, V. Dedeoglu, S. S. Kanhere and R. Jurdak, "Trust Management in Decentralized IoT Access Control System," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2020, pp. 1-9. DOI: `https://doi.org/10.1109/ICBC48266.2020.9169481`. (Chapter 3)

[5] **G. D. Putra**, V. Dedeoglu, S. S. Kanhere and R. Jurdak, "Poster Abstract: Towards Scalable and Trustworthy Decentralized Collaborative Intrusion Detection System for IoT," in *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2020, pp. 256-257. DOI: `https://doi.org/10.1109/IoTDI49375.2020.00035`. (Chapter 4)

[6] **G. D. Putra**, V. Dedeoglu, A. Pathak, S. S. Kanhere and R. Jurdak, "Decentralised Trustworthy Collaborative Intrusion Detection System for IoT," in *2021 IEEE International Conference on Blockchain (Blockchain)*, 2021, pp. 306-313. DOI: `https://doi.org/10.1109/Blockchain53845.2021.00048`. (Chapter 4)

Other

The following publications are not part of this thesis, but may be closely related:

[7] V. Dedeoglu, R. Jurdak, **G. D. Putra**, A. Dorri and S. S. Kanhere, "A Trust Architecture for Blockchain in IoT," in *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous)*, 2019, pp. 190-199. DOI: `https://doi.org/10.1145/3360774.3360822`.

[8] M. S. Ali, M. Vecchio, **G. D. Putra**, S. S. Kanhere and F. A. Antonelli, "Decentralized Peer-to-Peer Remote Health Monitoring System," in *Sensors*, 2020; 20(6):1656. DOI: `https://doi.org/10.3390/s20061656`

[9] J. Meijers, **G. D. Putra**, G. Kotsialou, S. S. Kanhere and A. Veneris, "Cost-Effective Blockchain-based IoT Data Marketplaces with a Credit Invariant," in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2021, pp. 1-9. DOI: `https://doi.org/10.1109/ICBC51069.2021.9461127`.

[10] N. Ahmed, R. A. Michelin, W. Xue, **G. D. Putra**, W. Song, S. Ruj, S. S. Kanhere and S. Jha, "Towards Privacy-preserving Digital Contact Tracing," in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2021, pp. 1-3. DOI: `https://doi.org/10.1109/ICBC51069.2021.9461052`.

[11] **G. D. Putra**, C. Kang, S. S. Kanhere and J. Won-Ki Hong, "DeTRM: Decentralised Trust and Reputation Management for Blockchain-based Supply Chains," in *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2022, pp. 1-5. DOI: `https://doi.org/10.1109/ICBC54727.2022.9805565`.

[12] N. Ahmed, R. A. Michelin, W. Xue, **G. D. Putra**, S. Ruj, S. S. Kanhere and S. Jha, "DIMY: Enabling privacy-preserving contact tracing," in *Journal of Network and Computer Applications*, Volume 202, 103356 (2022). DOI: `https://doi.org/10.1016/j.jnca.2022.103356`.

[13] K. Dunnett, S. Pal, Z. Jadidi, **G. D. Putra** and R. Jurdak, "A Democratically Anonymous and Trusted Architecture for CTI Sharing using Blockchain," in *2022 International Conference on Computer Communications and Networks (ICCCN)*, 2022, pp. 1-7. DOI: `https://doi.org/10.1109/ICCCN54977.2022.9868919`.

[14] K. Dunnett, S. Pal, **G. D. Putra**, Z. Jadidi and R. Jurdak, "A Trusted, Verifiable and Differential Cyber Threat Intelligence Sharing Framework using Blockchain," to appear in *2022 IEEE 21st International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2022. Preprint: `https://arxiv.org/abs/2208.12031`.

Chapter 1

Introduction

The Internet of Things (IoT) is a network where a collection of physical objects with computing capabilities connect and exchange data with each other to perform sensing and actuation tasks. IoT has been pervasively deployed in many application domains, enabling the so-called smart ecosystems, offering advantageous applications to the public, such as smart home [1], smart city [2] and smart healthcare [3]. In general, IoT ecosystems bring together heterogeneous IoT devices, infrastructures and components to achieve a common goal in providing services to users. For instance, a smart city ecosystem is enabled by a multitude of IoT sensors with different communication protocols, offering data analytics service to the end-users, such as real-time road traffic monitoring [4]. With the commercially available 5G networks and the increasing growth of global IoT adoption (See Figure 1.1), IoT demands ultra-dense connections with more distributed Device-to-device (D2D) communications [5]. Consequently, there will be a greater need for the IoT nodes to directly interact with other nodes, moving away from the traditional client-server paradigm [6].

However, ultra-dense connections would involve large number of interconnected devices with an absence of pre-established trust between participants. The incorporation of reliable authentication and authorisation schemes can protect the IoT network from unauthorised adversaries, giving some degree of trust, as only authorised nodes can access the network. However, these schemes could not ascertain in situ reliability and trustworthiness of authorised nodes, as these schemes do not monitor nodes' behaviour over the operational period. The case of Mirai Botnet exemplifies how IoT nodes can be compromised post-authentication and become malicious [9], which could severely impede the security and resiliency of the network. As critical

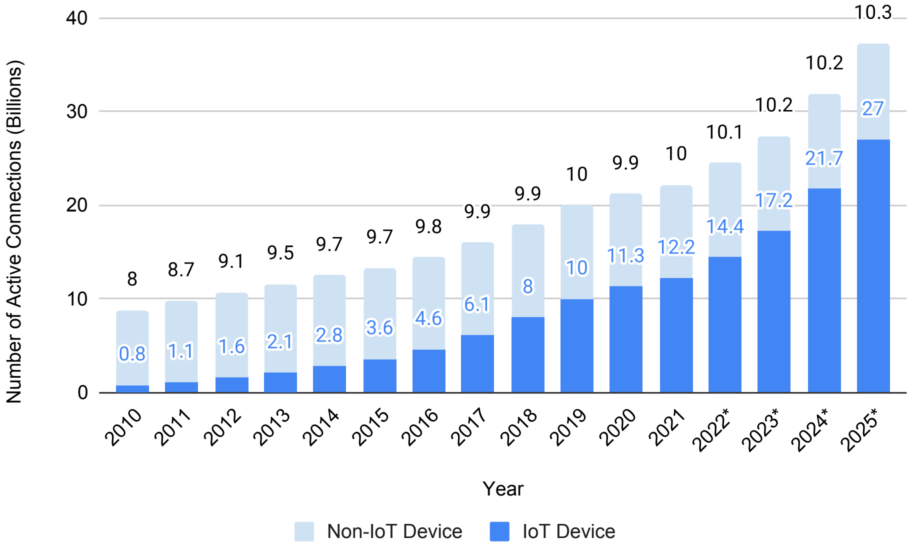

Figure 1.1: The number of global active IoT and non-IoT connections. The asterisk denotes forecasted numbers (Source: IoT Analytics Research [7], [8]).

infrastructures are being increasingly connected to IoT networks [10], the presence of malicious adversaries could eventually cause severe detrimental effects, as shown in the recent case of Colonial Pipeline attack, where a ransomware attack halted a major gas pipeline in the US [11]. In addition, the absence of trust between participants may discourage cooperation in the network, and thus highlights the urgent need to address the issues of IoT security and trustworthiness [12].

Trust and Reputation Management (TRM) is an effective approach to overcome the aforementioned trust issues, wherein an authority continuously evaluates the trustworthiness of each participant. TRM performs trust evaluation by processing feedback and ratings from other network participants, after which TRM quantifies the nodes' trustworthiness [13]. In TRM, each node is assigned trust or reputation scores that represent its trustworthiness level, using which other nodes in the network may conveniently infer the level of trustworthiness of each node. In addition, TRM could offer rewards and enforce punishments to the nodes based on these numerical measures [14]. The Vehicular Ad-hoc Network (VANET) is an example domain where TRM has been proposed. In VANET, dedicated Road Side Units (RSU) can collect information from surrounding vehicles to validate the exchanged message and assign trust scores to each vehicle and corresponding messages, using which the

TRM can scan for malicious or faulty vehicles [15]. Subsequently, each vehicle can query the RSU to obtain the latest trust scores of any vehicle in the proximity. A detailed background on the underlying processes in TRM is presented in Section 2.2.

However, relying on a trusted third party (TTP), e.g., RSU, to manage a TRM actually poses significant risks [16]. For instance, when the TTP is faulty or compromised, there is no precise guarantee that the underlying trust calculation would remain intact. Other nodes in the network cannot validate the trust calculation, as there is no transparency in evidence collection and trust quantification process. In addition, there is no assurance of the integrity and safety of the data, in cases where the TTP is compromised. These predicaments make the TRM susceptible to fraudulent modifications and data loss, for instance, when adversaries alter the interaction evidence to their advantage. Malicious nodes may attempt to gain advantage by spoiling the reputation of honest nodes or colluding to improve their scores, as this scheme has poor transparency. In addition, conventional TRM suffers from privacy issue as its authentication schemes may expose the real identities of the end-users, which should ideally be concealed and protected.

In recent years, blockchain, a peer-to-peer decentralised network where each node maintains and validates the data stored in interlinked blocks, has shown its potential applications beyond the financial domain [17]. With its salient features, such as transparency, tamper-resilient, verifiability, and pseudonymity, blockchain can be incorporated into TRM to improve its resiliency and address the aforementioned issues of conventional TRM [16]. For instance, the decentralisation inherent in blockchain eliminates the reliance on a TTP. Trust related data can also be stored on the shared immutable ledger, maintaining integrity and high availability. Smart contracts can be employed to enforce trustworthy collection of collaboration evidence and trust calculation. More discussions about blockchain technology and how its salient features would provide various benefits to TRM are presented in Section 2.4.

Designing a blockchain-based TRM to provide strong assurance of trustworthiness to secure the IoT network is a non-trivial task. There exists challenges in incorporating blockchain-based TRM for IoT, which specifically depend on the IoT applications, as each application has its trust challenges with their unique features and requirements, which are discussed in the following section.

1.1 Thesis Motivation

This thesis considers three core functionalities of IoT ecosystems, namely access control and intrusion detection system, which are instrumental for securing IoT network; and resource sharing, which is essential in achieving effective utilisation of scarce resources in IoT networks. This section discusses specific trust issues in these core functionalities and motivates the need of blockchain-based TRM.

This section is organised as follows. First, Section 1.1.1 discusses the challenges in developing an IoT access control scheme. Second, Section 1.1.2 examines the trust issues in Collaborative Intrusion Detection Systems for IoT. Lastly, Section 1.1.3 presents the challenges in securing resource sharing in 6G-enabled IoT.

1.1.1 Trustworthy Dynamic Access Control

Protecting important resources from illegitimate access has been one of the priorities in securing computer systems. The protection mechanism, which is commonly referred to as access control or authorisation, decides when to grant or deny an access request from an authenticated user based on certain access policies, which determine what operations the user is permitted to perform [18]. There are various models of conventional access control, for instance, Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), and Capability-Based Access Control (CapBAC) [19], where in general access levels are mapped to certain metrics, namely user's roles, attributes, and capabilities (access tokens), respectively. While these conventional notions are also applicable to IoT ecosystems, incorporating the same in practical deployments raises concerns due to the nature of centralised architecture, which poses a potential risk of single point of failure.

Recent research has shown that blockchain has the potential to overcome a number of unresolved problems related to access control in IoT, such as single point of failure [20]. For instance, ABAC mechanisms were implemented in a decentralised fashion to provide more fine grained access control by recording attribute registrations and revocations in blockchain transactions [21]. In this scheme, access control is enforced by the resource owner by searching the blockchain for such records. In addition, the authors in [22] proposed a transparent access control mechanism wherein a list of static access rights are stored in the blockchain and enforced by a smart contract.

While these proposals have addressed the issues in authorisation systems, such as a single point of failure and lack of transparency, such static authorisation schemes are unable to automatically capture the dynamics of the network and adapt their authorisation policies accordingly. In the case of unwanted circumstances in the network, such as a node being compromised, static authorisation schemes would continue to enforce predefined access control policies that assume normal behaviour rather than making proper adjustments to account for compromised conditions. The inability of these static authorisation schemes to reinforce adjustments in the access control policies, in fact, raises a critical question of how to achieve a dynamic and trustworthy access control system without overlooking the fact that access control requires a sensitive consideration of who can access a resource.

1.1.2 Securing Collaboration in IDS

Intrusion Detection Systems (IDS) have been widely deployed as a means of securing IoT networks, with the goal of detecting malicious activity. In general, IDS can be categorised into two groups based on their underlying mechanics: signature-based; and anomaly-based. Signature-based IDS identifies incoming attacks by matching the network traffic with known intrusion signatures or rules. Anomaly-based techniques observe certain disparities in network traffic using machine learning approaches.

In recent years, an expansion of the attack surface has been inevitable, partially due to the adoption of IoT devices in diverse areas. This has consequently escalated the importance of defending the network from emerging threats [23]. Unfortunately, conventional IDS that work in isolation may be easily compromised, since they are unaware of new attacks which are not in their detection database. Researchers have thus proposed utilising multiple IDS to work together, referred to as CIDS, in which each IDS node shares its expertise and experience, e.g., alarms and attack signatures, to build collective knowledge of the recent attacks and increase the detection accuracy [24].

Researchers have recently explored the use of blockchain to enhance the performance of CIDS. For instance, blockchain is utilised to provide a transparent layer for sharing detection signatures[25]. The decentralised nature of blockchain also ensures fault tolerance and removes the need of a fully trusted third party in managing collaboration between CIDS nodes [26]. A peer-to-peer consensus algorithm in block generation is employed to build a trusted database of CIDS detection models [27].

However, proposals of blockchain-based CIDS overlook the importance of continuous evaluation of each node's trustworthiness and generally work based on the assumption that the nodes are always honest [28]. In fact, a trusted node may later be compromised and share untruthful detection rules to contaminate the rule detection database which would potentially expose the network to attacks [29]. To achieve trustworthy and effective collaboration in CIDS, continuous evaluation of trustworthiness of CIDS nodes is required, which can be derived from their collaboration behaviour. The shared detection rules database would grow significantly as the collaboration continues, making scalability another key factor to consider. Additionally, there is a need for secure and transparent trust mechanisms with the goal of providing auditability.

1.1.3 Secure and Trusted Resource Sharing in 6G-enabled IoT

The sixth generation of communication networks, colloquially referred to as 6G, is envisioned to bring unprecedented scale in network capacity, which offers extremely low latency, ultra high throughput and massive interconnected terrestrial and non-terrestrial networks [5], [30]. These levels of network capacity would realise futuristic applications in 6G-enabled IoT networks, such as reliable Internet of Healthcare Things and large-scale Vehicular IoT and autonomous driving [31]. With the ambitious requirements for realising the unprecedented network performance, 6G-enabled IoT demands effective utilisation of scarce resources, such as radio spectrum and computation infrastructures [32]. To this end, 6G-enabled IoT can adopt resource sharing techniques to maximise the utilisation of scarce resources [33]. In addition, resource sharing can alleviate the computation workload on constrained IoT devices, by offloading the computation tasks to more powerful nodes, thus helping prolong their battery life.

In recent years, blockchain has received significant attention from research and industry communities as an integral building block to realise the Dynamic Resource Sharing (DRS) [32]. By utilising consensus algorithms with fast convergence time, blockchain can be applied in resource sharing applications, which are latency sensitive. Due to the verifiable and transparent automation enabled by blockchain, DRS can significantly improve the resource utilisation, compared to the conventional resource allocation schemes, which tend to be static and inflexible. In addition, blockchain can effectively act as a trustless intermediary to facilitate communications between parties in resource sharing, making the coordination and cooperation more effective and efficient.

However, blockchain-based resource sharing still suffers from several challenges. While blockchain provides consistency and trust in the stored data within a resource sharing environment, blockchain alone cannot ascertain the trust in the participating nodes when performing resource sharing. The lack of trust would discourage nodes in the network to share or use other resources, as some malicious nodes may present and can potentially launch, for instance, a selfish attack, where a node forges the computation results. These trust issues would undermine the initial goal of resource sharing, as its effectiveness is reduced. Trust and Reputation Management (TRM) is an effective solution to tackle the trust and related security issues. However, current TRM approaches are not compatible with 6G-enabled IoT, as they suffer from inefficiencies in the computation of the reputation scores [6]. In addition, while blockchain introduces a level of anonymity with the use of pseudonyms, concealing the real identity to the public, some studies suggest that blockchain de-anonymisation is possible by linking transactions with the public keys [34], [35]. To increase the level of anonymity, users may employ different keys in each transaction, obfuscating their traces on the blockchain and reducing the risk of de-anonymisation attacks [36]. However, this may render the TRM unusable, as the keys are now changeable, making the same node not recognisable by a single key to which the trust and reputation scores are bound. In addition, existing privacy-preserving TRM frameworks only focus on concealing the shared information, but leaving the nodes' identities unprotected.

1.2 Thesis Contribution

The main contribution of the thesis is the adoption of blockchain-based TRM to solve the trustworthiness issues in the core functionalities of IoT ecosystems discussed

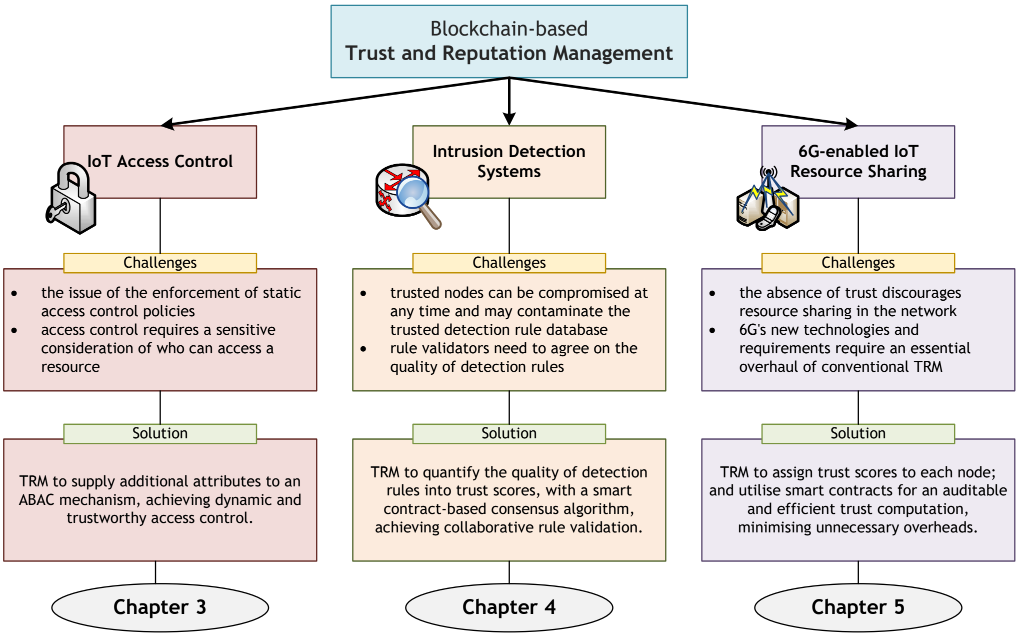
Blockchain-based
Trust and Reputation Management
IoT Access Control
Intrusion Detection Systems
6G-enabled IoT Resource Sharing
Challenges
the issue of the enforcement of static access control policies
access control requires a sensitive consideration of who can access a resource
trusted nodes can be compromised at any time and may contaminate the trusted detection rule database
rule validators need to agree on the quality of detection rules
the absence of trust discourages resource sharing in the network
6G's new technologies and requirements require an essential overhaul of conventional TRM
Solution
TRM to supply additional attributes to an ABAC mechanism, achieving dynamic and trustworthy access control.
TRM to quantify the quality of detection rules into trust scores, with a smart contract-based consensus algorithm, achieving collaborative rule validation.
TRM to assign trust scores to each node; and utilise smart contracts for an auditable and efficient trust computation, minimising unnecessary overheads.
Chapter 3
Chapter 4
Chapter 5

Figure 1.2: The graphical summary of the contributions of this thesis, which outlines the challenges in three core functionalities of IoT ecosystems and shows how the proposed TRM frameworks solve them.

in Section 1.1. Each functionality has its own trust issues and challenges with its unique features and requirements, which influence the design of the TRM framework. Figure 1.2 depicts the graphical summary of the thesis' contributions.

This section is organised as follows. Section 1.2.1 describes a dynamic access control framework for IoT. Section 1.2.2 presents a TRM framework for Collaborative Intrusion Detection Systems (CIDS). Section 1.2.3 outlines how resource sharing in 6G-enabled IoT can benefit from blockchain-based TRM.

1.2.1 Trust-based Blockchain Authorisation for IoT

A dynamic authorisation framework for IoT is proposed in Chapter 3, in which a decentralised ABAC scheme is enriched with a novel TRM design to address the challenges in building dynamic access control (see Section 1.1.1). The framework categorises the nodes into two groups, namely i) Service Consumers (SC) which request access to a particular resource and ii) Service Providers (SP) which own the resources. The framework quantifies both SC's and SP's behaviour into numerical measures, which are used as additional attributes for authorisation with the decentralised ABAC scheme.

In the proposed TRM, the trustworthiness and reputation of a node are calculated based on its adherence to the access control policies. Trust is defined as a subjective belief of a node's behaviour based on the previous interactions towards another node, which may help to determine the likelihood of the next interaction. On the other hand, reputation is seen as a global view of a node's past behaviour from aggregated trust relationships from multiple nodes [37]. In the proposed framework, both trust and reputation scores are transparently calculated by smart contracts on the blockchain. It is important to note that progressive evaluation of trust and reputation scores may help to detect and eliminate malicious or compromised nodes in the network [12].

The framework proposes a clear separation of the storage of sensitive information, such as users' attributes. A multi-tier decentralised architecture is designed, which consists of a main blockchain to enforce access control via smart contracts, and additional sidechains to store sensitive information securely.

A proof-of-concept of the proposed solution is implemented on a Rinkeby Ethereum test-network interconnected with a local IoT test-bed comprised of Raspberry Pi devices. However, the proposed design is agnostic to the blockchain platform with the only requirement being that it should support smart contracts.

1.2.2 TRM for Collaborative Intrusion Detection System

Chapter 4 proposes a trustworthy CIDS framework to address the challenges discussed in Section 1.1.2. The proposed framework continuously evaluates the trustworthiness of the CIDS nodes by evaluating the quality of the detection rules contributed by each IDS node. Each participating CIDS node can update its knowledge with the trustworthy detection rules to detect new attacks. The framework utilises a peer-to-peer decentralised storage to maintain a copy of the shared trustworthy detection rules, thus ensuring scalability. The participating nodes are divided into three categories, namely validator, contributor and regular nodes, each of which has a different role in the system.

The framework employs two smart contracts, namely Trust and Reputation Management (TRM) and Storage smart contract, to quantify each node's trustworthiness and manage the decentralised storage, respectively. The proposed framework offloads trust computation to the TRM smart contract which reduces the computation load for each CIDS node. A smart contract-based voting mechanism is designed to achieve collaborative detection rule validation and avoid an adversary from contributing deceptive detection rules. The framework is considered as a blockchain agnostic platform which can be implemented on any blockchain instantiation that supports smart contracts.

A proof-of-concept implementation of the proposed framework is implemented on a private Ethereum network hosted on a lab-scale testbed. The chapter evaluates the framework in terms of the evolution of trust scores, smart contract latency and Ethereum gas consumption. The results show that the the solution is feasible and performs within the expected benchmarks of the Ethereum platform.

1.2.3 TRM Framework for Resource Sharing in 6G-enabled IoT

A TRM framework for resource sharing in 6G-enabled IoT is proposed in Chapter 5, which specifically addresses the inherent trust issues that discourage resource sharing, discussed in Section 1.1.3. The chapter demonstrates how blockchain, with the aid of a TRM, could provide reliable assurance in the trust between participating nodes in the network, thus reducing the inherent risk of resource sharing. The chapter considers the use case of computation resource sharing in edge computing for 6G-enabled IoT, where the resource users offload computation tasks to resource owners, after which the trustworthiness of the resource owners are determined.

In the proposed framework, both resource owners and users are allowed to employ changeable keys to obfuscate their transaction traces in the network. To realise a privacy-preserving TRM, two interconnected blockchains are designed [38]. First, an Isolated Identity Chain (IIC), maintained by a set of semi-trusted authorities, where the trust score calculation takes place. Second, a Main Resource sharing Chain (MRC), which records the resource sharing transactions between the resource owners and users. Smart contracts are employed in both IIC and MRC to provide auditable trust calculation, where a recursive trust computation is developed to minimise unnecessary overheads, making the framework suitable for 6G networks. The proposed framework achieves rater and ratee anonymity, where the identities of both resource users (rater) and owners (ratee) are completely concealed.

A proof-of-concept implementation is developed on private Ethereum networks, where two Ethereum networks are deployed to realise IIC and MRC. However, the TRM framework is designed to be blockchain-agnostic, which can be implemented on any blockchain platform that supports smart contracts execution. The framework is benchmarked with a baseline TRM framework, where no privacy-preservation is implemented. The experimental results signify the feasibility of the proposed solution, as it only incurs minimal overheads with regards to gas consumption and overall latency.

1.3 Thesis Organisation

The thesis is organised as follows:

Chapter 2 discusses the background and literature review of the relevant work in the area of blockchain and TRM.

Chapter 3 presents a trust-based authorisation scheme for achieving dynamic access control for IoT, addressing the challenge in Section 1.1.1.

Chapter 4 proposes a TRM framework for securing collaboration between IoT nodes in CIDS, to tackle the trust issues discussed in Section 1.1.2.

Chapter 5 presents a TRM framework for securing resource sharing in 6G-enabled IoT, where the chapter addresses the specific issues in Section 1.1.3.

Chapter 6 concludes the thesis and presents several research directions for the future work.

Chapter 2

Background

This chapter presents the background and literature review of the relevant work in the area of blockchain and Trust and Reputation Management (TRM) for the Internet of Things (IoT). First, the notion of trust and reputation are introduced along with their relationship with the TRM in Section 2.1. Subsequently, Section 2.2 outlines the building blocks and properties of TRM for IoT applications. Section 2.3 presents an overview of blockchain technology, while its salient features that show promising potentials for enhancing TRM is discussed in Section 2.4. Section 2.5 discusses the relevant literature to the three core functionalities of IoT ecosystems, namely access control, intrusion detection system and resource sharing (see Section 1.1); and motivates the need for incorporating TRM as a solution to the issues. A summary of this chapter is presented in Section 2.6.

2.1 Trust and Reputation Management

Trust is a subjective and intangible belief about the behaviour of a particular entity or individual, which is built up from consecutive interactions [39]. Trust is context-related and thus cannot be generalised, as it is linked to a specific behaviour or traits. According to Gambetta, trust is defined as a subjective probability that an individual expects from another individual on performing an expected action [40]. Occasionally, trust and reputation are referred interchangeably in the literature. However, there is a subtle difference between these two terms. Trust refers to a subjective belief towards the behaviour of an entity that builds up as more interactions happen, while reputation can be seen as the aggregated opinion or trust degree of an entity from other entities that have prior interaction with the entity.

TRM aims to assess the accountability or trustworthiness of each participant in distributed systems by means of a quantitative approach. In TRM, trustworthiness is derived from direct experience or recommendations from other peers and is represented as numerical scores using which the trustworthiness level can be conveniently measured. In general, the trust and reputation score can be used as a safeguard to manage the associated risk in communicating with other peers in a distributed system, which might be very dynamic and hostile.

There have been many applications of TRM for IoT. For instance, in [41], a three-layered trust management framework is proposed, namely TrustChain, to address trustworthiness issues in supply chains. In general, the reputation system assesses the quality of the commodities based on multiple observations within the supply chain. In [42], the authors proposed a TRM to enhance data validation in crowd sourcing, wherein trust management is incorporated to select reliable validators to validate collected data based on the trustworthiness score of the participants in the crowd sourcing service.

However, TRM applications for IoT in the literature still suffer from several shortcomings. For instance, in [43], the underlying TRM architecture relies on a centralised actor to manage the collection of feedback and calculation of trust scores, which raises the risk of data loss and manipulation by the centralised party. If the centralised actor is compromised, an adversary may maliciously alter the trust computation thus undermining the use of these metrics. In addition, authentication and identification schemes in TRM may expose the actual identities of the devices' owners, which should be concealed and protected.

The following section outlines the building blocks and properties of a generic TRM that delineate how trust is empirically built by collecting and aggregating evidence of direct and indirect interactions to obtain a quantifiable trust measure. The following section then presents an overview of blockchain technology and discusses several blockchain properties that can help address the challenges in building TRM for IoT applications. This chapter also highlights the issues and challenges in three core functionalities of IoT ecosystems, namely access control, intrusion detection system and resource sharing; and motivates the necessity for incorporating TRM.

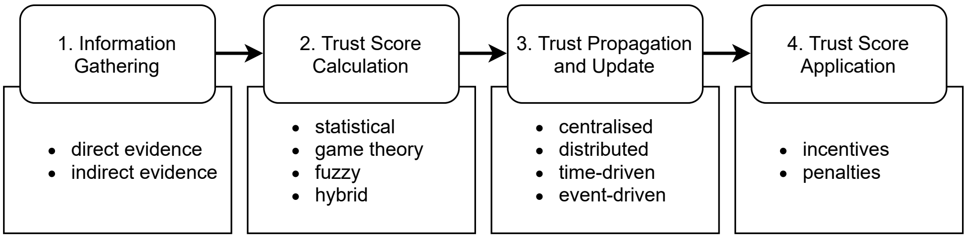

Figure 2.1: Trust derivation in a typical TRM framework, which is comprised of four major steps.

2.2 Properties of TRM

In general, IoT applications demand more distributed Device-to-device (D2D) communications, which involve a group of nodes collecting data from physical environments and performing specific tasks based on the collected data [5]. Consequently, there will be a greater need for IoT nodes to interact directly with other nodes. While the majority of nodes can be assumed to be honest, some nodes may behave opportunistically to maximise their gains through dishonest behaviour. In addition, the collected data may also be noisy, faulty or maliciously tampered. Ideally, an agent should not blindly trust other nodes due to these risks that may degrade the quality of service of their interaction. TRM are designed to quantitatively assess the trustworthiness of a particular agent or data in a system through numerical and tangible values. In IoT, TRM acts as an intermediary between service providers and requesters by providing measures of the trustworthiness of each participant. The following subsections describe the general properties of a generic TRM for IoT.

2.2.1 Trust Derivation and Application

Similar to real life social interactions, computational trust is built gradually from successive interactions between entities that correspond to positive or negative experiences, affecting the overall belief of the trustworthiness level. In a generic TRM, the interactions are assessed empirically, which includes four steps for collecting and applying trust computation, depicted in Fig. 2.1.

Information gathering The first step in TRM is defining the input parameters and attributes for quantifying or computing the trustworthiness level, which in general is highly application specific [44]. Some of the examples include adherence to communication protocols and quality of service. The TRM should gather all of these input values either by 1) direct observations or interactions, or 2) recommendation from other entities if prior interactions are unavailable.

Trust Score Calculation The next step includes the actual calculation of the trustworthiness level as quantifiable values or scores according to the preferred trust or reputation model. The TRM may use various computation models that suit the application requirements, for instance, statistical, game theory, fuzzy computation or hybrid. Note that, the input attributes also determine the appropriate computation model, e.g., sum and mean models are suitable for continuous input values, while Bayesian model is more suited for discrete binary values [14]. In addition, trust score calculation should also take into account the types of trust (see Section 2.2.2).

Trust Propagation and Update Typically, trust propagation can be performed in a centralised, distributed or semi-distributed fashion, depending on the underlying architecture of the system. The trust computation should be initiated based on temporal dynamics depending on the specific application, which includes time-driven and event-driven approaches [14]. In the time-driven approach, the trust score is updated on a regular basis, while the even-driven approach only requires updating the trust values upon new interactions and events.

Trust Score Application The specific manner in which the trust score is used depends on the requirements and operation of the application. Generally the trust score is employed to give certain quantified and fair measures for providing incentives or enforcing penalties [16], which may include certain privileges and monetary incentives or some restrictions and punishments.

2.2.2 Types of Trust

IoT applications rely on the data collected, processed and transferred within the system, and the interactions among entities. With regards to the computation type, trust can be categorised as follows:

Behaviour-based Trust Computation In behaviour-based trust computation, the trustworthiness level of an entity is derived from how the entity behaves in the system as perceived by a subject during its interaction with the entity. A subject identifies a positive behaviour if the observed entity conforms to the prescribed protocols and expectations, while negative behaviour corresponds to a deviation from protocols and expected behaviour.

Data-based Trust Computation In data-based trust computation, the trust values are calculated based on the quality of data provided by an entity. For instance, in IoT data trading [45], trust can be derived from the data quality acquired from the data provider. Here, trust grows with the authenticity of the data, i.e., deliberate manipulation, noise or anomaly in the data would degrade the trust. In this type of trust, data validation plays an integral role and one approach to validate the data quality may include using correlated observations obtained from other entities in proximity.

Hybrid Approach Relying on a single type of trust may not be sufficient for deriving trust in certain IoT applications, for instance, mobile crowd sourcing, wherein trustworthy nodes are seen as those who provide reliable data and conform to the predetermined governance. In such scenarios, trust can be computed considering both the data-based and behaviour-based characteristics. In the hybrid approach, weightings are used to give favourable emphasis on either data or behaviour characteristics.

2.2.3 Evidence Aggregation Approach

TRM may adopt one of the evidence aggregation approaches to accumulate trust evidence and calculate the final trust and reputation score. While there is an exhaustive list of aggregation approaches [46], the following approaches are among the most widely adopted:

Sum and Mean The most intuitive and popular aggregation approach in TRM is summation or average of the aggregated trust evidence [47]. Due to its simple operation, this method can also be validated manually to provide an objective confirmation. Some weighting parameters may also be incorporated to give more weight to recent or more important evidence. One of the challenges with this approach is the determination of appropriate weights, which would have an impact on the performance of the TRM.

Flow Network This approach is proposed in Advogato [48], wherein each participant is seen as a node in the network, while the interactions between participants are modelled as network flows. Consequently, the trust is derived from the number of flows a participant obtained from others. This approach is relatively robust to trust-related attacks, as the total active flows in the network are assumed to be constant and strictly regulated by the TRM.

Markov Chain As shown in EigenTrust [49], Markov chain model could be implemented to quantify the nodes' trustworthiness, where the approach works by modelling the probability of transitioning from one state to another as the feedback of a rater to a particular ratee. To determine the trustworthiness score of an unknown node, one needs to follow random transitions from a known trusted node to arrive at that particular node, using which the score is calculated as a probability function.

Bayesian In this approach, the trust and reputation score are computed using statistics. For instance, in [50], the trustworthiness score is described as a beta distribution of two parameters where α and β denote positive and negative recommendation, respectively. To calculate and update the score, an update to the provided beta distribution is performed, through which unfair ratings can also be removed.

2.2.4 Trust Dimensions

In general, trust is strongly attached to a particular context and generally cannot be transferred to another context without rigorous adjustment and re-calculation. With this regard, context-awareness is an important factor to consider in designing a TRM. A TRM can work on a single context or multiple-context awareness in deriving trust from collected evidence depending on its initial design [16]. Here, trust dimensions correspond to the number of contexts that a TRM supports, which can generally be classified into single-dimension (one context) and multi-dimension (multiple contexts).

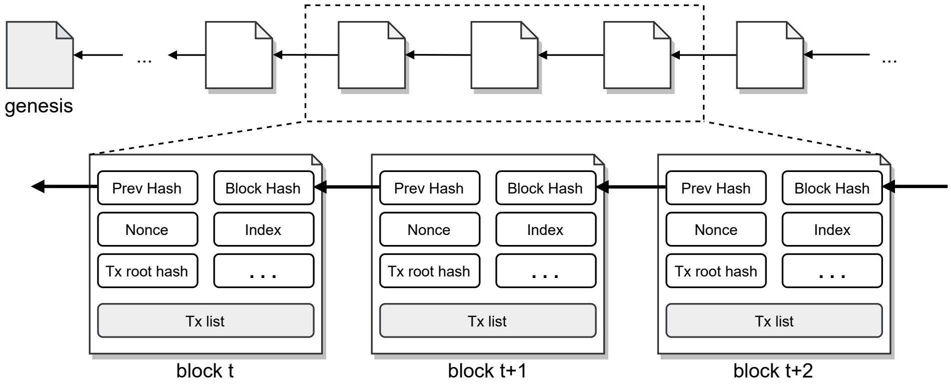

Figure 2.2: An overview of blockchain's immutable storage.

Single-dimension A lightweight TRM with a simplified trust and reputation model might only incorporate a single-dimension trust evaluation for the sake of limited resources in IoT applications. While single-dimension trust model may not be comprehensive, it may be preferred depending on the application design, e.g., when a majority of resource-constrained devices are in use.

Multi-dimension On the other hand, multi-dimensional trust and reputation model represents trust and reputation scores in multiple parameters or a single value derived from multiple parameters with appropriate weightings. In practice, this type of TRM may require extensive computation and may not be suited for constrained devices.

2.3 Blockchain Overview

Blockchain, the underpinning technology behind the popular cryptocurrency Bitcoin, has gained a lot of interest from both the industry and academia. Blockchain uses the notion of transaction, as the fundamental communication primitive, which records an exchange of assets, e.g., cryptocurrency, between two parties in the network. In essence, blockchain groups the transactions into a block by recursively hashing the transactions to generate a Merkle tree [51], from which the block hash is derived. Blockchain then chains multiple blocks together using their block hashes to form an immutable chain of interconnected blocks (sometimes referred to as ledger), depicted in Figure 2.2. As the blocks are chained to each other by their block hash,

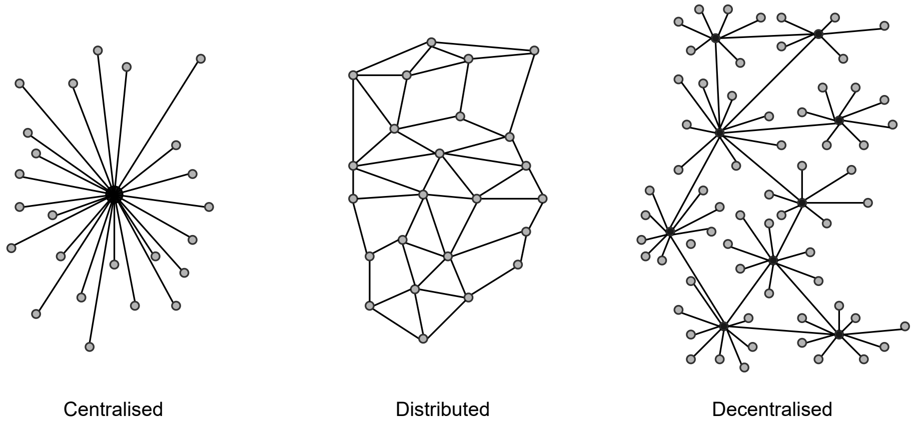

Figure 2.3: Different types of architecture: centralised, distributed and decentralised.

one should break the hash cryptography to tamper with the data; and may need to traverse all the way back to the block that contains the data to be tampered with, which is virtually impossible.

Blockchain is designed as a decentralised network (see Figure 2.3) and employs asymmetric cryptography, where each participating node is identifiable by its public keys. In general, blockchain is grouped into two categories according to the network access restriction. First, a public or permission-less blockchain is an open blockchain network which can be freely joined by any nodes. Second, a private or permissioned blockchain is a more restrictive network where access is only given to known parties.

In blockchain, the same ledger is distributed to all participating nodes in the network, achieving transparency and high availability. All nodes have the same privilege to append data to the ledger, removing the trust to a centralised entity. A consensus mechanism is utilised to write data to the ledger, where the majority of the nodes should agree upon the next block to be appended to the chain. There exists several consensus mechanisms for blockchain, such as Proof-of-Work (PoW), Proof-of-Stake (PoS) and Proof-of-Authority (PoA) [52].

Ethereum is an example of a permission-less blockchain instantiation which uses Ether as the native cryptocurrency [53]. Ethereum implements the notion of smart contract, which is a form of execution codes agreed by a set of users. Smart contracts allow deterministic and trusted execution of business logic with reliable guarantees that the process would be accomplished and validated collaboratively in the network [44]. With the introduction of Ethereum 2.0 [54], the former implementation of PoW is being replaced by PoS to enhance its overall performance.

Hyperledger Fabric [55], initially introduced by IBM, is an example of a permissioned blockchain network, where access to the network is managed by a consortium of organisations. Hyperledger Fabric follows a modular design approach, which gives more flexibility to construct the network, for instance, by allowing the consortium to incorporate a customisable consensus algorithm. In addition, the modular design of Hyperledger Fabric makes it suitable for building an enterprise-grade blockchain network. Hyperledger Fabric also provides support for smart contract execution, which is referred to as Chaincodes.

In recent years, blockchain has shown its potential applications beyond the financial domain [17]. With its salient features, such as transparency, tamper-resilient, verifiability, and pseudonymity, blockchain can be incorporated to TRM for improving its resiliency and address the aforementioned issues of conventional TRM. For instance, blockchain may replace the trusted centralised actor that assesses the trustworthiness of participants in traditional TRM.

2.4 Motivation for Blockchain Adoption

This section describes the inherent properties of blockchain that would bring enhancements and benefits to TRM.

Decentralisation Conventional TRM relies on a third party aggregator to collect trust evidence and calculate trust scores. Trusting a third party aggregator actually introduces significant risks, for instance, if the aggregator is compromised then any underlying processes of trust computation could be maliciously altered and the sensitive data could be in danger. Blockchain removes any Trusted Third Party (TTP) and comes with a decentralised architecture, which eliminates associated risks of employing third party aggregators in TRM. In addition, blockchain can also be incorporated in a distributed TRM to enhance the mechanism, for instance, by utilising smart contracts.

Smart Contract Smart contracts can be embedded into a blockchain-based TRM to perform collection and calculation of trust scores which can offload the trust computation from the IoT devices. For instance, a node may submit feedback to the smart contract about it's experience of interacting with a service provider, which later will be used to calculate the service provider's reputation score. Another node in the network can also query the smart contract to obtain the reputation scores of particular service providers. That is, a smart contract acts as a reliable intermediary for computing and querying trust related information.

Pseudonyms There is an inherent risk of leaking sensitive information in conventional TRM, as the authentication mechanism may link the identification details to real-life identities. Blockchain utilises an elliptic curve public key cryptography mechanism which utilises pseudonyms, i.e., public key, for identification purposes, resulting in higher privacy preservation, as real identities are not used. The use of pseudonyms is, to some extent, beneficial for protecting users' privacy, which is a desirable property in designing a TRM. In a blockchain-based TRM, each node is identifiable by its public key which hides any personal data, such as device ownership details.

Immutable Storage Traditional TRM stores trust evidence and interaction history on each device's internal memory, which may overwhelm the devices, especially if there is a large amount of information in a network with thousands of nodes. As discussed earlier, a traditional TRM can also rely on a TTP to keep track of the trust related information, but with fundamental risk of data loss and manipulation linked to the centralised approach. As discussed in Section 2.3, blockchain's data structure enforces immutability as it is difficult, if not almost impossible, to tamper with the on-chain data. With proper removal of any Personally Identifiable Information (PII), blockchain is a perfect and safe solution to store interaction evidence that would later be used to calculate the trust score.

Transparency In conventional centralised TRM, the process of computing the trust and reputation scores is performed by a centralised aggregator, which conceals the actual process from other participants in the network. On the other hand, blockchain offers transparent mechanisms in collaborative trusted execution of business logic via smart contracts and transparent immutable storage via a transparent shared ledger. With precautions in handling and storing sensitive information in the ledger, this type of transparent mechanism is preferred as it enables a traceable source of evidence where any participant can ascertain the integrity of a trust calculation by examining the ledger.

2.5 Blockchain-based TRM for Core Functionalities of IoT

This section presents the relevant work and motivates the need of incorporating TRM for three core functionalities of IoT ecosystems. First, Section 2.5.1 discusses the state of the art in blockchain-based IoT access control. Second, Section 2.5.2 describes the trust and security issues in blockchain-based Collaborative Intrusion Detection Systems (CIDS). Finally, Section 2.5.3 discusses the need of TRM for securing resource-sharing in 6G-enabled IoT application.

2.5.1 Blockchain-based IoT Access Control

Access control limits the actions a user may perform on a computer system, based on predetermined access control policies, thus preventing access by illegitimate actors. However, common authorisation schemes in IoT employ conventional schemes, which suffer from overheads and centralisation. There have been numerous efforts that study blockchain incorporation in IoT access control. These efforts utilise blockchain mainly for addressing the limitations of conventional access control mechanisms, such as reliance on trusted third parties and centralised processing.

In [56], Andersen *et al.* presented a scalable decentralised authorisation scheme, wherein access right is given to the requester via cryptographic access delegation method. A reverse-discoverable decryption mechanism is implemented to ensure the privacy of the access, which may span across different administrative domains. In [57], the authors proposed a fully decentralised IoT access control system comprised of four interconnected blockchains, namely, accountability, context, relationships, and rules blockchains. The framework stores entities and authorisation information as blockchain transactions and supports three types of access control mechanisms, namely Access Control List (ACL), CapBAC, and ABAC. In [58], a Lightweight Scalable blockchain (LSB) is proposed to deliver end-to-end security and access management for IoT via an ACL. LSB consists of interconnected and independent clusters, wherein a cluster head stores and maintains an ACL, using which access request is validated. Ding *et al.* [21] proposed a framework for decentralised ABAC, where blockchain transactions play a significant role in the authorisation and revocation of attributes. In their framework, the resource owner is responsible for enforcing the access control policies, wherein, the attribute validation process is performed by searching the blockchain. However, these proposals [21], [56]–[58] overlook the full potential of blockchain, e.g., leveraging the capabilities of smart contracts, and mainly rely on off-chain processes. These schemes are also not practical in a network comprised of a large number of connected devices, as the ACL requires manual updating to record the potential mobility of participating nodes.

To fully utilise blockchain's potential for IoT access control, some proposals have designed frameworks that aim to deliver decentralisation by means of smart contracts. Zhang *et al.* [22] proposed a framework that uses smart contracts to replace the centralised validation of access policies. The mechanism consists of three smart contracts, namely access control contracts, a judge contract, and a register contract, in which access control policies are stored as ACLs. Access control contracts enforce authentication and authorisation by checking if a user is allowed to access the resource based on the access policies. In [59], the authors proposed a new concept, called Delegation of Trust (DoT), which embodies the notion of trust that a resource owner has in a legitimate user to access the owner's resource. DoTs use identity based encryption and are issued by smart contracts in the blockchain. In [20], Novo designed an architecture that utilises a smart contract but only for managing access control in a permissioned blockchain. The design requires IoT device managers to execute a function call in the smart contract to authorise service consumers by checking if the access request aligns with the ACLs. Pal *et al.* proposed a blockchain-based access control mechanism for IoT with built-in support of access delegation enforced by smart contracts [60]. The mechanism leverages identity-less and asynchronous authorisation, in which access request is validated against privately stored attributes that preserve user's privacy. While these proposals [20], [22], [59], [60] use smart contracts to achieve decentralisation of access control logic, they are unable to capture the network dynamics that are inherent in IoT ecosystems.

Trust Computation for IoT Access Control

A trust computation model may protect against unfair or malicious activities from misbehaving nodes in a network [61]. To date, there have been some trust management protocols for IoT [62]–[71], some of which are mainly tailored for inclusion in IoT access control mechanisms. In [64], the authors proposed a trust computation model for a trust-aware access control mechanism for IoT. The model uses fuzzy computation and is calculated by a centralised trust manager. Gwak *et al.* proposed TARAS, an adaptive role-based access control for IoT that incorporates trust computation for granting or denying access without requiring any prior knowledge of the requester [65]. In [66], the authors designed a decentralised trust-aware access control mechanism for IoT, where a customised ABAC is employed with an additional trust management systems to quantitatively assess the trustworthiness of the requester according to the prior experience. Di Pietro *et al.* proposed an access control mechanism where both requester and requestee should agree upon particular terms and obligations on resource access, which also incorporate the requestee's global reputation [37]. However, the model [37] is relatively complex and inefficient, which is unsuited for IoT. Some proposals [65], [66] also disregard the importance of privacy preservation and centralised TTP processing is still used [64], [65].

Table 2.1 summarises the relevant work in IoT access control. In summary, little attention has been devoted to provide a secure TRM for access control in IoT. In fact, due to the scalability and dynamic interactions of IoT nodes in a network with intensive data streams and interactions, it is imperative to build a dynamic and flexible authorisation system, with the incorporation of blockchain-based TRM.

2.5.2 Collaborative Intrusion Detection System

Intrusion Detection Systems (IDS) have been widely deployed as a means of securing IoT networks, with the goal of detecting malicious activity. In general, IDS can be categorised into two groups based on their underlying mechanics: signature-based; and anomaly-based. Signature-based IDS identifies incoming attacks by matching the network traffic with known intrusion signatures or rules. Anomaly-based techniques observe certain disparities in network traffic using statistical or Machine Learning (ML) approaches.

In recent years, an expansion of the attack surface has consequently escalated the

Table 2.1: An overview of related work in access control for IoT.

Proposal	**Smart Contracts**	**Privacy Preservation**	**Trust Computation**	**Asynchronous Authorisation**	**Authorisation Type**
Novo [20]	●	○	○	○	ACL
Ding [21]	○	○	○	○	ABAC
Zhang [22]	●	○	○	○	ACL
Di Pietro [37]	○	EN	○	◐	Terms and obligation
WAVE [56]	○	EN	○	○	Authorization graph
ControlChain [57]	○	○	○	○	ACL, ABAC, CapBAC
LSB [58]	○	CK	○	○	ACL
WAVE [59]	●	EN	○	◐	Permission graph
Pal [60]	●	SC	○	●	ABAC
TACIoT [64]	○	SY	●	○	XACML
TARAS [65]	○	○	●	○	RBAC
IoT TM [66]	●	○	●	○	ABAC

Legend

○	Not supported
◐	Partially supported
●	Supported
EN	Encryption
CK	Changeable Keys
SC	Side chains
SY	Security Policy
ACL	Access Control List
ABAC	Attribute Based Access Control
CapBAC	Capability Based Access Control
XACML	eXtensible Access Control Markup Language

importance of defending networks from emerging threats [23]. Researchers have thus proposed utilising multiple IDS to work together, referred to as Collaborative-IDS (CIDS), in which each IDS node shares its expertise and experience, e.g., alarms and attack signatures, to build collective knowledge of the recent attacks [24].

Blockchain has the potential to enhance the performance of CIDS, as it can provide a transparent layer for sharing detection signatures. The decentralised nature of blockchain also ensures fault tolerance and removes the need of a fully trusted third party in managing collaboration between CIDS nodes.

There have been several proposals in the literature about blockchain incorporation in CIDS. Li *et al.* proposed a framework called CBSigIDS which aims to avoid insider attacks, where malicious nodes can generate untruthful signatures to contaminate the network [25]. Here, blockchain is incorporated to provide a mechanism for sharing detection signatures between different IDS nodes in a verifiable manner. CBSigIDS also implements trust computation to evaluate the reputation levels of the IDS nodes by means of a challenge-based trust mechanism, wherein the CIDS nodes are required to send challenge messages to known acquaintances to assess the trustworthiness level in detecting known attacks. While the authors incorporated blockchain, CBSigIDS does not utilise smart contracts to take full advantage of blockchain.

In [72], blockchain is incorporated in a CIDS solution to store and disseminate calculated trust scores with the underlying evidence that justifies the trust calculation. The solution mainly aims to enhance the overall security by recording misbehaving CIDS nodes. In [73], a collaborative Host-based IDS (HIDS) framework is proposed to overcome advanced insider attacks with the help of a TRM. The framework uses spatial correlation via a hybrid trust management, which quantifies the trustworthiness of a CIDS node into three levels, namely trust, expertise and behavioural trust level. Similar to [25], these solutions [72], [73] employ a challenge-based trust evaluation. However, the solution does not explore incentives and penalties as a reward mechanism and no performance evaluation has been undertaken.

Blockchain has also been proposed for building a CIDS framework in Software Defined Network (SDN) environment, where SDN controllers act as CIDS nodes [74]. The framework employs permissioned blockchain to guard against adversaries that attempt to manipulate detection signatures by sharing untruthful detection signatures to the participating SDN controllers. The authors proposed digital certificates to establish trust between SDN controllers by designing a scheme called Certificate-Chain (C-Chain) which is essentially a blockchain-based distributed Public Key Infrastructure (PKI). However, the framework does not quantitatively evaluate the SDN controllers' trustworthiness. In addition, the framework utilises IPFS as a storage medium for detection signature files. However, the framework does not impose penalties for fraudulent manipulations.

Golomb *et al.* proposed CIoTA, a HIDS framework which aims to address the issue of false positives and adversarial attacks in an anomaly-based IDS. The framework utilises ML and blockchain to collaboratively build a trusted anomaly detection model, using which each HIDS node can update their detection database. As the framework is aimed for deployment on IoT devices, it is designed to be lightweight and scalable. However, the framework does not incorporate trust management to evaluate HIDS nodes and determine the quality of the contributed anomaly detection model.

Alkadi *et al.* proposed a CIDS platform for cloud based IoT networks, wherein blockchain facilitates immutable data exchange between several cloud services [75]. In contrast to signature-based methods, the platform combines blockchain with a Deep Learning (DL) technique to provide a secure CIDS with smart contracts in cloud based IoT networks. The platform utilises a consortium blockchain with a Trusted Execution Environment (TEE) for securely logging the transactions between multiple cloud providers, while a Bidirectional Long Short-Term Memory (BiLSTM) DL algorithm is trained to detect anomalies in the network. However, this DL platform does not incorporate trust management for evaluating cloud providers (CIDS nodes) and does not explore incentive mechanisms for the collaborations.

Another proposal that uses DL for blockchain-based CIDS is presented in [26]. The authors designed a collaboration framework where blockchain is employed to consistently train and test detection models for anomaly-based CIDS. The proposed blockchain-based CIDS aims to achieve a lifetime learning framework, which is able to gradually build a secure and co-maintained database of labelled training set for classification. The framework adopts Growing Hierarchical Self-Organising Map with probabilistic relabelling (GHSOM-pr) as the off-chain classifier, which can adapt to the dynamic nature of the co-maintained database. While the framework introduces Data Coins (DCoins) as the incentives for collaboration, it does not incorporate smart contracts and include mechanisms for trustworthiness evaluation.

Table 2.2 summarises the relevant work in CIDS for IoT. While various techniques have been proposed to secure CIDS networks, some issues remain unsolved. For instance, current trust management mechanisms for CIDS employ a challenge-based method that was initially designed for HIDS [24]. Challenge-based trust management requires each node to calculate and store trust scores on the device itself, which could be seen as redundant. While the scheme may work well for medium sized networks, challenge-based techniques would be impractical and raise scalability issues when the number of CIDS nodes are relatively large.

2.5.3 Blockchain-based Resource Sharing for 6G-enabled IoT

Several studies highlight that blockchain would be an integral part of the envisioned 6G networks [32], [33]. With the help of the salient features of blockchain, for instance, decentralisation, verifiability and transparency, 6G networks can attain a trusted and transparent platform for pooling and managing scattered resources with enhanced efficiency and security [32]. The trusted automation in blockchain realises the notion of DRS, in which radio and computation resources are dynamically allocated according to the recent resource supply and demand, improving the resource utilisation rate compared to the conventional static resource allocation schemes [33].

Within the resource sharing context, TRM is essential to provide trust scores for all nodes in the network, using which a node can determine the trustworthiness and reliability of the target nodes for task offloading. There have been several efforts in incorporating TRM for blockchain-based resource sharing schemes. Iqbal *et al.* proposed a blockchain-based reputation management for resource sharing in 5G IIoT environment, where network edges maintain the blockchain and calculate reputation scores [76]. In the framework, constrained devices utilise nearby fog nodes instead of cloud nodes for task offloading (i.e., resource sharing), due to the proximity and

Table 2.2: An overview of related work in CIDS for IoT.

Proposal	**Blockchain type**	**Consensus Algorithm**	**Smart Contracts**	**IDS Deployment Type**	**IDS Detection Type**	**Scalability**	**TRM**	**Trust Mechanism**
Fung *et al.* [24]	○	○	○	H	AB	P	●	Challenge-based
CBSigIDS [25]	R	W	○	N	SB	P	○	not supported
Liang *et al.* [26]	R	C	○	N	AB	M	○	not supported
CIoTA *et al.* [27]	R	V	○	H	AB	M	●	Extensible Markov Model
Kolokotronis *et al.* [72]	R	W, S	○	N	SB	P	●	Challenge-based
Li *et al.* [73]	○	○	○	N	SB	P	●	Hybrid challenge-based
Fan *et al.* [74]	R	F	●	N	SB	M	○	Assumes that PKI provides trust
Alkadi *et al.* [75]	R	W	○	H, N	AB	M	○	not supported

Legend

○	Not supported
●	Supported
R	Private (Permissioned) blockchain
W	Proof-of-Work
S	Proof-of-Stake
C	Proof-of-Classification
V	Voting-based
F	PBFT
H	Host-based IDS (HIDS)
N	Network-based IDS (NIDS)
AB	Anomaly-based
SB	Signature-based
P	Poor scalability
M	Moderate scalability

latency. TRM is employed to achieve a secure and trusted collaborative computing service, in which the trust and reputation scores are used as the base for device selection for task offloading. To quantify the trustworthiness, the network edges compare the actual task completion time with the expected completion time.

Bellaj *et al.* proposed BTrust, a blockchain-based trust overlay network for collection and dissemination of feedback in a large scale peer-to-peer network [77]. In this work, the trust and reputation scores are calculated from several weighted parameters, which is computed and disseminated by the smart contracts. The trust framework employs random walk function to determine the trustworthiness level of the neighbouring peers without overloading the network with broadcast messages.

In [78], a TRM framework is proposed to tackle the challenges in establishing trust during resource sharing process in the Internet of Vehicles (IoV), where it follows task offloading approach to share resources. The framework proposes the notion of Proof of Reputation, implemented in a consortium blockchain network, to reduce the latency and computation in the consensus process, as IoV devices typically have high mobility and limited computational resources. The framework proposes resource pricing mechanism using Deep Reinforcement Learning (DRL) which is implemented in the smart contract. The reputation scores of the vehicles are calculated using Gompertz function, using which the task owners can select where to offload the computation tasks.

Xiao *et al.* proposed a blockchain-based trust mechanism for Mobile Edge Computing (MEC) with the aim to evaluate the performance of edge network [79]. The resource sharing framework follows task offloading approach to share resources, where a DRL allocation algorithm is employed to optimise the task allocation. With the implementation of a TRM, the framework could detect and punish selfish edge devices which do not truthfully compute the offloaded task and forge the offloading report.

While blockchain, to some extent, offers partial privacy-preservation with the use of pseudonyms for authentication, some studies suggest that de-anonymisation is still possible [35], highlighting the need of further effort on privacy-preservation. These solutions [76]–[79], however, overlook the importance of extending privacy preservation beyond blockchain. In addition, these solutions cannot be directly ported to 6G due to their inefficient trust computation model, which requires an iteration through the full history of trust evidence, resulting in unnecessary overheads especially when applied to the scale of 6G [6].

Several technologies have been utilised to achieve privacy-preserving TRM on top of blockchain, such as blind and ring signature. In [80], the authors propose BPRF, a privacy-preserving TRM framework for participatory sensing, where the reputation scores are derived from the veracity of the sensing data and the feedback. BPRF utilises group signatures to unlink each transactions and conceal the actual identity of the group member when submitting sensing data.

In [81], a privacy-preserving TRM is proposed with the aim of preserving the identity of raters when submitting feedback about a particular Service Provider (SP). To receive feedback, an SP issues blinded tokens, using which a rater can conceal its actual identity when submitting feedback. As such, the raters can submit the feedback without being identified by the SP, thus achieving unlinkability between the feedback and the raters. The reputation of an SP can be calculated by aggregating the feedback which is stored publicly on the blockchain. The framework does not specifically define the feedback aggregation method, which means any aggregation function can be used to calculate the reputation score (see Section 2.2.3).

A privacy-preserving trust model for resource sharing is proposed by Ye *et al.* [82], where the proposed model addresses the trust issues in dynamic spectrum access for IoT networks. Here, the model evaluates the reliability of sensing nodes when performing collaborative sensing to sense unused radio spectrum. Specifically, the ring signatures and commit-and-reveal scheme are employed to protect the actual geolocation information of the sensing nodes, protecting their privacy.

Lu *et al.* [83] proposed BARS, a blockchain-based anonymous TRM for Vehicular Ad-hoc Networks (VANET), in which vehicles can submit messages about the current road conditions and the surrounding vehicles can provide feedback about the veracity of the messages. The trustworthiness of each vehicle is determined by the validity of its submitted messages, which are scores computed based on the historical interactions (direct evidence) and aggregated opinions about a particular vehicle (indirect evidence). The privacy preservation works by exploiting lexicographic Merkle trees as an extension to the typical blockchain authentication mechanism.

Table 2.3 summarises the relevant work in resource sharing for IoT. In summary, previous studies mainly focus on providing a trust management in resource sharing to address the inherent trust issues, without putting adequate attention to conceal the node identities to encourage resource sharing (i.e., achieving complete anonymity for both rater and ratee). Most privacy preserving mechanisms only focus on concealing the shared information but leaving some of the nodes' identities unprotected.

2.6 Chapter Summary

This chapter presents a background in blockchain for TRM in IoT. Firstly, the notions of trust and reputation are discussed as the metrics to quantify trustworthiness in IoT networks. Then, the properties and building blocks of generic TRM for IoT applications are described, along with the underlying motivation for blockchain adoption in developing TRM for IoT. Lastly, an elaborated discussion is presented about the state of the art in three core functionalities of IoT ecosystems, namely i) access control, ii) intrusion detection system and iii) resource-sharing, which motivates the necessity to incorporate TRM in the IoT network.

Table 2.3: An overview of related work in resource sharing for IoT.

Proposal	**Smart Contracts**	**Scalability**	**TRM**	**Efficient Score Calculation**	**Privacy Preservation**	**User Anonymity**	**Privacy Preservation Method**
Iqbal *et al.* [76]	○	P	●	○	◐	◐	blockchain's pseudonyms
Bellaj *et al.* [77]	●	M	●	◐	◐	◐	blockchain's pseudonyms
Chai *et al.* [78]	●	P	●	○	◐	◐	wallet address
Xiao *et al.* [79]	○	P	●	○	◐	◐	blockchain's pseudonyms
Jo and Choi [80]	●	P	●	○	●	◐	group signatures
Schaub *et al.* [81]	○	P	●	○	●	◐	blind signatures
Ye *et al.* [82]	●	P	●	○	●	◐	ring signatures
Lu *et al.* [83]	○	P	●	○	●	◐	lexicographic Merkle trees

Legend

○	Not supported
◐	Partially supported
●	Supported
P	Poor scalability
M	Moderate scalability

Chapter 3

Trust-based Blockchain Authorisation for IoT

This chapter presents a TRM framework to address the issues in blockchain-based IoT access control, i.e, authorisation, as discussed in Section 1.1.1 and 2.5.1. In general, conventional blockchain-based authorisation schemes enforce static predefined access policies with the assumption that authenticated and authorised nodes would exhibit benign behaviour throughout the operational period. The assumption overlooks the fact that the authenticated nodes may be compromised post-authentication and impede the reliability and resilience of the network, as shown in the case of Mirai botnet. The inability of these static authorisation schemes to reinforce adjustments in the access control policies, in fact, raises a critical question of how to achieve a dynamic and trustworthy access control system. In addition, the dynamic access control scheme should not overlook the fact that access control requires a sensitive consideration of who can access a resource. To do so, the proposed framework exploits TRM to supply additional attributes to an attribute-based access control mechanism. The framework progressively quantifies the trust and reputation scores of each node in the network and incorporates the scores into the access control mechanism to achieve dynamic and flexible access control. The solution is implemented on a public Rinkeby Ethereum test-network interconnected with a lab-scale testbed of Raspberry Pi computers. The evaluations consider various performance metrics to highlight the applicability of the proposed solution for IoT contexts.

3.1 Introduction

Authorisation or access control limits the actions a user may perform on a computer system, based on predetermined access control policies, thus preventing access by illegitimate actors [18]. Access control for the Internet of Things (IoT) should be tailored to take inherent IoT network scale and device resource constraints into consideration [20]. However, common authorisation systems in IoT employ conventional schemes, which suffer from overheads and centralisation. Recent research trends suggest that blockchain has the potential to tackle the issues of access control in IoT [22]. However, proposed solutions overlook the importance of building dynamic and flexible access control mechanisms.

In this chapter, a dynamic authorisation framework is proposed, in which a decentralised Attribute Based Access Control (ABAC) system is enriched with a novel TRM. The approach quantifies both Service Consumer (SC) and Service Provider (SP) behaviour while also simplifying the trust and reputation score computation using recursion. In the proposed TRM, the trustworthiness and reputation of a node are calculated based on their adherence to the access control policies. Trust is defined as a subjective belief of a node's behaviour based on the previous interactions towards another node, which may help to determine the likelihood of the next interaction. On the other hand, reputation is seen as a global view of a node's past behaviour from aggregated trust relationships from multiple nodes [37]. In the proposed TRM, both trust and reputation scores are transparently calculated by smart contracts in the main public blockchain. It is important to note that progressive evaluation of trust and reputation scores may help to detect and eliminate malicious or compromised nodes in the network [12]. In addition, the framework proposes a clear separation of sensitive information storage, such as users' attributes. The framework employs a multi-tier decentralised architecture, which consists of a main public blockchain to enforce access control via smart contracts, and additional private sidechains to store sensitive information securely. A proof-of-concept of the proposed solution is implemented and tested on a Rinkeby Ethereum test-network interconnected with a local IoT test-bed comprised of Raspberry Pi computers. However, the proposed framework is blockchain-agnostic, which means it can be implemented to any blockchain platform with the only requirement being that it should support smart contracts.

3.1.1 Chapter Contributions

In summary, the contributions of this chapter are as follows:

- A decentralised IoT access control framework is proposed, which is based on ABAC with an additional TRM, dubbed as blockchain-based trust management (BC-TRM), to capture the dynamics of the network.
- The framework separates the storage of sensitive information, e.g., user's attributes, and the public TRM data, by incorporating a main public blockchain and additional private sidechains, and supports asynchronous authorisation.
- A lightweight TRM is designed, which involves a simple recursive calculation that captures bi-directional interactions of SC and SP.
- A proof-of-concept implementation of the proposed solution is developed on a public Rinkeby Ethereum test-network interlinked to a lab-scale IoT test-bed. To demonstrate the practicability of the proposed solution in the IoT context, the experiments evaluate the solution in terms of trust and reputation score evolution, authorisation latency, and Ethereum gas consumption.

3.1.2 Chapter Organisation

The rest of the chapter is organised as follows. The proposed system model, TRM, and access control mechanism are described in Section 3.2, 3.3, and 3.4, respectively. Section 3.5 presents the proof-of-concept implementation with the corresponding experimental results. The findings are discussed in Section 3.6 and the chapter is concluded in Section 3.7.

3.2 System Model

In this section, we illustrate the proposed decentralised trust management for IoT access control and describe the main components and threat model of our proposed system in detail.

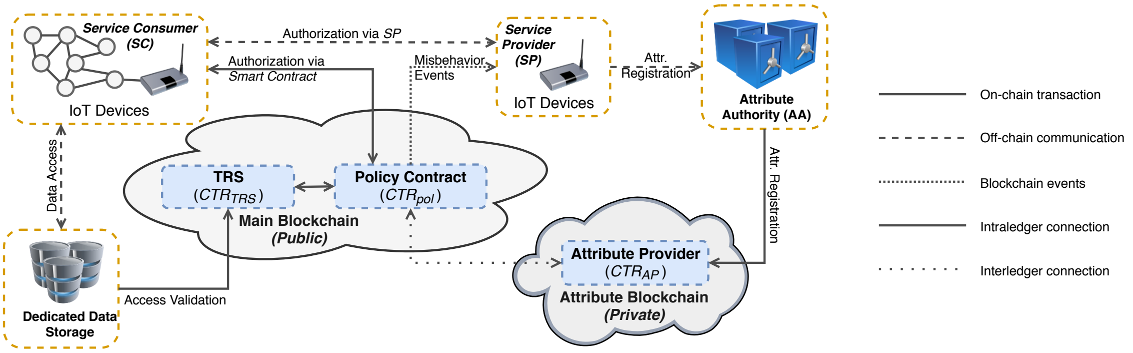

Figure 3.1: Architecture overview.

3.2.1 Main Components

We consider a network model, wherein each IoT device is registered to a local regulator that maintains a list of technical specifications and ownership information of the corresponding device (i.e., device attributes). To manage and limit who can access which resources under certain conditions, an attribute-based access control mechanism is employed. Figure 3.1 illustrates our proposed trust management and decentralised access control architecture. To support separation of sensitive information, we utilise two types of blockchain networks, a public and a private blockchain, in which different categories of data are stored.

Service Providers and Service Consumers

In general, the IoT nodes are grouped into two categories, namely SPs and SCs. Typically, SPs own a set of resources, denoted r, which can be accessed by others for free or by paying for a small access fee. Meanwhile, SCs are user devices interested to consume resources r owned by SPs. For identification purposes, we use Public Keys (PKs) for both SPs and SCs. The IoT devices run a light blockchain client to directly connect to the public blockchain network [84], which implies that the devices should have sufficient resources to support asymmetric cryptography. In instances where devices are resource constrained, they may opt to use a third party service or rely on their communication gateway for providing connectivity to the blockchain.

Dedicated Data storage

The off-chain Dedicated Data Storage (DDS) stores high volume of data over a longer timespan. The SPs store and update the data in a regular basis with an additional signature of SP to ensure integrity. We assume that there are multiple DDS in the network with sufficient redundancy level to provide high availability and maintain scalability. An SC may request access to the main blockchain for obtaining a legitimate access token for accessing the data in the DDS. We describe the access control framework in more detail in Section 3.4.

Attribute Authorities

We use inherent attributes of an IoT device (e.g., sensor types and hardware specifications) to determine whether a node is allowed to access a resource, as defined in an ABAC scheme. The Attribute Authority (AA) is responsible to issue legitimate attributes to the participating IoT nodes, according to the prescribed guidelines that conform to hardware specifications and ownership information. There are multiple AAs that form decentralised attribute authority networks, using which an SC can request for attributes issuance. We describe the attribute registration process in detail in Section 3.4.1.

Blockchain Networks

In our proposed model, we implement a single main public blockchain, denoted MB, which provides decentralised and collaborative trusted execution of access control logic and tamper-proof trusted data storage. As publicly revealing IoT attributes may pose some privacy concerns, we implement a set of permissioned blockchains, $PB = \{pb_1, pb_2, pb_3, \ldots, pb_N\}$, to maintain a private immutable list of attribute records. While MB is maintained by independent miners that are motivated to gain financial benefits from mining the blocks, each pb_k is maintained by a consortium of independent attribute authorities $AA_k = \{AA_k^1, AA_k^2, \ldots, AA_k^Y\}$, in which access to pb is limited to the corresponding AAs. Note that, here AA is partially-trusted, i.e., only $(y-1)/3$ out of available y AAs are untrusted for PBFT fault tolerance (see Section 3.2.2). In addition, each pb is interlinked to the main blockchain, hence, acting as a sidechain for supplying attribute information to the main blockchain.

Note that MB can also be implemented as a permissioned blockchain network, in which the participants may be known in advance and may be partially trusted. However, a more stringent access restriction should apply to PB to avoid leakage of sensitive information, i.e., access to PB is given only to the corresponding AAs.

We deploy two public smart contracts to MB, namely TRM (CTR_{TRM}) and Policy Contract (CTR_{pol}), which store the logic of trust calculation and access policy validation, respectively. In addition, a private smart contract, namely Attribute Provider Contract (CTR_{AP}), which resides in each *pb*, is responsible for attribute registration and validation. As such, we have N number of CTR_{AP}, i.e., $\{CTR^1_{AP}, CTR^2_{AP}, \ldots, CTR^N_{AP}\}$. We employ a bridging mechanism to interlink MB and PB, with the assumption that both types of blockchains use the same public key cryptography mechanism to handle authentication and signature process [85]. The bridging mechanism works by matching the PKs from both blockchains.

3.2.2 Threat Model and Assumptions

In our architecture, the adversaries can be SPs or SCs that are maliciously intent on disrupting the network. We group our threat model into three categories, namely access control policies attacks, reputation attacks, and other network protocol attacks. Firstly, a malicious SC may perform attacks related to access control, for instance, performing a replay attack in which the adversary maliciously captures and reuses an access token to gain illegitimate access. A malicious SC may also successively try to access the data using forged and expired access tokens or to leverage access rights, which would result in network congestion, i.e., a DoS attack. In addition, a malicious SP may be unreliable in providing the data, i.e., an SP does not fulfil its obligation in providing frequent updates of sensor readings. Secondly, the adversaries are also capable of performing the following types of reputation attacks:

- *Self-promoting attacks:* As an IoT node may act as both SP and SC, a malicious actor might try to increase its own reputation score by providing positive feedback to itself.
- *Bad-mouthing attacks:* A malicious SC_i may attempt to ruin the reputation of an SP_j by constantly providing negative feedback regardless of the quality

of the service. Moreover, an SC_i may also attempt to ruin the trust and reputation of an SC_k by requesting illegitimate access on behalf of SC_k.

- *Ballot-stuffing attacks:* A malicious SC_i may collude with SP_j to increase their reputation scores.
- *Whitewashing or newcomer attacks:* An SC attempts to rejoin the network using a new identity aiming to maliciously erase its previously recorded bad behaviour and obtain a fresh reputation score.

Third, we presume other types of network protocol attacks may emerge. However, those attacks are handled by established intrusion detection mechanisms and thus beyond the scope of this chapter, while the other two groups of attacks are handled by our TRM mechanism.

As our system model primarily runs on top of a commodity blockchain platform, we assume that the blockchain is secure against peer-to-peer and consensus attacks, such as eclipse, Sybil (on the consensus layer), and 51% attacks (for PoW-based consensus) [86]. We further presume that AAs are partially trusted and are not required to be always connected to the system. However, to maintain PBFT fault tolerance in *pb*, we require $3f + 1$ online AAs at any given time, where f is the number of faulty or untrusted AAs in *pb*. We also assume that DDS is secure and allows only authorised users to access the stored data.

3.3 Trust and Reputation Management

We design the proposed TRM to achieve a dynamic and self-adaptive authorisation system which is capable of capturing the dynamics of the network and detecting and eliminating malicious or compromised nodes, as explained in our threat model (Section 3.2.2). If a malicious activity occurs, either CTR_{pol} or CTR_{TRM} will blacklist the offending node from the network and notify all network participants via blockchain events.

Figure 3.2 illustrates the trust relationship model between the CTR_{TRM} and IoT nodes (SP and SC). The trust score of SP_j towards SC_i, denoted $T_{SC_i}^{SP_j}$, is calculated by CTR_{TRM} based on their previous interaction, which corresponds to a binary experience, i.e., either positive or negative. A positive experience is an honest action of SC_i to SP_j that conforms to the access control policies, while negative experience is otherwise. An initial value of 0 is assigned to $T_{SC_i}^{SP_j}$, if there are no prior interactions of $SC_i \rightarrow SP_j$. We calculate $T_{SC_i}^{SP_j}$ after t interactions as follows:

$$T_{SC_i}^{SP_j}(t) = (1-\gamma)\sum_{m=1}^{t} \delta_m \gamma^{t-m} \tag{3.1}$$

where

$$\delta_m = \begin{cases} \delta_{pos} & \text{if } \ m^{th} \ \text{ interaction was positive} \\ \delta_{neg} & \text{otherwise} \end{cases}$$

where γ is the ageing parameter ($0 < \gamma < 1$) which affords more weight to recent observations than older ones. We exclude 0 and 1 from γ, as these extreme values would always give static $T_{SC_i}^{SP_j}(t)$ regardless of the observations (see Fig. 3.4). Next, $\delta_{pos} > 0$ is the weight associated with a positive interaction, while $\delta_{neg} < 0$ represents the weight for negative interactions. We choose $\delta_{pos} < |\delta_{neg}|$, to make it harder to build trust than to lose it, as how humans perceive trust in real-life social relationships.

Note that in the extreme case where all interactions are positive, we have:

$$T_{SC_i}^{SP_j}(t) = (1-\gamma)\delta_{pos}\sum_{k=0}^{t-1} \gamma^k \tag{3.2}$$

As $t \rightarrow \infty$ and all interactions are positive, we have the limiting case:

$$\begin{aligned} T_{SC_i}^{SP_j}(\infty) &= (1-\gamma)\delta_{pos}\sum_{k=0}^{\infty} \gamma^k \\ &= \delta_{pos} \end{aligned} \tag{3.3}$$

Similarly, in the other extreme case where all experiences are negative, we obtain $T_{SC_i}^{SP_j}(\infty) = \delta_{neg}$. The trust scores are therefore bounded by these two limiting cases:

$$\delta_{neg} \leq T_{SC_i}^{SP_j}(t) \leq \delta_{pos} \tag{3.4}$$

where $T_{SC_i}^{SP_j} = \delta_{neg}$ is the score when SC_i is totally untrusted and $T_{SC_i}^{SP_j} = \delta_{pos}$ is the score when SC_i is completely trusted by SP_j.

Besides being conveniently bounded, the trust score has another advantage, in that it is computable by simple recursion:

$$\begin{aligned} T_{SC_i}^{SP_j}(t+1) &= (1-\gamma)\sum_{m=1}^{t+1}\delta_m\gamma^{t+1-m} \\ &= \gamma(1-\gamma)\sum_{m=1}^{t}\delta_m\gamma^{t-m} + (1-\gamma)\delta_{t+1} \\ &= \gamma\, T_{SC_i}^{SP_j}(t) + (1-\gamma)\,\delta_{t+1} \end{aligned} \tag{3.5}$$

We follow a similar approach as in Eq. (3.1) to calculate the trust score of a particular SP from an SC's standpoint, but use different weighting that results in differences in the evolution of trust. The trust score of SC_i towards SP_j, denoted $T_{SP_j}^{SC_i}$, is derived from accumulated feedback from SC_i after receiving the service, i.e., accessing a resource owned by SP_j. Similarly, the feedback corresponds to a binary expression of either positive or negative experience. The trust score of $T_{SP_j}^{SC_i}$ after t feedback instances is calculated as follows:

$$T_{SP_j}^{SC_i}(t) = (1-\mu)\sum_{n=1}^{t}\varepsilon_n\mu^{t-n} \tag{3.6}$$

where

$$\varepsilon_n = \begin{cases} \varepsilon_{pos} & \text{if } n^{th} \text{ feedback was positive} \\ \varepsilon_{neg} & \text{otherwise} \end{cases}$$

where $0 < \mu < 1$, $\varepsilon_{pos} > 0$, $\varepsilon_{neg} < 0$, and $\varepsilon_{pos} < |\varepsilon_{neg}|$.

We adopt Gompertz function to model the reputation growth [87]. As in real-life social interaction, reputation increases gradually after successive positive interactions and drops significantly after a negative interaction. In general, the reputation R_{SC_i} of SC_i is calculated by feeding the aggregation of its trust scores $A_{SC_i}(t)$ across different SPs to the Gompertz function, as follows:

$$A_{SC_i}(t) = \frac{\ln|Peers_i(t)|}{|Peers_i(t)|}\sum_{SP_j\in Peers_i(t)} T_{SC_i}^{SP_j}(t) \tag{3.7}$$

$$R_{SC_i}(t) = ae^{-be^{-cA_{SC_i}(t)}} \tag{3.8}$$

where $Peers_i(t)$ denotes the set of SP_j's that have interacted with SC_i until time t and $|Peers_i(t)|$ denotes the cardinality of $Peers_i(t)$, i.e., the number of unique peers.

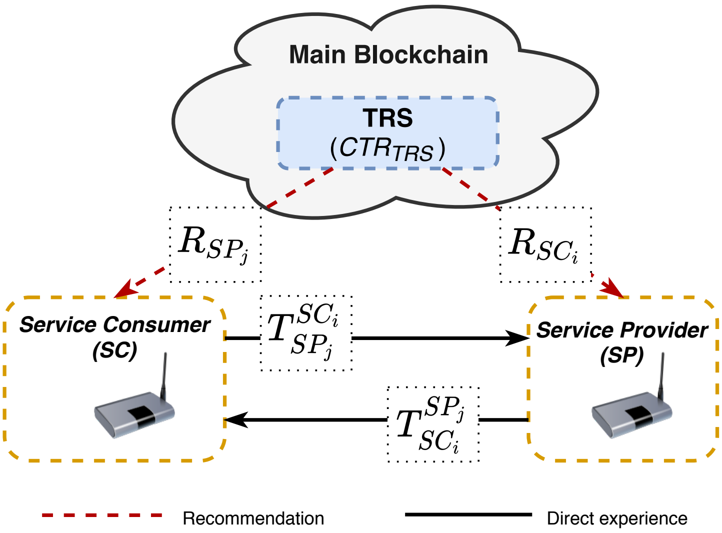

Figure 3.2: The trust relationship model.

Note that, $A_{SC_i}(t)$ is equal to $ln|Peers_i(t)|$ times the mean of $T_{SC_i}^{SP_j}(t)$ of all peers of SC_i. In this way, a larger number of $Peers_i(t)$ increases A_{SC_i} but in a tempered fashion, preventing an SC_i to achieve a large value of A_{SC_i} by having a large value of $T_{SC_i}^{SP_j}(t)$ but only by interacting with a small number of peers. In the Gompertz function (3.8), a, b, and c are the asymptote, the displacement parameter along x-axis, and the growth rate, respectively. This way, interactions with larger number of peers reinforce the reputation, but with a more appropriate sub-linear growth rate. Note that, the Gompertz function also guarantees that R_{SC_i} is bounded between 1 and 0, which corresponds to high and low reputation, respectively.

The reputation R_{SP_j} of SP_j is calculated in an identical fashion as follows:

$$A_{SP_j}(t) = \frac{\ln|Peers_j(t)|}{|Peers_j(t)|} \sum_{SC_i \in Peers_j(t)} T_{SP_j}^{SC_i}(t) \tag{3.9}$$

and

$$R_{SP_j}(t) = ae^{-be^{-cA_{SP_j}(t)}} \tag{3.10}$$

TRM is administered by CTR_{TRM} on MB by updating the trust and reputation scores on certain events. First, $T_{SC_i}^{SP_j}$ and R_{SC_i} are updated when SC_i requests an authorisation to access a resource of SP_j. $T_{SC_i}^{SP_j}$ and R_{SC_i} are also updated if an access control violation happens. Second, $T_{SP_j}^{SC_i}$ and R_{SP_j} are updated when SC_i

sends its feedback after receiving service from SP_j. We describe how the trust model works in action in Section 3.4.

3.4 Access Control Framework

This section presents the proposed adaptive access control framework, which employs TRM to manage authorisations and data access.

3.4.1 Access Control Primitives

We employ an ABAC scheme, which incorporates a novel TRM. In general, registration of attributes is mandatory for any SC prior to authorisation. An attribute α_i^m of SC_i is denoted as:

$$\alpha_i^m = \langle key, type, val \rangle \tag{3.11}$$

where key, $type$, and val correspond to the name, type, and value of α_i^m, respectively. Note that, it is common for an SC to have a set of attributes, denoted $A_i = \{\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^M\}$. An SC registers itself by sending an attribute registration request via a secure channel to an AA_k^y of pb_k, with which the SC is associated. AA_k^y should have some underlying evidence to validate and issue attributes, such as a smart building specification, and to prevent attributes forgery. Upon receiving a registration request, AA_k^y verifies the request and stores the attributes to pb_k by invoking a transaction:

$$TX_{reg} = \left[A_i | Sig_{SC_i} | timestamp | Sig_{AA_k^y}\right] \tag{3.12}$$

where Sig_{SC_i} and Sig_{AA_k} correspond to the signatures of SC and AA_k, respectively. As AAs are partially trusted, the other online AAs in pb_k need to validate TX_{reg} against the prescribed guidelines of attribute issuance, i.e., if the attributes actually match the underlying evidence. If TX_{reg} is valid, AA_k^x submits an endorsement message $E_i^x = \left\langle H(TX_{reg}), Sig_{AA_k^x} \right\rangle$ to pb_k, which contains TX_{reg} hash $H(TX_{reg})$ and AA_k^x signature $Sig_{AA_k^x}$. Note that to withstand faulty or untrusted AA, we require $3f + 1$ online AAs at any given time, where f is the number of faulty or untrusted AAs in pb. Consequently, TX_{reg} should obtain a minimum of $2f + 1$

endorsements to be considered valid [88]. Upon successful validation by other AAs, CTR_{AP_k} issues a sealing transaction TX_{seal} to MB as a proof that the attributes have been successfully registered:

$$TX_{seal} = [H(A_i)|E_i] \tag{3.13}$$

where $H(A_i)$ and E_i correspond to the hash of A_i and the collection of endorsements from other AAs, respectively.

To explicitly define the requirements to access resource r, the resource owner (i.e., SP_j) constructs an access policy $P_{r,c}$, which is declared as a Boolean rule of the required attributes. $P_{r,c}$ defines a set of actions, denoted $\tau \subseteq \{read, write, stream\}$, that an authorised SC may perform on resource r in context c. Note that SP_j has the authority to construct, update, and revoke access policies for all of its resources. $P_{r,c}$ is expressed as follows:

$$P_{r,c} = \left\langle A_p, c_p, \tau_p, U_r, \varphi_r, R_{SC}^{min}, T_{SC}^{min} \right\rangle \tag{3.14}$$

where $A_p = \{\alpha_p^1, \ldots, \alpha_p^n\}$ is a set of mandatory attributes, $c_p = \langle t, l \rangle$ is a set of allowed contexts (time and access throughput limit), U_r is the refresh rate of the data, φ_r is the fee (if any) to access the resource in a cryptocurrency unit, and R_{SC}^{min} and T_{SC}^{min} are the minimum reputation and trust scores to access resource r. Note that, half of the fee φ_r is returned back to the SC, if the SC submits an honest feedback, as described in Section 3.4.3.

A blockchain transaction TX_{pol} is executed to store access policy $P_{r,c}$ to the main blockchain MB and serves as the basis for the smart contract CTR_{pol} that manages authorisation of any SC. TX_{pol} is expressed as follows:

$$TX_{pol} = [P_{r,c}|timestamp|Sig_{SP}] \tag{3.15}$$

where $timestamp$ and Sig_{SP} correspond to the timestamp of policy creation and signature of SP, respectively.

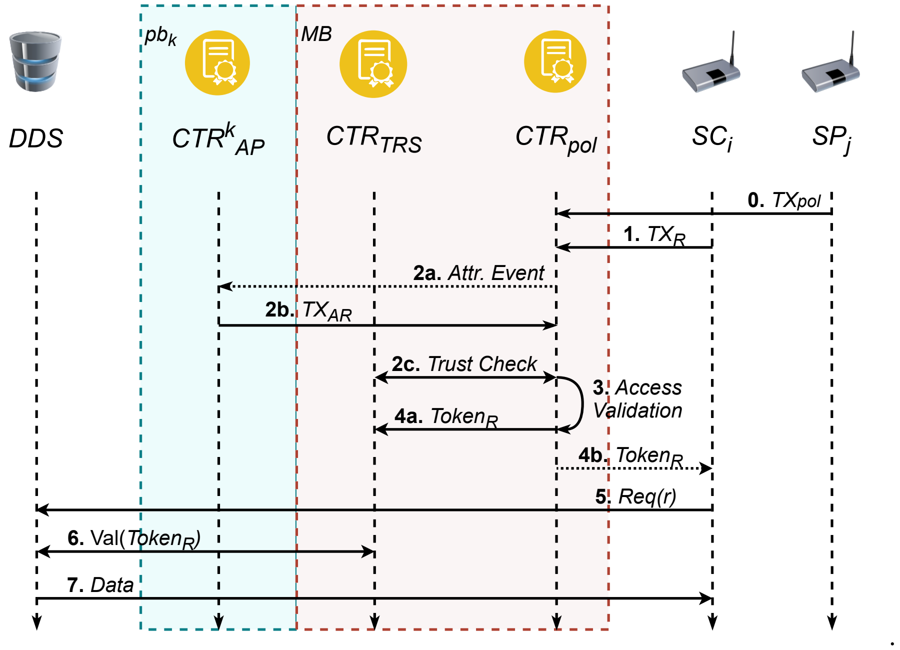

Figure 3.3: Authorisation process.

3.4.2 Authorisation Process

The proposed authorisation process is based on the ABAC scheme, in which access is given to users that satisfy certain attributes described in the access policy. Access control policy is enforced by CTR_{pol} in MB by evaluating an incoming authorisation request to access resource r on context c based on specific Boolean attribute rule-sets in the access policy $P_{r,c}$. Successful authorisation results in issuance of an access token by CTR_{pol}, which can be used by the SC to access the resource multiple times without repeating the authorisation process.

Initial system setup

We presume that there is a trusted smart city regulator that initially deploys the two core smart contracts (i.e., CTR_{TRM} and CTR_{pol}) to the main blockchain MB. In addition, independent AAs create and maintain their own pb, acting as sidechains, and announce their presence. Each AA_k deploys its own Attribute Provider Contract (CTR^k_{AP}) to pb_k and actively listens for incoming attribute lookup events from CTR_{pol} on MB.

Steps in authorisation

The authorisation process is shown in Figure 3.3. Prior to authorisation, SC_i must have its attributes registered to one of any available AAs. We assume that resource related information is already known in advance and stored in DDS.

- *Initialisation (Step 0)*: First, SP_j defines access policy $P_{r,c}$ and then initiates a transaction TX_{pol} to store the access control policy to MB.

- *Step 1*: SC_i authorises itself to CTR_{pol} by initiating transaction TX_R, which is defined as:

  $$TX_R = [r|\tau|Sig_{SC_i}] \tag{3.16}$$

  where r is the resource identifier, τ is the requested action, and Sig_{SC_i} is the signature of SC_i used for authentication.

- *Step 2*: Step 2a and 2b correspond to the bridging mechanism, which CTR_{pol} uses to validate SC_i's attributes. In Step 2a, CTR_{pol} emits an attribute validation event to the blockchain. Each pb connected to MB listens to the events and checks if PK_{SC_i} is registered on their chain. One of the CTR^k_{AP}'s will find PK_{SC_i} on pb_k. This CTR^k_{AP} validates the attributes and returns the result as an attribute response transaction TX_{AR} to CTR_{pol} (Step 2b). In addition, CTR_{pol} also calls CTR_{TRM} to obtain $T^{SP_j}_{SC_i}$ and R_{SC_i} (Step 2c).

- *Step 3*: CTR_{pol} executes Algorithm 3.1 to validate whether SC_i satisfies the required attributes and reputation threshold on $P_{r,c}$ and has sufficient balance to pay the access fee.

- *Step 4*: Upon successful access validation, CTR_{pol} issues $Token_R$, which is defined as:

  $$Token_R = \langle Exp_R, l, t\rangle \tag{3.17}$$

  where Exp_R is the token expiration time, l is the rate limit, and t is token timestamp. CTR_{pol} sends $Token_R$ to CTR_{TRM} to update the trust and reputation score of SC_i (Step 4a) and to SC_i as a proof that SC_i has been authorised to access resource r (Step 4b). In addition, CTR_{pol} sends half of the fee φ_r to SP_j's account and the other half of φ_r to CTR_{TRM} for feedback reward, as described in Section 3.4.3.

- Step 5: Based on the specification of resource r, SC_i locates the corresponding DDS to access the data. SC_i initiates the process by submitting a request to obtain a nonce from DDS. Subsequently, SC_i sends $Req(r)$ message to DDS over a secure channel that contains $Token_R$, the cryptographic nonce, and the signature of SC_i.

- Step 6: To prevent access token forgery, DDS validates $Token_R$ to CTR_{TRM}. DDS also checks if an SC sends quick successive requests at a rate higher than a certain rate limit l to prevent DoS attack. The use of forged token and a DoS attack would result in violation of access policy and would be reported to CTR_{TRM}, hence a drop in $T^{SP_j}_{SC_i}$. In addition, we assume that a dedicated intrusion detection mechanism exists to mitigate DoS attacks, e.g., blacklisting the node.

- Step 7: Upon successful validation, DDS responds to the access request by sending the data signed by the SP and an access timestamp signed by the DDS.

Asynchronous Authorisation

Unlike typical authorisation mechanisms which require both SP and SC to be active and connected to the system simultaneously [21], the nature of our proposed authorisation mechanism allows SP to be offline when SC is requesting for an access. In Step 0, SP_j initiates TX_{pol} to indicate that a resource is available to access with an access policy of $P_{r,c}$. Step 0 is actually an implicit delegation process, in which SP_j delegates the authorisation process to CTR_{pol}. As such, while an SC_i is requesting an access, SP_j may be offline temporarily, for instance, to save energy which is typical in IoT. We argue that this kind of asynchronous authorisation, in which all participating nodes are not required to be synchronously active at the same time, would offer greater flexibility for the SP.

Algorithm 3.1 Access Request Validation

```
Input: $P_{r,c}$, $TX_R$, $T_{SC_i}^{SP_j}$, $R_{SC_i}$, and $A_i$
Output: $Token_R$ or $\emptyset$
authorised ← 0
if $A_p \subset A_i$ and $\tau \subset \tau_p$ then
    if $T_{SC_i}^{SP_j} \geq T_{SC}^{min}$ and $R_{SC_i} \geq R_{SC}^{min}$ then
        if $getBalance(PK_{SC_i}) \geq \varphi_r$ then
            $authorised \leftarrow True$
        end if
    end if
end if
if authorised then return $Token_R \leftarrow \langle Exp_R, l, t \rangle$
elsereturn $\emptyset$
end if
```

3.4.3 Feedback Mechanism

While $T_{SC_i}^{SP_j}$ and R_{SC_i} are updated during the authorisation process and data access, the trust and reputation score of SP_j (i.e., $T_{SP_j}^{SC_i}$ and R_{SP_j}) are updated when SC_i submits a feedback TX_F to CTR_{TRM} after accessing resource r, which is defined as:

$$TX_F = \left[F_{SP_j,r}^{SC_i} | Data | H(Token_R) | Sig_{SC_i} \right] \tag{3.18}$$

where $F_{SP_j,r}^{SC_i}$ is the binary feedback from SC_i after accessing resource r (i.e., either positive or negative), $Data$ is the obtained data with last update and access timestamps signed by SP_j and DDS, $H(Token_R)$ is the hash of $Token_R$, and Sig_{SC_i} is the signature of SC_i used for authentication. Although SC may access r multiple times, TX_F can only be submitted once for each $Token_R$ and CTR_{TRM} will always check for duplicated feedback. To motivate SC_i to provide feedback, SC_i receives half of φ_r as a reward for submitting an honest feedback.

Recall that as defined in $P_{r,c}$, SP_j is expected to periodically update the data according to U_r. Positive $F_{SP_j,r}^{SC_i}$ refers to timely data, while the negative refers to obsolete data that has not been updated accordingly. CTR_{TRM} validates TX_F by comparing the last update and access timestamps against U_r and verifying the signature of SP_j and DDS as supporting evidence. The signature based evidence prevents the SC_i to forge the timestamps for malicious purposes. CTR_{TRM} inspects the evidence and increases the trust and reputation score of SP_j when the evidence supports the feedback. On the contrary, CTR_{TRM} will drop SC_i's trust and reputation score if SC_i submits misleading feedback. We show the procedures of our feedback mechanism in Algorithm 3.2. Note that, the validity of the data is beyond the scope of the present feedback mechanism.

Algorithm 3.2 Feedback Mechanism

```
Input: $P_{r,c}$, $TX_F$
Output: $True$ or $False$
$last\_update \leftarrow getUpdateTimestamp(Data)$
$access\_timestamp \leftarrow getAccessTimestamp(Data)$
$evidence \leftarrow checkSig(Data)$
$result \leftarrow False$
if $H(Token_R)$ exist then
    return $result$
end if
if $(access\_timestamp - last\_update) < U_r$ then
    if $F^{SC_i}_{SP_j,r} = positive$ and $evidence = True$ then
        $\varepsilon_t \leftarrow \varepsilon_{pos}$
        $result \leftarrow True$
    end if
else
    if $F^{SC_i}_{SP_j,r} = negative$ and $evidence = True$ then
        $\varepsilon_t \leftarrow \varepsilon_{neg}$
        $result \leftarrow True$
    end if
end if
if $result = True$ then
    $reCalculate\ T^{SC_i}_{SP_j}, A_{SP_j}, R_{SP_j}$
    $sendCryptoTo(SC_i)$
else
    $\delta_t \leftarrow \delta_{neg}$
    $reCalculate\ T^{SP_j}_{SC_i}, A_{SC_i}, R_{SC_i}$
    $sendCryptoTo(SP_j)$
end if
return $result$
```

3.5 Performance Evaluation

In this section, we present the performance evaluation of our solution based on our proof-of-concept implementation, that was tested on a public Ethereum test network. We evaluate our solution on reputation specific metrics, namely trust and reputation evolution and blockchain performance metrics such as, authorisation latency and gas used.

3.5.1 Proof-of-Concept Details

We selected Ethereum as the blockchain platform for the proof-of-concept implementation, primarily due to three reasons. First, Ethereum is suitable for both permissionless and permissioned environment, which is necessary for our proposed system. In the Ethereum permissionless environment, each peer may freely join and leave the network at any given time and there is no authority that administers user registration. Note that, each peer must use the elliptical curve SECP-256k1 [53] key generation approach used in Ethereum to obtain valid credentials. Second, Ethereum supports smart contracts, written in Solidity, which is a Turing complete programming language [89]. The smart contract is executed within a decentralised Ethereum Virtual Machine (EVM), which acts as a trusted platform for decentralised computation [53]. Third, Ethereum offers an in-built cryptocurrency for token transactions, with Ether being the default fundamental token for transactions. We tested our smart contracts on the Rinkeby public Ethereum test-network, which implements Proof-of-Authority as the consensus mechanism. Note that, our proposed solution can also be implemented on any blockchain platform that supports smart contract execution, for instance, Hyperledger Fabric [55].

To implement our proposed framework, we built a lab-scale testbed which consists of six Raspberry Pi 3 B+ (1GB RAM, 1.4GHz 64-bit quad-core ARM CPU, Raspbian 10 buster) as IoT nodes and a single MacBook Pro 2019 (8GB RAM, 1.4 GHz quad-core Intel Core i5 CPU, macOS 10.14.6) acting as both AA and DDS. MacBook Pro computer and all Raspberry Pis ran `geth` v1.9.12 as an Ethereum client to connect to the Rinkeby test-network. We used Python v3.7.7 and bash scripts to simulate interactions among nodes with `web3py` v5.11 and `py-solc` v3.2.0 library for communicating with Ethereum peers and compiling the smart contracts, respectively. We wrote our implementation of CTR_{pol}, CTR_{TRM}, and CTR_{AA} in Solidity v0.6.6 [89], with native support for efficient computation and verification of hashes [90]. A summary of the evaluation details is presented in Table 3.1.

It is important to note that, as of Solidity v0.6.6, there is no support for floating

Table 3.1: Evaluation details.

Parameters	***Values***
δ_{pos}	1
δ_{neg}	-3
γ	$[0.0, 0.6, 0.7, 0.8, 0.9, 1.0]$
$\|Peers_i(t)\|$	$[2, 5, 10, 20]$
a	1
b	-4
c	2
Iteration	30

point computation [89], which requires minor changes in the implementation of the computations. Recall that it is impossible to remove a smart contract once it has been deployed to the blockchain, as the blockchain is immutable. In practice, upgrading a smart contract involves marking the old smart contract as obsolete and deploying another replacement smart contract. To facilitate secure smart contract updates, we separate the implementation of access control logic and data storage into main and secondary smart contracts. The main smart contracts, which contain all functions for the access control and TRM logic, do not store data and connect to the secondary contracts to collect and store data. This way, when a `selfdestruct` method is called and a new smart contract is deployed, the old smart contract becomes unusable but the old data remains accessible to the new smart contract. This is important when a bug is discovered and an update is necessary.

3.5.2 Evaluation Results

Trust and Reputation Convergence

We study the convergence of our proposed TRM by varying the weighting parameters and present the results in Figure 3.4, 3.5 and 3.6. By varying the ageing parameters (i.e., γ and μ), we basically allocate different weights for recent and older interactions in calculating the trust score. Figure 3.4 shows the convergence of $T_{SC_i}^{SP_j}$ with different γ for all positive interactions with $\delta_{pos} = 1$ and $\delta_{neg} = -3$. We can observe that higher values of γ deliver more gradual growth in trust score evolution, which converges to 1, i.e., the maximum value. However, for the two extreme values of

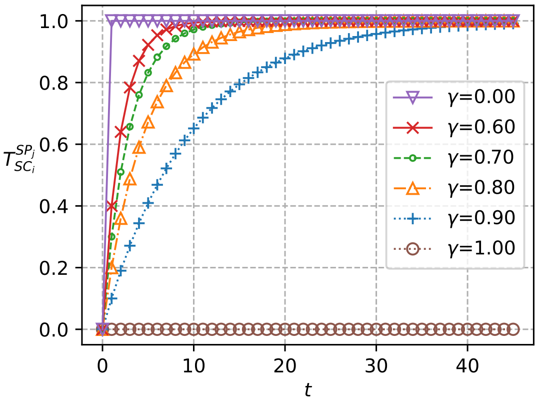

Figure 3.4: Convergence of trust score $T_{SC_i}^{SP_j}$ with different γ ($\delta_{pos} = 1$ and $\delta_{neg} = -3$)

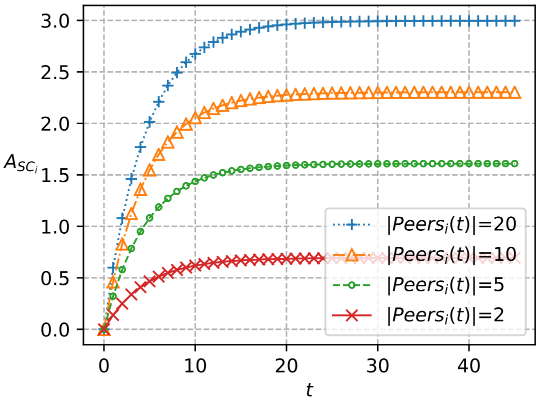

Figure 3.5: Aggregation of trust score A_{SC_i} with different $|Peers_i(t)|$

$\gamma = 0$ and $\gamma = 1$, the trust scores remain static at 1 and 0, respectively, regardless of the interactions. Recall that, in Eq. (3.4), the value of $max(T_{SC_i}^{SP_j})$ is capped at δ_{pos}. Recall in Section 3.3 that the reputation score of a node (i.e., R_{SC} or R_{SP}) is affected by how many unique peers the node has interacted with. Intuitively, for positive interactions, having more peers would increase the reputation score. As shown in Fig 3.5, A_{SC_i} is directly proportional to $\ln|Peer_i(t)|$. In Fig 3.6, we can see that although the reputation score R_{SC_i} is positively correlated with $|Peer_i(t)|$, the reputation score R_{SC_i} is capped at 1, which corresponds to the variable a in Eq. (3.8) and (3.10).

Authorisation and access latency

We compare the authorisation latency, i.e., the time taken from Step 1 to Step 7 in Fig 3.3, for two different connection methods. First, an SC connects to MB to authorise itself by running a light geth node. Second, an SC connects to the blockchain via Infura[1], a third-party API provider. Note that the fundamental difference in both connection schemes is about main blockchain synchronisation, which is required in geth but not in Infura. When an IoT node runs a geth light node, the IoT node itself is responsible for updating and keeping a local copy of the blockchain, which may result in a small increase in CPU utilisation and require some disk storage. In our experience, running a geth light client on the Rinkeby

[1] https://infura.io/

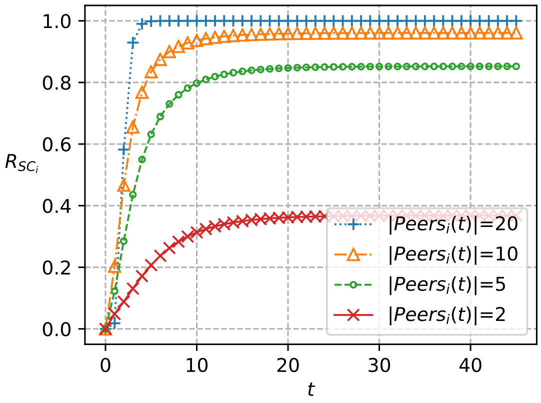

Figure 3.6: Reputation score R_{SC_i} with different $|Peers_i(t)|$ ($a = 1$, $b = -4$, and $c = -2$)

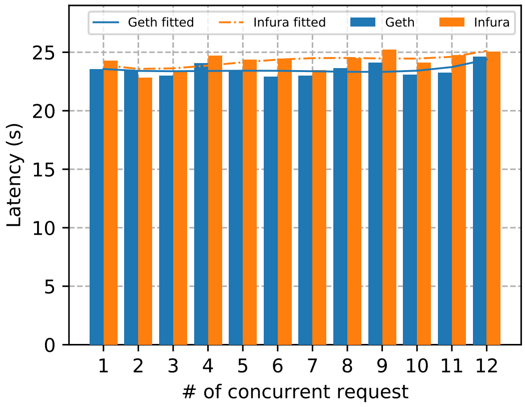

Figure 3.7: Comparison of authorisation latency via Geth light client and Infura API Gateway (Step 1-7).

test-network (as of July 2020) occupies approximately 500MB of disk space with insignificant CPU usage. On the other hand, if the IoT node opts to connect to the blockchain via Infura, then it is not required to keep synchronising a local copy of the blockchain. Since, the IoT nodes sign the transactions locally to ensure the security of their private keys, it implies that Infura provider does not have access to the private keys of the nodes. We repeated the experiment 30 times with different number of concurrent requests and plot the results in Figure 3.7. In general, authorisation via geth achieves slightly lower latencies than via Infura. We can observe that increasing number of concurrent requests results in no significant increase of the authorisation latency. As there is no significant difference in the authorisation latency between the two methods and Infura does not require IoT nodes to keep a local copy of the blockchain, authorisation via Infura seems to be the preferred method. However, it is important to note that the IoT node must assume that Infura is trusted and does not perform any malicious actions. We also examine the access latency (i.e., Step 5-7), in which the SC re-uses the $Token_R$ to access r before the token expires. We measured the latency for different number of concurrent requests and repeated the experiments 30 times. As shown in Figure 3.8, the access latency is three orders of magnitude lower than the authorisation latencies, as the SC is not required to repeat Step 1-4. However, there is an increasing trend when the number of concurrent requests is increased.

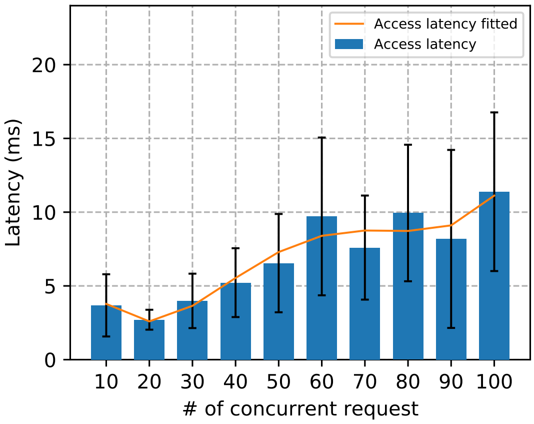

Figure 3.8: Comparison of access latency, in which SC has already obtained $Token_R$ beforehand (Step 5-7).

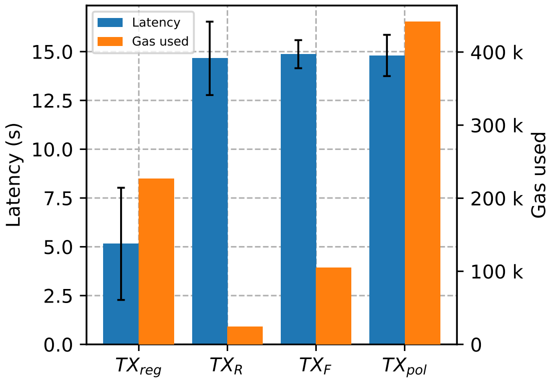

Figure 3.9: Latency and required gas of attribute registration, authorisation, feedback, and policy registration.

Latency and required gas

In Ethereum, there is a fee to execute transactions that alter the blockchain state. The fee, which is referred to as Gas, depends on the number of EVM opcodes involved during the execution of a particular function [53]. The gas also helps to avoid excessive execution of smart contracts, e.g., infinite loops. In practice, the gas is relatively small and sometimes negligible.

We investigate the amount of gas required to execute essential functions in our smart contract design, namely attribute registration (TX_{reg}), authorisation (TX_R), feedback (TX_F), and policy registration (TX_{pol}). In addition, we investigate the latency to execute these functions. We repeated the experiments 30 times and plot the results in Figure 3.9. The transaction latencies are similar for TX_R, TX_F, and TX_{pol}, as these transactions are executed in MB. The transaction latency for TX_{reg} is relatively lower as it is executed in *pb*, which has faster block generation time. TX_R requires the least Gas, while TX_{pol} requires the most, due to the different amount of data contained within the TX_R and TX_{pol}, which in turn results in a different number of EVM opcodes in the execution (refer to Eq. (3.14) (3.15) (3.16)).

Trust evolution of honest and malicious node

To study how the convergence of trust scores, we simulate three SPs with different behaviour. One of the SPs is honest for all t, another is malicious for all t, while

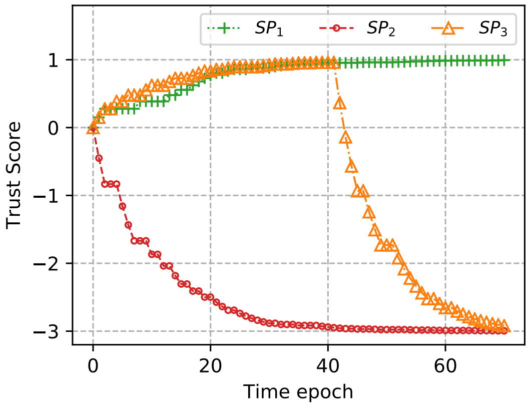

Figure 3.10: The trust evolution comparison of honest and malicious SP (μ = 0.8, $\varepsilon_{pos} = 1$, and $\varepsilon_{neg} = -3$).

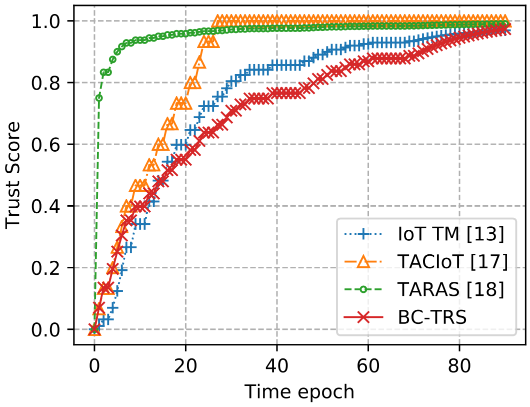

Figure 3.11: The comparison of trust score convergence between several proposals for positive interactions.

the third is an SP that initially acts honestly until $t = 40$, then becomes malicious for the rest of the simulation. We use $\mu = 0.8$, $\varepsilon_{pos} = 1$, and $\varepsilon_{neg} = -3$ as the parameters and plot the results in Fig 3.10. We can see that all SP scores start at 0 as the initial trust value. The trust scores for honest and malicious SPs reach the maximum and minimum boundaries approximately at similar time $t = 40$. Note that the maximum and minimum trust scores follow equation (3.4). At $t > 40$, the trust score of the third SP drops significantly. As the interactions continue consistently, the trust scores remain unchanged.

3.5.3 Comparative Analysis

As a quantitative comparison, we examine the trust evolution of our proposed solution against other proposals in trust management for IoT authorisation [64]–[66]. We simulated two nodes (i.e., SC) which have different behaviour and calculated the trust score of each node using different trust models. For each trust model, we set the parameters as recommended in the corresponding paper, i.e., $a = 1$, $b = 6$, $c = 0.1$, $\gamma = 0.95$, $\delta_{pos} = 3$, and $\delta_{neg} = -5$ for [66]; $S = 7$, $c_j^p = 0.5$, and $N = 15$ for [64]; and $W = 1$ for [65]. However, the systems' parameters could still be adjusted from their default values depending on the implementation scenario. We plot the trust score evolution of benign and malicious interactions in Figure 3.11 and Figure 3.12, respectively. For honest SCs, we see that although all trust models converge to similar upper boundary, each trust model has different convergence rate. TARAS

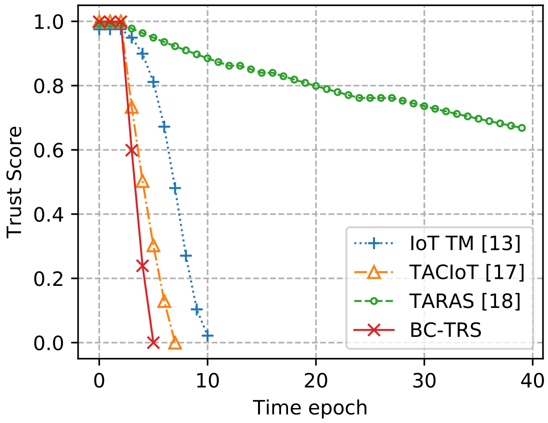

Figure 3.12: The comparison of trust score convergence between several proposals for negative interactions.

seems to have the quickest convergence rate at which it is able to reach 0.8 at less than 10 time epoch, while BC-TRM reaches 0.8 at time epoch around 50. Note that a faster convergence rate is generally less preferred as it is more vulnerable to newcomer attacks. On the other hand, a fast decline is preferred to penalise malicious nodes. Figure 3.12 shows that three models demonstrate a significant abrupt decline for malicious SCs, in which the trust score drops to zero within 10 time epoch. We note that BC-TRM reaches zero only within 3 malicious interactions.

In Table 3.2, we compare our proposed solution against other related trust management models for IoT authorisation. We see that in terms of trust computation, the complexity of other models depends on the following parameters: i) the number of previous interaction p, ii) the number of trust dimension j incorporated in the trust calculation, and iii) the number of I-sharing group member s_i. Recall in Section 3.3 that our trust score can be calculated using simple recursion, which reduces our trust computation complexity to $O(1)$. In TACIoT and TARAS, the IoT devices rely on a centralised trust manager to store interaction history, which reduces the storage requirement to $O(1)$. IoT TM, however, requires each SP to store the interaction history with any SC in its own storage, which amounts to the number of service consumer N and the number of previous interaction p_x for each SC. Our proposed solution offloads the storage to blockchain, which reduces the storage complexity to $O(1)$. Our proposed solution also offers privacy preservation by storing sensitive information on private sidechains. Our solution supports asynchronous au-

Table 3.2: Comparing the complexity of trust managements for IoT authorisation.

Proposal	**Complexity**	
	Trust Computation	*Storage*
TACIoT [64]	$O(p * j)$	$O(1)$
TARAS [65]	$O(s_i)$	$O(1)$
IoT TM [66]	$O(p)$	$O(\sum_{x=1}^{N} p_x)$
BC-TRM	$O(1)$	$O(1)$

thorisation, in which both SP and SC are not required to be simultaneously online to perform authorisation. Lastly, our model also supports bidirectional trust assessment, wherein each SP can assess the trustworthiness of SC and vice versa, as seen in Figure 3.2.

3.6 Discussion

3.6.1 Reliability and Scalability

We design our proposed solution to be resilient against attacks discussed in Section 3.2.2. When an adversary launches bad mouthing attacks, the request signatures confirm the authenticity of the sender, which prohibits the adversary from stealing or forging invalid tokens of other nodes for maliciously reducing its reputation score. Our framework is also resilient to Sybil and newcomer attacks, which are handled by the attribute registration mechanism. To prevent Sybil attacks, participants are prevented from self-registering to AA more than once, as the AA keeps track of the attributes relation to the technical specifications and ownership information of the device. In addition, our trust computation model impedes illegitimate attempts to increase reputation scores, i.e., performing ballot-stuffing and self-promoting attacks, as a larger number of adversaries would have to collude in order to make the attack effective (see Section 3.3).

In our solution, blockchains serve as the backbone of the network. Subsequently, the scalability of our system is inherited from the underlying blockchain instantiation. Note that our solution does not address the issue of blockchain scalability, but there have been active contributions from the community about possible solutions to limited scalability [91]. Figure 3.7 indicates that our solution can achieve a stable latency for different number of concurrent requests. However, a slightly increasing trend is observed in Figure 3.8, which indicates that DDS might introduce a bottleneck in the network and hinder scalability. Note that this trend is observed as we implemented DDS on a single node, in fact a possible solution to remove the bottleneck is to add redundancy in DDS via a scalable data storage [92].

As shown in Fig. 3, our authorisation process involves three main stages, e.g., attribute validation, trust computation, and access validation, which has different computational complexity in each stage. First, in attribute validation, CTR^k_{AP} may perform binary search to find the attributes of an SC, which results in a complexity of $O(\log n)$, where n is the number of registered SCs. Second, our trust model involves a simple recursion in trust computation, which only requires CTR^k_{TRM} to either increment or decrement the previous trust value (complexity of $O(1)$). Third, access validation involves iteratively comparing the required attributes in $P_{r,c}$ against SC's attributes, which has a computational complexity of $O(n)$. In general, the overall complexity of the authorisation process is linear ($O(n)$).

3.6.2 Implications of TRM for Theory and Practice

It is known that access control demands highly sensitive consideration that requires certainty of who can access what resource. However, trust-based approaches are generally probabilistic in nature, which may make them vulnerable to exploitation. Relying entirely on a trust score for access control is risky, as attackers could conceivably build up trust to gain access. To reconcile these issues, we based our trust-based approach on an established attribute based access control scheme with trust and reputation score as auxiliary attributes. We argue that incorporating explicit trust scores would help to achieve a dynamic and flexible access control system while also prohibiting access to malicious and compromised nodes.

The evaluation results indicate that the authorisation process incurs appreciable latency. However, the latency of re-accessing a resource with a previously obtained token is relatively low. Note that, in general SCs may only need to authorise themselves once $Token_R$ has expired (defined in Exp_R). One possible application of our proposed model is in managing access to critical infrastructure, such as power grid or water supply network, that require stringent measures of authorisation, limited to highly trusted actors with explicit inherent attributes. For instance, we can implement our solution to quantitatively oversee the performance of smart building contractors. Ideally, smart building contractors that regularly maintain critical infrastructure should be highly trusted, demonstrated by an exceptional track record of prior interactions. The contractors, after deploying IoT devices on the infrastructure premises, would either act as a SP or SC depending on the specifications. The contractors regularly monitor the infrastructure condition by frequently checking IoT sensor readings and reporting certain anomalies to corresponding authorities. Violations in the maintenance procedures, i.e., malicious actions, would result in rigorous penalty as per our proposed TRM.

3.7 Chapter Summary

In this chapter, we proposed a trust-based access control framework for decentralised IoT network. We design an auxiliary TRM as part of a blockchain-based ABAC mechanism that incorporates trust and reputation scores as additional attributes for achieving dynamic and trustworthy access control mechanism. We design our framework to be blockchain-agnostic, which can be implemented in any blockchain platforms that have adequate support for smart contract execution. We implemented a proof-of-concept in a public Rinkeby Ethereum test-network interconnected with a lab-scale testbed. The experimental results show that our proposed framework achieves consistent processing latencies. In addition, the comparison against related work in trust-based authorisation reveals that our trust model exhibits more gradual growth and more rapid decrease, which are preferred properties in TRM to avoid trust-based attacks. In conclusion, our framework is feasible for implementing an effective access control in decentralised IoT networks.

Chapter 4

Blockchain-based TRM for Collaborative IDS

This chapter addresses the issues in the trustworthiness of the collaboration between nodes in the blockchain-based Collaborative Intrusion Detection Systems (CIDS), as outlined in Section 1.1.2 and 2.5.2. In CIDS, network nodes share their expertise and experience, for instance, detection rules, to build collective knowledge of the recent attacks and increase the detection accuracy. Conventional CIDS assumes that the nodes are always honest. However, a trusted node may later be compromised and share untruthful detection rules to contaminate the detection database, which would potentially expose the network to attacks [72]. In this chapter, a decentralised CIDS framework is proposed with an emphasis on building trust between CIDS nodes. Unlike the TRM framework in Chapter 3 which derives trust based on the nodes' adherence to the access control policies, i.e., behaviour-based trust, the proposed TRM for CIDS utilises data-based trust. Here, the TRM framework quantifies the trustworthiness of CIDS nodes from the quality of their contributed detection rules. However, both TRM frameworks in Chapter 3 and 4 employ smart contracts to provide transparent and verifiable trust computation. In addition, the framework utilises a decentralised storage to host the shared trustworthy detection rules to ensuring scalability. The proof-of-concept implementation in a lab-scale testbed shows that the framework is feasible and performs within the expected benchmarks of the Ethereum platform.

4.1 Introduction

Intrusion Detection Systems (IDS) have been the industry standard for securing IoT networks against known attacks. In recent years, an expansion of the attack surface has been inevitable, partially due to the adoption of IoT devices in diverse areas. This has consequently escalated the importance of defending the network from emerging threats [23]. Unfortunately, conventional IDS that work in isolation may be easily compromised, since they are unaware of new attacks which are not in their detection database. To increase the capability of an IDS, researchers proposed the concept of blockchain-based CIDS, wherein blockchain acts as a decentralised platform allowing collaboration between CIDS nodes to share intrusion related information, such as intrusion alarms and detection rules [24]. However, proposals in blockchain-based CIDS overlook the importance of continuous evaluation of the trustworthiness of each node and generally work based on the assumption that the nodes are always honest.

To address the trust issues in CIDS, a TRM framework for building trustworthy CIDS is proposed to continuously evaluate the trustworthiness of the CIDS nodes by evaluating the quality of the detection rules contributed by each IDS node. Each participating CIDS node can update its knowledge with the trustworthy detection rules to detect new attacks. The proposed framework utilises a peer-to-peer decentralised storage to maintain a copy of the shared trustworthy detection rules, thus ensuring scalability. The framework divides the participating nodes into three categories, namely validator, contributor and regular nodes, each of which has a different role in the system. In the proposed framework two smart contracts are designed, namely Trust and Reputation Management (TRM) and Storage smart contract, to quantify each node's trustworthiness and manage the decentralised storage, respectively. The framework offloads trust computation to the TRM smart contract which reduces the computation load for each CIDS node. A smart contract-based voting mechanism is proposed to achieve collaborative detection rule validation and avoid an adversary from contributing deceptive detection rules. The framework is designed as a blockchain-agnostic platform, which can be implemented on any blockchain instantiation that supports smart contracts. While the chapter uses signature-based CIDS as an illustrative example, the TRM concept can be generalised to other types of CIDS.

4.1.1 Chapter Contributions

In summary, this chapter makes the following contributions:

- This chapter proposes a trustworthy CIDS framework that continuously evaluates contributions from each CIDS node to protect the network from invalid detection rules. The framework presents transparent and accountable trust mechanisms that provide auditability.
- The proposed TRM framework for CIDS offloads the trust computation and trustworthy detection rules to the blockchain and the decentralised storage, thus reducing the load on each CIDS node.
- The proposed TRM framework utilises an efficient and tailored trust model with relatively better scalability than the classical challenge-based methods in the existing work. While challenge-based scheme may work well for medium sized networks, challenge-based techniques would be impractical and raise scalability issues when the number of CIDS nodes are relatively large.
- The trust mechanism separates the trust score for each contribution (rules) and overall trustworthiness of the CIDS node. As such, each CIDS node can conveniently infer the quality of the rules by looking at both scores.
- A proof-of-concept implementation of the proposed framework is developed in a private Ethereum network hosted on a lab-scale testbed. The framework is evaluated in terms of the evolution of trust scores, smart contract latency and Ethereum gas consumption. The experimental results show that the solution is feasible and performs within the expected benchmarks of the Ethereum platform.

4.1.2 Chapter Organisation

The remainder of the chapter is organised as follows. Section 4.2 presents the proposed framework with its underlying assumptions. Section 4.3 outlines the trust and reputation system, while the CIDS framework is discussed in Section 4.4. The performance evaluation of our solution is presented in Section 4.5, while the conclusion of this chapter is presented in Section 4.6.

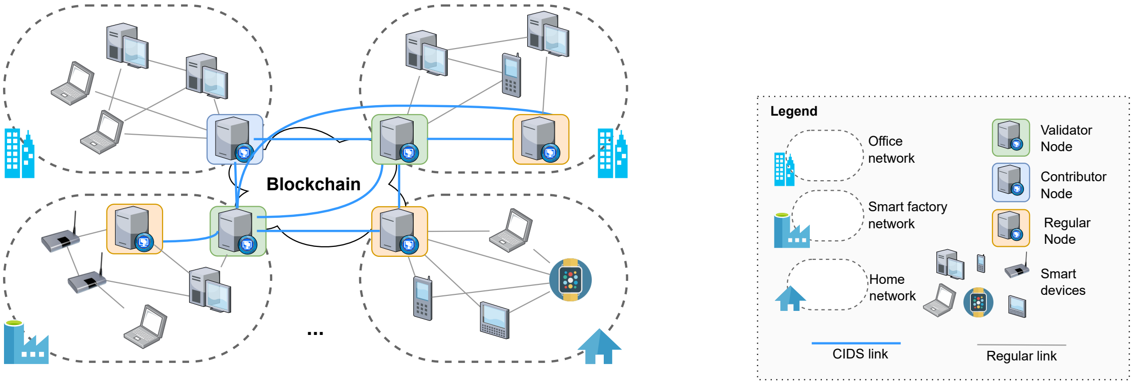

Figure 4.1: An overview of the proposed decentralised CIDS. Here, blockchain is an instrumental part, using which all CIDS components, e.g., validator, contributor and regular nodes, communicate to collaboratively build a global trusted rules database.

4.2 Proposed Architecture and Assumptions

In this section, we describe our decentralised CIDS architecture by elaborating the fundamental components and outlining the threat model. Lastly, we explain the assumptions in the present study.

4.2.1 Architectural Overview

We present the overview of our proposed system in Figure 4.1. We design our architecture to span across multiple organisational networks, each of which typically comprises multiple smart devices with varying computational and storage capacity. In each organisation, we require at least two CIDS nodes which act as signature-based IDS nodes that monitor the network for any attack occurrence. CIDS nodes communicate with other CIDS nodes through the TCP/IP protocol suite with an industry-standard encryption mechanism to ensure security, e.g., Transport Layer Security (TLS).

We model our system as $\mathbb{C} = (C, \mathbb{B}, \mathbb{I})$, where C is a set of collaborating CIDS nodes that share intrusion related information and $\mathbb{B}$ is the blockchain and $\mathbb{I}$ is the decentralised storage network. In our model, each CIDS node is connected to both blockchain $\mathbb{B}$ and decentralised storage network $\mathbb{I}$ for collaboration and storing shared detection rules, which contain a pattern of malicious network attacks, such as file hashes, malicious domains or particular byte sequences. The blockchain network

Table 4.1: Important notations and their description.

Notations	***Description***
C, $\mathbb{B}$, $\mathbb{I}$	CIDS, blockchain and dec. storage network
cv, cc, cr	a validator, contributor and regular node
k_p and k_s	the public and secret key
$r_{a,j}$	an IDS rule from cc_a
$S^e_{a,j}$	validity score of $r_{a,j}$ from cv_e
$t_{a,j}$ and T^m_a	the trust score of $r_{a,j}$ and cc_a
R_{db} and R_{loc}	the global and local rules database
SC_{trm} and SC_{str}	the TRM and storage smart contracts
$E^v_{r_{a,j}}$	new rule blockchain event
$E^o_{r_{a,j}}$	new validated rule blockchain event
φ^e_j	validation result from cv_e
$M_{r_{a,j}}$	description of $r_{a,j}$ according to IDMEF [93]
Z_{cc_a}	a zip archive of $M_{r_{a,j}}$ and $r_{a,j}$
D_c	decision rule function

$\mathbb{B}$ is the main hub for collaboration, through which each CIDS node exchanges information via blockchain transactions. Each CIDS node collaboratively builds a global trusted rules database R_{db} that contains a collection of detection rules contributed by each CIDS node. Each CIDS node also keeps a local copy of an IDS database $R_{loc} \in R_{db}$. We summarise important notations used in this chapter in Table 4.1.

CIDS network

All CIDS nodes form the CIDS network, which is connected to the other components of the system, e.g., blockchain network $\mathbb{B}$ and decentralised storage network $\mathbb{I}$. We assume that all CIDS nodes are equipped with sufficient resources to run a blockchain client. Each CIDS node holds a corresponding blockchain public-private key pair $\{k_p, k_s\}$ and is identifiable by the public key k_p. We define three types of IDS nodes, denoted $C = (C_V, C_C, C_R)$, where C_V, C_C and C_R correspond to validator, contributor and regular nodes, respectively.

Validator nodes: Validator nodes, denoted $C_V = \{cv_1, cv_2, \ldots, cv_n\}$, are in charge of maintaining the CIDS network, including registration of new CIDS nodes and validation of contributed detection rules from other nodes by means of off-chain processes [25]. Validator nodes validate detection rule $r_{a,j}$ submitted by other CIDS nodes via a consensus algorithm, which assigns a trust score to each rule. Validator nodes then append the validated rules to the trusted rules database R_{db}.

Contributor nodes: We define a contributor node $\{cc_a | \forall cc \in C_C\}$ as a CIDS node that contributes a detection rule $r_{a,j}$ to the CIDS network $\mathbb{C}$. Each cc_a has a trust score T_a derived from the trustworthiness of its contributions.

Regular nodes: We refer to the rest of the CIDS nodes in C as regular nodes, denoted $C_R = \{cr_1, cr_2, \ldots, cr_k\}$. Regular nodes are only interested in using validated IDS rules in R_{db} to update their local rules database R_{loc} by subscribing to the system for a notification of newly validated $r_{a,j}$ in R_{db}. A cr can conveniently examine the trust scores of both rule $r_{a,j}$ and node cc to decide which rules to be included into their R_{loc} based on a locally determined threshold. A regular node is passive and consequently does not get assigned a trust score.

Blockchain network

We design our architecture to be blockchain-agnostic, which supports any commodity blockchain platform with a prerequisite of supporting Turing-complete smart contract execution [89]. In this work, we consider a single permissioned blockchain $\mathbb{B}$, wherein access is limited to certain known parties, which helps to build the first layer of defence. To avoid malicious actions, registration of new CIDS node is handled by C_V. In general, blockchain $\mathbb{B}$ is used to track the contributions of each cc and to store the metadata of the contributed detection rules. Note that, we do not store the IDS rules on-chain but in the decentralised storage layer.

We deploy two smart contracts onto blockchain $\mathbb{B}$. First, a Trust and Reputation Management (TRM) smart contract SC_{trm} manages the contributions of all cc by quantifying them via a transparent and verifiable TRM mechanism (see Section 4.3). Second, a storage smart contract SC_{str} maintains the hash and metadata of each $r_{a,j}$ to provide a connection between the blockchain and the decentralised storage network.

Decentralised storage network

We employ IPFS, the InterPlanetary File System, as the decentralised storage network [94]. We separate the data storage to prevent the blockchain size from becoming too large and thus ensuring scalability. The decentralised storage network is managed by C_V, which limits the access only to subscribed cr.

4.2.2 Threat Model and Assumptions

In our architecture, we assume that the adversaries are able to launch a poisoning attack aimed to inject misleading detection rules to the trusted database R_{db}. The adversaries can also compromise a maximum of ℓ validator nodes. However, we require that there are n available validator nodes such that $n = 3\ell + 1$ to tolerate ℓ faulty validator node(s), i.e., 1/3 fault tolerance as in PFBT [88]. The adversaries are also capable of performing trust and reputation attacks as follows:

- *Self-promoting attacks:* A malicious actor may try to increase its own reputation score by submitting and validating a detection rule by itself.
- *Bad-mouthing attacks:* A malicious node may attempt to ruin the reputation of another node providing negative validation results regardless of the quality of the contributed model.
- *Ballot-stuffing attacks:* A node may collude with other nodes to deliberately increase their reputation scores, for instance by submitting the same detection rules multiple times.
- *Whitewashing or newcomer attacks:* A node attempts to rejoin the network using a new identity aiming to reset its previously recorded bad behaviour and obtain a fresh reputation score.

We assume that our system inherits the assumptions of a commodity blockchain platform, which include security against peer-to-peer and consensus attacks, such as Sybil, eclipse and 51% attacks [86]. We assume that each validator holds a sufficient local detection rules database to validate the submitted detection rules and detect whether a submitted rule is malicious. All CIDS nodes in our architecture, regardless of their type, are bound to cryptographic primitives, which prevents manipulation and duplication of blockchain identities, i.e., public and private key pairs.

4.3 Trust and Reputation Management

We design the Trust and Reputation Management (TRM) to evaluate the trustworthiness of each contributor node *cc* in the network $\mathbb{C}$ and also to protect the network from any corrupted or malicious detection rules. Each *cc* will obtain a trust score after submitting a detection rule from which the score is calculated. In general, the score is increased when *cc* contributes valid detection rules and decreased when submitting compromised rules. Intuitively, a high trust score indicates a trustworthy contributor node, while a low score indicates otherwise.

Figure 4.2 shows the trust relationship model in our proposed TRM. Suppose a contributor node $cc_a \in C_C$ submits a detection rule $r_{a,j}$. A validator node cv_e validates $r_{a,j}$ and based on its accuracy, gives a score $S^e_{a,j} \in [0.5, 1]$ to indicate valid rules, or $S^e_{a,j} \in [0, 0.5)$ to indicate invalid rules. As such, $S^e_{a,j}$ functions as a vote from cv_e and consequently we have n scores $\{S^1_{a,j}, S^2_{a,j}, \ldots, S^n_{a,j}\}$ for each $r_{a,j}$ as there are n available validator nodes. We use the following formula to calculate the aggregated trust value of $r_{a,j}$, denoted by $t_{a,j}$:

$$t_{a,j} = \frac{1}{n} \sum_{e=1}^{n} S^e_{a,j} \delta_e \tag{4.1}$$

where

$$\delta_e = \begin{cases} \delta_{val}, & \text{if } S^e_{a,j} \geq 0.5 \\ \delta_{inv}, & \text{otherwise,} \end{cases}$$

where δ_{val} and δ_{inv} are non-negative weights associated with a valid and invalid rule, which can be determined via a heuristic method. To make it more difficult to build trust than to lose it, we set $\delta_{val} < \delta_{inv} \leq 1$ so that malicious contributions are assigned a higher weight. We propose a mechanism to collaboratively determine valid rules based on the scores from each validator node, which is elaborated in Section 4.4. A valid $r_{a,j}$ is appended to R_{db}, while invalid $r_{a,j}$ would not be included.

Without loss of generality, let us assume that the first q votes are valid and the rest are invalid, then we get

$$t_{a,j} = \frac{1}{n} \left(\delta_{val} \sum_{e=1}^{q} S^e_{a,j} + \delta_{inv} \sum_{e=q+1}^{n} S^e_{a,j} \right) \tag{4.2}$$

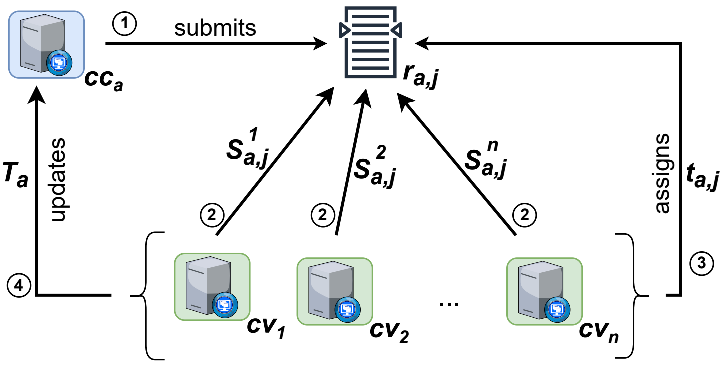

Figure 4.2: The trust relationship model.

Since $S^e_{a,j} \in [0.5, 1]$ and $S^e_{a,j} \in [0, 0.5)$ for valid and invalid rules respectively, we get the following bounds

$$\frac{q\delta_{val}}{2} \le \delta_{val} \sum_{e=1}^{q} S^e_{a,j} \le q\delta_{val} \tag{4.3}$$

and

$$0 \le \delta_{inv} \sum_{e=q+1}^{n} S^e_{a,j} < \frac{\delta_{inv}(n-q)}{2} \tag{4.4}$$

Then from (4.2), (4.3) and (4.4) we get the lower and upper bounds for $t_{a,j}$ as

$$\begin{aligned} &0 \le t_{a,j} \le \delta_{val}, && \text{for } \ \delta_{inv} < 2\delta_{val} \\ &0 \le t_{a,j} < \delta_{inv}/2, && \text{for } \ \delta_{inv} \ge 2\delta_{val} \end{aligned} \tag{4.5}$$

Subsequently, we can calculate the trustworthiness scores based on the quality of the contributed rules. We follow the model in [95], such that the score T^m_a of cc_a after contributing m detection rules can be calculated as follows:

$$T^m_a = (1-\gamma) \sum_{j=1}^{m} \gamma^{(m-j)} t_{a,j} \tag{4.6}$$

where $0 < \gamma \le 1$ is the decaying constant to give more weights to recent contributions relative to older ones. To get the lower and upper bounds for T^m_a, let us assume that $t_{a,j}$ is constant. Then, from (4.6) we get

$$\begin{aligned} T_a^m &= \frac{1-\gamma^m}{1-\gamma}(1-\gamma)t_{a,j} \\ &= (1-\gamma^m)t_{a,j}. \end{aligned} \tag{4.7}$$

From (4.5) and (4.7), we get the lower and upper bounds of T_a^m as

$$\begin{aligned} 0 \leq T_a^m &\leq (1-\gamma^m)\delta_{val}, && \text{for } \delta_{inv} < 2\delta_{val} \\ 0 \leq T_a^m &< (1-\gamma^m)\delta_{inv}/2, && \text{for } \delta_{inv} \geq 2\delta_{val} \end{aligned} \tag{4.8}$$

We implement the trust calculation for both $t_{a,j}$ and T_a^m in the SC_{trm} smart contract, using which each cv can cooperate to calculate the score and collaboratively determine valid rules. In addition, the SC_{trm} smart contract also holds the computed trust scores. As the calculation and storage are offloaded to blockchain $\mathbb{B}$, regular nodes cr can conveniently query blockchain $\mathbb{B}$ and examine $t_{a,j}$ and T_a^m scores to infer the quality of the detection rules, thus reducing the computation loads. We elaborate on how the TRM mechanism is used in practice in Section 4.4.

4.4 Collaborative Intrusion Detection System

In this section, we describe the mechanism of our proposed CIDS framework. We present the overview of the underlying processes as a sequence diagram in Figure 4.3, which displays the process from when cc_a submits $r_{a,j}$ until the rule is validated and stored in the rules database. Refer to Table 4.1 for a summary of the notations used in this chapter.

We assume there is a set of n online validator nodes $C_V = \{cv_1, cv_2, \ldots, cv_n\}$ to initialise the network $\mathbb{C}$ and to deploy the smart contracts SC_{trm} and SC_{str}. As we support less than 1/3 byzantine nodes for PBFT fault tolerance, (see Section 4.2.2), we require at least $n = 4$ online cv. Note that the CIDS network $\mathbb{C}$ is a private network, thereby all CIDS nodes should be known in advance, although they are not necessarily trusted. We presume that each new CIDS node is able to generate a public-private key pair $\{k_p, k_s\}$. A new CIDS node can join $\mathbb{C}$ as a regular node

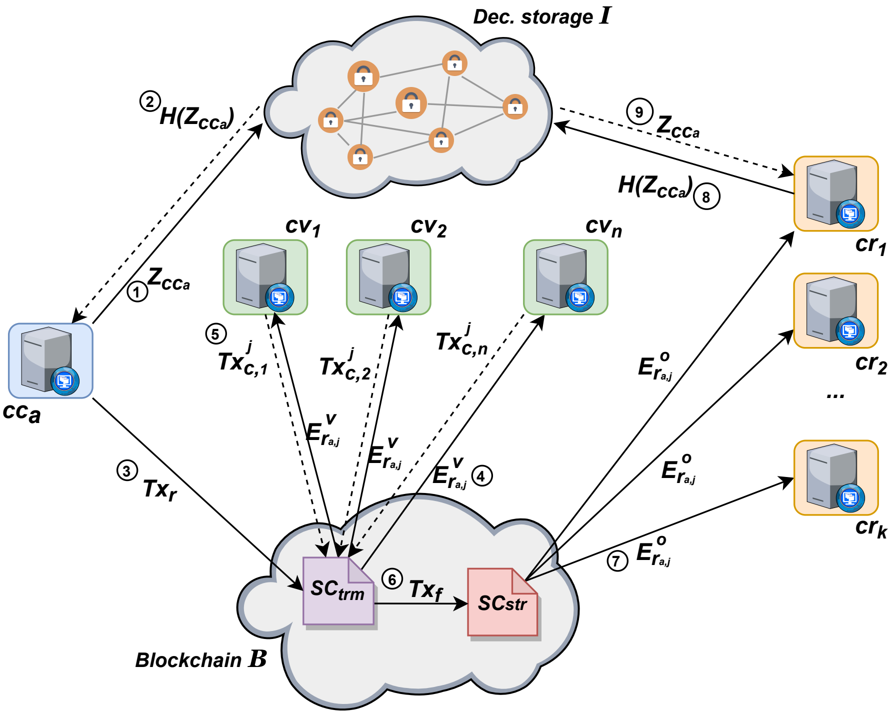

Figure 4.3: The workflow of the proposed CIDS platform.

by sending a request Req_{cr} over a secure channel to any cv:

$$Req_{cr} = \langle k_p, attr_{cr}, timestamp, Sign_{cr} \rangle \tag{4.9}$$

where k_p is the node's public key, $attr_{cr}$ is an attribute which also includes its IP address and a unique identifier, while $Sign_{cr}$ is the signature of the message. Once the request is approved, the corresponding validator node will give a response containing the details for IPFS connection (e.g., the IP address and the hash address of the bootstrap node) and the addresses of both SC_{trm} and SC_{str}, using which the nodes can access the network rules database R_{db} and obtain updates when a new detection rule has been added to R_{db}.

SC_{trm} facilitates collaborative rule validation to determine whether a new detection rule should be included to R_{db} based on the votes from each cv. To submit a new rule $r_{a,j}$, a cc_a node is required to invoke a function on SC_{trm} which then triggers a blockchain event to notify all regulator nodes $\forall cr \in C_R$ about the newly submitted rule. Note that this smart contract-based consensus mechanism is not intended to replace the built-in consensus algorithm as it serves a different purpose and runs on top of the built-in consensus algorithm of the underlying blockchain platform. Although the validator nodes are relatively trusted nodes in $\mathbb{C}$, we incorporate our rule validation consensus mechanism to mitigate if some cv's are compromised. Next, we explain our proposed rule validation consensus mechanism.

Let cc_a be a contributor node that is about to contribute a new detection rule $r_{a,j}$ to R_{db}. Firstly, cc_a needs to add a zip file $Z_{cc_a} = \left\langle r_{a,j}, M_{r_{a,j}} \right\rangle$ to the decentralised storage network $\mathbb{I}$, where $M_{r_{a,j}}$ is the description of $r_{a,j}$. To provide compliance and interoperability, we follow Intrusion Detection Message Exchange Format (IDMEF) [93] as the format of $M_{r_{a,j}}$. Subsequently, cc_a obtains the hash address of Z_{cc_a}, denoted $H(Z_{cc_a})$, required for file retrieval. cc_a is then required to invoke transaction Tx_r to SC_{trm}:

$$Tx_r = [\, H(Z_{cc_a}) \parallel timestamp \parallel Sig_{cc_a} \,] \tag{4.10}$$

where Sig_{cc_a} is the signature on $hash(H(Z_{cc_a}) \| timestamp)$ using the signing key $k_{s_{cc}}$ for authentication. Subsequently, SC_{trm} triggers a blockchain event $E^{v}_{r_{a,j}}$ to notify all validator nodes that a new rule $r_{a,j}$ has been added to the queue for validation.

Using the hash address $H(Z_{cc_a})$, each validator node cv retrieves and extracts Z_{cc_a} from storage $\mathbb{I}$ for off-chain validation [25]. Here, each cv compares $r_{a,j}$ against its local database to confirm whether $r_{a,j}$ performs as per $M_{r_{a,j}}$. The validator also inspects if $r_{a,j}$ has been previously submitted either by the same cc_a or another node to avoid ballot-stuffing attack. Depending on the validation results, a validator node cv_e may either approve or reject $r_{a,j}$, along with a score $S^{e}_{a,j}$ that indicates the quality of $r_{a,j}$. To submit the vote about the validity of $r_{a,j}$, all validator nodes submit $Tx^{j}_{c,e}$ to SC_{trm}:

$$Tx^{j}_{c,e} = [\, \varphi^{e}_{j} \parallel S^{e}_{a,j} \parallel timestamp \parallel Sig_{cv_e} \,] \tag{4.11}$$

where $\varphi^{e}_{j} \in \{1, -1\}$ is the validation result to indicate a valid (1) or invalid rule (-1) and Sig_{cv_e} is the signature on $hash(\varphi^{e}_{j} \| S^{e}_{a,j} \| timestamp)$ using signing key $k_{s_{cv}}$, which is used for authentication. We adapt a weighted majority rule [96] to make a decision on the validity of $r_{a,j}$. We define a decision rule $D_c : \mathbf{x}_j \to \{-1, 1\}$ which receives $\mathbf{x}_j = (Tx^{j}_{c,1}, Tx^{j}_{c,2}, \ldots, Tx^{j}_{c,n})$ as an input and outputs a decision $\{1, -1\}$, where 1 and -1 indicate valid and invalid rules, respectively. We define D_c

as follows:

$$D_c(\mathbf{x}_j) := \begin{cases} 1, & \text{if } {}^1\!/_n \sum_{e=1}^{n} S_{a,j}^{e} \varphi_j^e \geq q \ , \\ -1, & \text{otherwise} \end{cases} \tag{4.12}$$

where $q \in (0,1]$ is the threshold of a valid rule, which can be determined via a heuristic method.

SC_{trm} proceeds with the decision making process, once all votes from all cv have been received, as described in Algorithm 4.1. A valid $r_{a,j}$ would be appended to R_{db}, while an invalid $r_{a,j}$ would be ignored. SC_{trm} calculates the trust score $t_{a,j}$ and T_a^m as defined in (4.1) and (4.6), respectively. Subsequently, SC_{trm} invokes Tx_f to SC_{str}:

$$Tx_f = [\, H(Z_{cc_a}) \parallel t_{a,j} \parallel timestamp \,] \tag{4.13}$$

to include the newly approved $r_{a,j}$ to the rules database R_{db}. To notify all regular nodes about the new rule, SC_{str} triggers a blockchain event $E^o_{r_{a,j}}$. A regular node cr can now retrieve Z_{cc_a} from $\mathbb{I}$ using the address $H(Z_{cc_a})$. After inspecting the score $t_{a,j}$ and metadata $M_{r_{a,j}}$, cr may opt to include $r_{a,j}$ into its local IDS rules database R_{loc} for better detection of new attacks.

Unlike typical consensus mechanisms where each device sends messages to each other (broadcast message), in our proposed mechanism, the participating nodes are only required to send a message to the smart contract, which acts as an aggregator. Thus, our proposed mechanism would work on any blockchain instantiation that supports smart contracts for on-chain logic execution.

Algorithm 4.1 Consensus mechanism for rule validation.

```
Require: $Tx_c$, $n$, $q$ and $C_R$
Output: $Tx_f$ or $\emptyset$
$r\_count \leftarrow getState()$                ▷ from the blockchain
if $r\_count == (n-1)$ then                       ▷ all validation received
    calculate $t_{a,j}$                               ▷ as in (4.1)
    update $T_a^m$                                    ▷ as in (4.6)
    calculate $D_c(\mathbf{x}_j)$                     ▷ as in (4.12)
    if $D_c(\mathbf{x}_j) == 1$ then
        return $Tx_f$                                 ▷ $r_{a,j}$ is added to $R_{db}$
    else
        return $\emptyset$                            ▷ $r_{a,j}$ is rejected
    end if
else
    $saveToState(r\_count{+}{+})$                    ▷ write to blockchain
    return $\emptyset$
end if
```

4.5 Performance Evaluation

This section presents the performance evaluation of the proposed CIDS architecture based on our proof of concept (POC) implementation. We first describe the details of the POC that was implemented on a lab-scale private Ethereum network. Then, we present the experimental results to show the feasibility of our solution, which include trust evaluation for honest and malicious nodes and the blockchain performance with regards to latency and gas consumption.

4.5.1 Implementation Details

We opted for a private Ethereum blockchain as our POC platform, as Ethereum natively supports smart contracts written in a Turing complete programming language, Solidity [89]. Ethereum utilises a decentralised environment named Ethereum Virtual Machine (EVM), as a trusted and secure platform for decentralised computation [53]. However, we note that our proposed CIDS architecture can also be implemented in any blockchain platform that offers support for smart contracts, for instance Hyperledger Fabric and Sawtooth.

We built a lab-scale testbed which comprises 15 nodes of Raspberry Pi with differ-

Table 4.2: Evaluation details.

Parameters	***Values***
No. cv	4
No. cc	4
No. cr	8
λ	0.85
ϕ	2
τ	0
δ_{val}	0.85
δ_{inv}	0.9
γ	$[0.90, 0.85, 0.80]$

ent specifications (7 Raspberry Pi 3 and 8 Raspberry Pi 4) and a Lenovo ThinkCentre mini PC (8GB RAM, 2.9 GHz quad-core Intel Core i5 CPU) as the platform to run our experiments. We selected Raspberry Pi computers to imitate the different smart devices with varying capacity, while we used the Lenovo ThinkCentre as a more powerful node. To orchestrate the CIDS nodes, we utilise Docker containers[1] running on the testbed using which we simulated a total of 16 nodes: 4 validator, 4 contributor and 8 regular nodes. We built a private Ethereum network with a single miner and a private IPFS network for the decentralised storage. We utilise an open-source IDS, Snort[2], as the signature-based IDS platform and geth as the Ethereum client. We wrote a Python v3.8.10 and a bash script to control and simulate interactions between CIDS nodes with web3py v5.11 and py-solc v3.2.0 libraries as the middleware for Ethereum connection and smart contracts compilation, respectively. We implemented the smart contracts, i.e., SC_{trm} and SC_{str} in Solidity v0.6.6 [89], which offers built-in libraries for computation and verification of hashes efficiently. In Table 4.2, we present a summary of the parameters used in the evaluation.

4.5.2 Evaluation Results

The following subsections present the experimental results of our POC implementation. We compare our results against challenge-based trust mechanism as the baseline [72], where a CIDS node sends challenge messages to another CIDS node

[1] `https://docs.docker.com/engine/`

[2] `https://www.snort.org/`

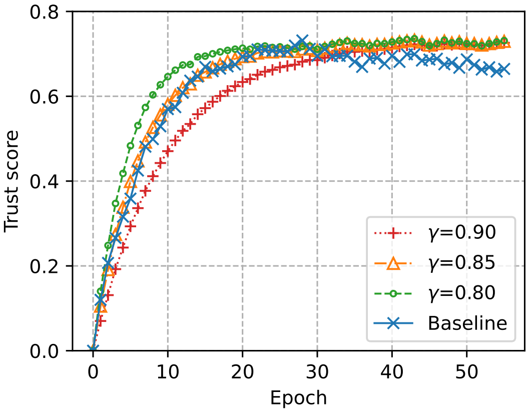

Figure 4.4: Convergence of T_a^m for an honest cc with varying γ, $\delta_{val} = 0.85$ and baseline [72].

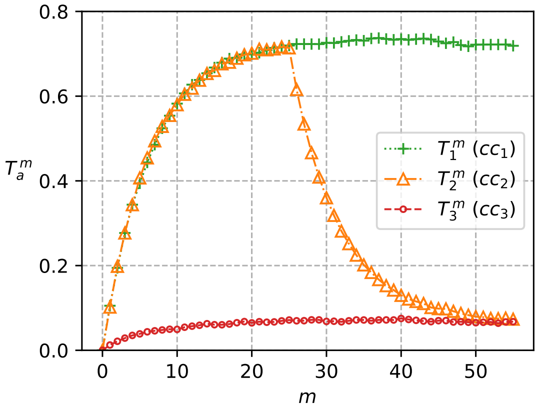

Figure 4.5: Trust score comparison for an honest and malicious cc ($\delta_{val} = 0.85$, $\delta_{inv} = 0.9$, $\gamma = 0.85$).

and waits for the responses (e.g., valid, invalid, unsure), using which the trustworthiness level is derived. To mimic similar characteristics with our proposal (e.g., trust score range), we set the values of the baseline's parameters as: forgetting factor $\lambda = 0.85$, severity level $\phi = 2$ and initial trust score $\tau = 0$.

Convergence of the trust score

We evaluate the convergence of the trust score T_a^m by varying the decaying constant γ and plot the results in Figure 4.4. This experiment simulates an honest cc which constantly contributes $m = 55$ valid detection rules. As we vary the value of γ from 0.9 to 0.8, different weights are allocated for the latest and the previous $t_{a,j}$ values in calculating the final T_a^m. However, as all contributions are valid, T_a^m converges to a similar level as illustrated in Figure 4.4, regardless of the γ value, which confirms the theoretical upper bound of T_a^m, described in (4.8). A subtle difference is that lower values of γ would result in a gradual growth of T_a^m, thereby γ can be tuned according to the practical settings. Figure 4.4 also shows a minor fluctuation of T_a^m, as T_a^m depends on $t_{a.i}$ values, which are weighted averages of different scores $S_{a,j}^e$ from each validator cv_e. Relative to the baseline, our trust evolution exhibits similar trend of positive interaction with less fluctuations.

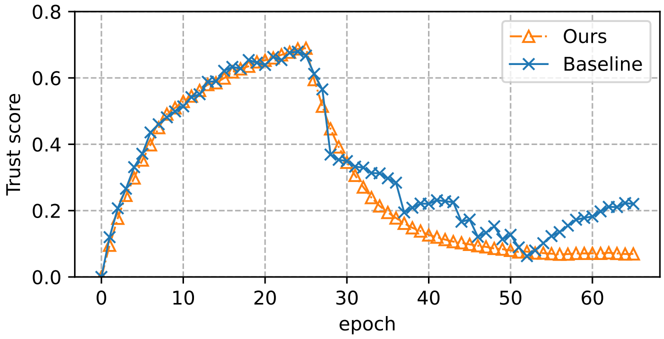

Figure 4.6: Quantitative comparison of trust evolution with baseline [72].

Trust evolution of honest and malicious nodes

We simulate three contributor nodes (cc_1, cc_2 and cc_3) to examine the evolution of T_a^m for different node behaviours over $m = 55$ contributions. First, cc_1 is an honest contributor node that consistently contributes trustworthy rules. Second, cc_2 initially submits trustworthy rules up to $m = 25$ and turns malicious by contributing false rules afterwards. Third, cc_3 is a compromised node that constantly submits poor detection rules for the entire simulation. We apply $\delta_{val} = 0.85$, $\delta_{inv} = 0.9$, $\gamma = 0.85$ as the parameters and plot the experimental results in Figure 4.5. While all T_a^m initially begins at 0, T_1^m and T_2^m continue to grow gradually in a similar rate and start to plateau approximately at $T_a^m = 0.75$. On the other hand, T_3^m saturates at approximately $T_a^m = 0.1$, as the validators are assigning very low $S_{a,j}^e$ scores (non-zero). At $m > 25$, T_2^m declines significantly and saturates at a similar score of T_3^m for the rest of the experiment. We note that the scores are in line with the theoretical bounds of T_a^m, as described in (4.5) and (4.8).

We plot the evolution of T_a^m against the baseline in Figure 4.6. Similar to Figure 4.5, node cc_2 initially acts honestly by sending valid responses until 25-th interaction, resulting in high trust scores, and turns malicious by sending invalid and *unsure* responses for epoch time > 25. Our trust evolution exhibits stable and gradual growth and decline when the node acts honestly and maliciously, respectively. However, the baseline suffers from undesired fluctuations when the node acts maliciously, i.e., epoch time > 25, as *unsure* responses perturb the slope of the curve, cf. Section 3.B of [72].

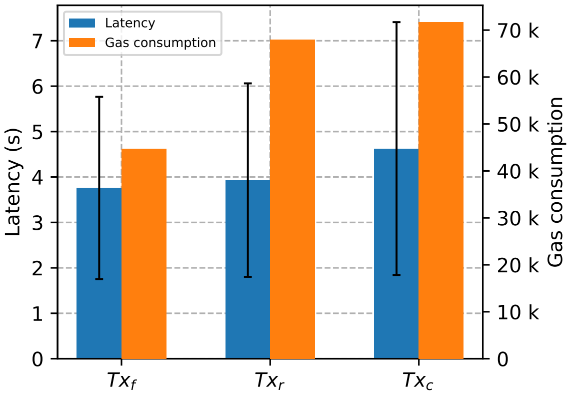

Figure 4.7: Comparison of the latency and gas consumption for rule confirmation, validation and submission.

Latency and gas consumption

Ethereum requires a fee for executing blockchain transactions, with Gas as the unit of measurement. The fee (Gas) is only necessary for each transaction that updates the blockchain state, whereas reading the blockchain state does not incur any fee. The Gas is calculated based on the required EVM opcodes during the execution of smart contract's functions [89], which are relatively small and often negligible. To examine the feasibility of our blockchain implementation, we measure the gas consumption and execution latency for the following smart contract functions: 1) rule submission (Tx_r), 2) rule validation (Tx_c) and 3) rule confirmation (Tx_f). We repeated the experiments 30 times and plot the results in Figure 4.7. The average execution latencies are relatively similar for all transactions, which fall within the range of 3 and 5 seconds. There is a relatively high variance of the latencies, as there is no assurance of when the miner processes and appends the transaction to the blockchain. Tx_f consumes the least Gas, while Tx_c consumes the most Gas among these three transactions, due to the different amount of required EVM opcodes (cf. (4.13), (4.10), (4.11)).

4.6 Chapter Summary

In this chapter, we proposed a trust framework to build a trustworthy CIDS architecture that continuously evaluates the trustworthiness of the CIDS nodes with regard to the detection rules contributed by each IDS node. We used signature-based CIDS as an illustrative example, while our architecture is blockchain-agnostic and could be implemented on any blockchain platform that supports smart contracts. We built a lab-scale testbed to evaluate our proposed architecture in a private Ethereum network. The experimental results indicated that our trust model exhibits stable and gradual evolution, which is absent in the baseline, as it suffers from undesired fluctuations. In addition, the results showed that the performance falls within the expected benchmarks of the Ethereum platform, signifying the feasibility of our concept.

Chapter 5

Resource Sharing for 6G-enabled IoT

This chapter presents a TRM framework to tackle the issues in providing secure and trustworthy resource sharing for 6G-enabled IoT, as outlined in Section 1.1.3 and 2.5.3. 6G-enabled IoT demands effective utilisation of scarce resources by means of trustworthy resource sharing, to provide massive scale in network capacity. While TRM provides assurance in resource sharing, novel technologies in 6G, such as Large Intelligent Surfaces (LIS) and incorporation of Non-Terrestrial Networks (NTN), requires an essential overhaul of conventional blockchain-based TRM. In addition, to avoid de-anonymisation attacks that may reveal sensitive information of critical network infrastructures, blockchain employs changeable keys in each transaction. However, this may render the TRM unusable, as the keys are now changeable, making the same node not recognisable by a single public key to which the trust and reputation scores are bound. Compared with the TRM frameworks in Chapter 3 and 4 which require the nodes to use static keys, the proposed TRM framework in this chapter allows the nodes to use changeable keys in each future transaction, making it impossible to trace the sharing history. To realise a privacy-preserving TRM, the framework introduces an Isolated Identity Chain, maintained by a set of semi-trusted authorities, in which the mappings between the main pseudonyms and changeable public keys are stored. Similar to both frameworks in Chapter 3 and 4, the TRM framework for resource sharing utilises smart contracts to implement an auditable recursive and efficient single-dimension trust computation to minimise unnecessary overheads, making the scheme suitable for 6G networks. The experimental results on a proof-of-concept implementation indicate the feasibility of the framework as it only incurs insignificant overheads.

5.1 Introduction

6G-enabled IoT demands effective utilisation of scarce resources to provide massive scale in network capacity. While blockchain-based resource sharing schemes have been proposed to enable effective resource allocation, it alone cannot ascertain the trust in the participating nodes. In fact, some network nodes may exhibit adverse behaviour when sharing their resources, for instance, by providing forged computation results. Trust and Reputation Management (TRM) has the potential to solve these trust issues, as network nodes can utilise TRM to determine the trustworthiness and reliability of the target nodes for task offloading by looking at their trust scores. However, changeable keys employed in blockchains to improve privacy-preservation may render the TRM unusable, as the same node is no longer recognisable by a single key to which the trust and reputation scores are bound.

In this chapter, a privacy-preserving TRM framework for 6G-enabled IoT is proposed, which specifically addresses its inherent trust issues that discourage sharing of resources. This chapter demonstrates how blockchain, with the aid of a TRM, could provide reliable assurance in the trust between participating nodes in the network, thus reducing the inherent risk of resource sharing. This chapter considers the use case of computation resource sharing in edge computing for 6G-enabled IoT, where the resource users offload computation tasks to resource owners, after which the trustworthiness of the resource owners are determined. The framework allows for both resource owners and users to employ changeable keys to obfuscate their transaction traces in the network, avoiding the de-anonymisation attacks that may reveal sensitive information of critical network infrastructures. To realise the privacy-preserving TRM, two interconnected blockchains are designed [38]. First, an Isolated Identity Chain (IIC), maintained by a set of semi-trusted authorities, where it stores the mapping between changeable keys and the main pseudonyms of a node. Also, the trust score calculation takes place in the IIC. Second, a Main Resource sharing Chain (MRC), which records the resource sharing transactions between the resource owners and users. The chapter also briefly discusses the design consideration for TRM to meet 6G performance requirements, which highlights the need of TRM re-designing. For instance, the proposed framework exploits smart contracts to provide auditable trust calculation, where a recursive trust computation is proposed to meet the efficiency requirements of trust calculation in 6G networks. The framework achieves rater and ratee anonymity, where the identities of both resource users (rater) and owners (ratee) are completely concealed. The proof-of-concept implementation is developed on lab-scale private Ethereum networks, where two Ethereum networks are deployed to realise IIC and MRC. However, the TRM framework is designed to be blockchain-agnostic, which can be implemented on any blockchain platform that supports smart contracts execution. The proposed framework is benchmarked with a baseline TRM framework, which employs no privacy-preservation. The experimental results signify the feasibility of the proposed framework, as it only incurs minimal overheads with regards to gas consumption and overall latency.

5.1.1 Chapter Contributions

In summary, the contributions of this chapters are as follows:

- The chapter proposes a privacy-preserving TRM framework for 6G-enabled IoT, to address the inherent trust issues in resource sharing, while preserving the privacy of the participating nodes. The proposed framework assigns reputation scores to each resource owner, using which resource users can select a particular resource owner to offload the computation tasks.

- The proposed framework allows the resource owners and users to employ changeable keys in each transaction, obfuscating their traces on the blockchain to preserve their privacy. The framework thus achieves both ratee and rater anonymity, where the identities of both resource users (rater) and owners (ratee) are completely concealed.

- To provide TRM functionalities with changeable keys, the framework utilises two interconnected blockchains, namely i) IIC, which records the node's identities and calculates their reputation scores; and ii) MRC, which records the corresponding resource sharing transactions.

- The chapter discusses TRM design considerations to meet 6G performance requirements. The chapter argues that TRM should be re-designed to fit with the new requirements and technologies in 6G networks, wherein blockchain would play a significant supporting role.
- A proof-of-concept implementation of the framework is developed on lab-scale private networks, where two Ethereum networks are deployed to realise IIC and MRC. The experimental results signify the feasibility of our framework as it only incurs minimal overheads relative to the baseline.

5.1.2 Chapter Organisation

The rest of this chapter is organised as follows. Section 5.2 and 5.3 describe the model of the proposed framework and the resource sharing mechanism, respectively. Section 5.4 presents the proof-of-concept implementation of the framework, while the chapter is concluded in Section 5.5.

5.2 Our Proposed TRM Framework

In this section, we first discuss the challenges of meeting 6G performance requirements for TRM. Then, we outline the system model of our proposed TRM framework, which covers the blockchain, identity and resource sharing models. We also describe the privacy-preserving trust and threat models along with the assumptions made. We present an overview of the system model in Figure 5.2.

5.2.1 Design Considerations for TRM in 6G

Multiple vision papers note that the 6G requirements would be driven by novel enabling technologies and trends, such as Large Intelligent Surfaces (LIS), incorporation of Non-Terrestrial Networks (NTN), and Convergence of Communications, Computing, Control, Localisation, and Sensing (3CLS) [5], [97]. As shown in Figure 5.1, we argue that TRM should also be re-designed to fit with these new requirements and technologies, wherein blockchain would play a significant supporting role [6]. We elaborate on the need of efficient trust and reputation score calculation for 6G-enabled IoT, as discussed in Section 5.2.3.

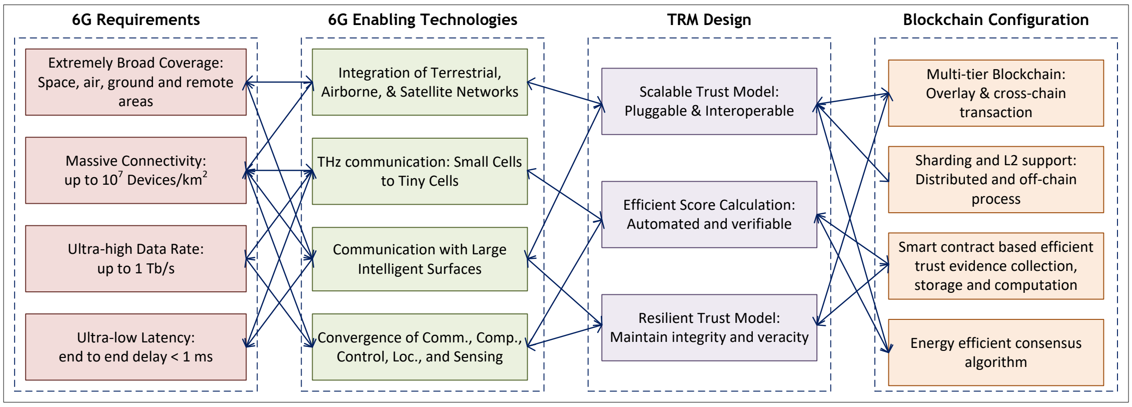
6G Requirements
6G Enabling Technologies
TRM Design
Blockchain Configuration
Extremely Broad Coverage: Space, air, ground and remote areas
Massive Connectivity: up to 10^7 Devices/km^2
Ultra-high Data Rate: up to 1 Tb/s
Ultra-low Latency: end to end delay < 1 ms
Integration of Terrestrial, Airborne, & Satellite Networks
THz communication: Small Cells to Tiny Cells
Communication with Large Intelligent Surfaces
Convergence of Comm., Comp., Control, Loc., and Sensing
Scalable Trust Model: Pluggable & Interoperable
Efficient Score Calculation: Automated and verifiable
Resilient Trust Model: Maintain integrity and veracity
Multi-tier Blockchain: Overlay & cross-chain transaction
Sharding and L2 support: Distributed and off-chain process
Smart contract based efficient trust evidence collection, storage and computation
Energy efficient consensus algorithm

Figure 5.1: The 6G Requirements and the improvements of TRM with the help of blockchain.

5.2.2 System Model

We model our proposed framework as $\mathbb{RS} = (NA, RO, RU)$. We denote each of the actors, namely $NA = \{na_1, na_2, \ldots, na_{|NA|}\}$, $RO = \{ro_1, ro_2, \ldots, ro_{|RO|}\}$ and $RU = \{ru_1, ru_2, \ldots, ru_{|RU|}\}$ as a set of network authorities, resource owners and resource users, respectively. In our model, a resource owner $ro_o \in RO$ owns and manages a set of computation resources $CR_o = \{cr_{o,1}, cr_{o,2}, \ldots, cr_{o,|CR_o|}\}$. The model's goal is to allow ro_o to share resource $cr_{o,c} \in CR_o$ with resource user $ru_u \in RU$ for some fee $f_{o,c}$ while preserving both ro_o's and ru_u's privacy. Network authorities NA are the parties managing the overall cellular network, e.g., the government organisations, which will be the key points for the on-boarding process (cf. Section 5.3.1). In our model, resource owners RO are the network nodes with a multitude of computation power, e.g., Mobile Network Operators, Mobile Virtual Network Operators and Service Providers. On the other hand, resource users RU are the nodes in the network with a need of computation, but they have relatively lower computation power, e.g., User Equipment (UE) and IoT devices. We present a summary of notations and their description in Table 5.1.

Blockchain model: We consider a blockchain agnostic model, where we assume that there is a pre-existing consortium blockchain for the 6G-IoT network with built-in cryptocurrency [98]. We consider a hierarchical blockchain network to realise the MRC and IIC, denoted $\mathbb{BC} = (B_m, \widehat{B}_s)$, where B_m is the main-chain (MRC) and $\widehat{B}_s$ is a set of identity side-chains (IIC). We require the authorities NA, resource owners

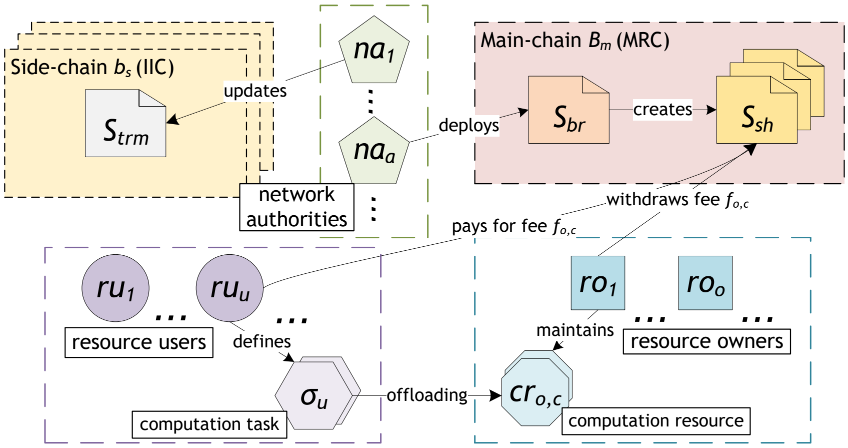

Figure 5.2: The system model of the proposed framework which consists of two blockchains, namely MRC (main-chain) and IIC (side-chain). The model defines three user types, namely network authority, resource owners and users.

RO and users RU to run blockchain nodes for participation in the network $\mathbb{BC}$. We consider two types of blockchain nodes, namely full and light node. While full nodes keep the whole copy of the blockchain and participate in the mining process, light nodes only store the block headers and do not participate in block mining. Here, a blockchain node may opt to run a light node to save resources, depending on their computational and storage capacity.

All nodes in the consortium network $\mathbb{BC}$ have access to the main blockchain B_m. However, $\widehat{b}_{s,z} \in \widehat{B}_s$ is only accessible to the network authorities that manage $\widehat{b}_{s,z}$. We utilise B_m for record keeping and traceability of resource sharing transactions (cf. Table 5.2), while sidechain $\widehat{b}_{s,z}$ is used to provide relevant information for trust score calculation. Here, NAs act as oracles for inter-ledger communications [99].

We define three smart contracts with different purposes. First, a broker contract $\mathbb{S}_{br}$ matches the resource supply and demand between ro_o and ru_u in a single-price auction mechanism (cf. Section 5.3.3). Second, a sharing contract $\mathbb{S}_{sh}$ records and enforces sharing agreements between ro_o and ru_u, which is unique for each resource sharing agreement. Third, TRM contract $\mathbb{S}_{trm}$ stores the mapping between identities and resource advertisements and calculates the trust and reputation scores. While there is only one $\mathbb{S}_{br}$ to which ro_o advertises their resources, there are multiple $\mathbb{S}_{sh}$ contracts, as $\mathbb{S}_{sh}$ is created every time ro_o and ru_u agree to share a resource $cr_{o,c}$. As shown in Figure 5.2, $\mathbb{S}_{br}$ and $\mathbb{S}_{sh}$ are deployed on B_m, while $\mathbb{S}_{trm}$ is deployed on the

Table 5.1: Notations and their description

Notations	***Description***
$na_a \in NA$	network authority a
$ro_o \in RO$	resource owner o
$ru_u \in RU$	resource user u
$cr_{o,c} \in CR_o$	computation resource c of ro_o
$\sigma_u = \left\langle \sigma_u^\lambda, \sigma_u^\delta \right\rangle$	task $u = \langle$computation load and data size$\rangle$
d_u	prescribed deadline to finish σ_u
t_{sh}	binary experience to tx_{sh}
R_o	reputation of ro_o
B_m	Main Resource sharing Chain (MRC)
$\widehat{b}_{s,z} \in \widehat{B}_s$	side-chain z in the side-chain set $\widehat{B}_s$ (IIC)
$\mathbb{S}_{br}$, $\mathbb{S}_{sh}$, $\mathbb{S}_{trm}$	broker, sharing and TRM contracts
$\langle pk_o^*, sk_o^* \rangle$	main public and secret key-pair of ro_o
$\langle pk_{o,x}, sk_{o,x} \rangle \in KP_o$	changeable public and secret key-pair of ro_o
$g_{\mathcal{M}} \colon KP_j \mapsto pk_j^*$	mapping function for pseudonyms
e_{pos}, e_{neg}	weights for positive and negative experiences
$0 < \alpha < 1$	time discounting parameter

side-chain $\widehat{b}_{s,z}$. We elaborate on the details of the smart contracts in Section 5.3.3.

Identity model: We employ public-key cryptography as the method to obtain valid credentials, e.g., SECP-256k1 key generation [53]. Each na_a, ro_o and ru_u are identifiable by their main public keys (i.e., pseudonyms), denoted pk_a^*, pk_o^* and pk_u^*, respectively. While na_a uses the same key over its operational period, ro_o and ru_u use different keys in each transaction to conceal their main keys and avoid de-anonymisation attacks [36]. Each user $j \in \{RO, RU\}$ maintains a set of w public-secret key pairs $KP_j = \{\langle pk_{j,1}, sk_{j,1} \rangle, \langle pk_{j,2}, sk_{j,2} \rangle, \ldots \langle pk_{j,w}, sk_{j,w} \rangle\}$, where $pk_{j,w}$ and $sk_{j,w}$ denote the public and secret key, respectively.

We consider a Pseudo Random Number Generator (PRNG) [100], using which ro_o and ru_u select a pair $\{\langle pk_{o,w}, sk_{o,w} \rangle, \langle pk_{u,w}, sk_{u,w} \rangle\}$ for each transaction. There exists a mapping function $g_{\mathcal{M}} \colon KP_j \mapsto pk_j^*$ such that for a given key $pk_{j,w}$, we can obtain the main key pk_j^*. To avoid Sybil attacks, these changeable keys are signed by NA. We describe the key registration and revocation in the onboarding process (cf. Section 5.3.1).

Resource sharing model: We follow a task-offloading approach to model the

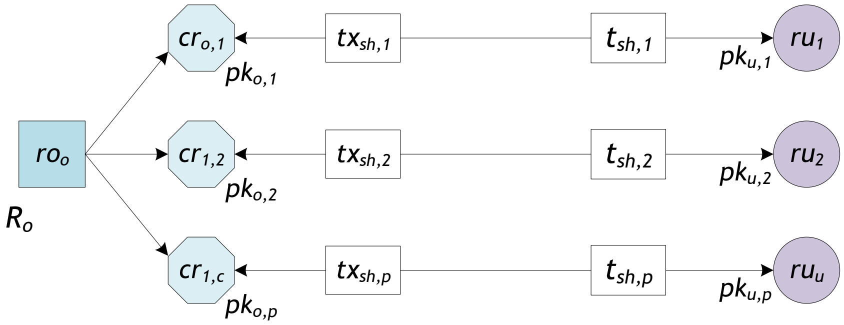

Figure 5.3: The trust and reputation relationship model. Although resources $cr_{o,1}$, $cr_{o,2}$ and $cr_{o,c}$ are owned by ro_o, they are identified by different keys $pk_{o,1}$, $pk_{o,2}$ and $pk_{o,p}$. The score R_o can be calculated by gathering $tx_{sh,1}, \dots, tx_{sh,p}$ and $t_{sh,1}, \dots, t_{sh,p}$ to construct $\mathbf{T}_o$.

resource sharing [101]. We denote a computation resource $cr_{o,c} \in CR_o$ as $cr_{o,c} = \left\langle cr^{id}_{o,c}, cr^{\lambda}_{o,c}, f_{o,c} \right\rangle$, where $cr^{id}_{o,c}$ is its ID, $cr^{\lambda}_{o,c}$ is its computation cycles per second (Hz) and $f_{o,c}$ denotes the access fee. To share $cr_{o,c}$, ro_o submits a listing transaction tx_{ls} to $\mathbb{S}_{br}$ contract using one of its keys $pk_{o,x}$ (cf. (5.9)), which would assign $cr^{id}_{o,c} \leftarrow pk_{o,x}$. Then after the sharing agreement, ru_u offloads a computation task $\sigma_u = \left\langle \sigma^{\lambda}_u, \sigma^{\delta}_u \right\rangle$ to $cr_{o,c}$, where σ^{λ}_u and σ^{δ}_u denote the computation load (computation cycles) and the data input size, respectively. ro_o is expected to complete task σ_u within the prescribed deadline

$$\frac{\sigma^{\lambda}_u}{c^{\lambda}_{o,c}} \leq d_u. \tag{5.1}$$

It is not possible to reshare the computation resource $cr_{o,c}$ with another resource user while it is still processing σ_u. Transaction tx_{sh} records the sharing agreements (cf. 5.12), including the the obligations of both parties, upon which the trust and reputation scores are updated. We elaborate the resource sharing mechanism in Section 5.3.

5.2.3 Privacy-preserving Trust Model

We incorporate Trust and Reputation Management (TRM) to determine the reliability and trustworthiness of ro_o, while maintaining the privacy of both ro_o and ru_u. We base our privacy-preserving model on a conventional trust model in [95], as it is of the same trust dimensionality (see Section 2.2.4) and it offers flexibility in adjusting the trust growth. However, the current model requires changeable keys for privacy preservation, which is not supported in [95]. We thus add a privacy preservation mechanism on top of the trust model, which is explained as follows.

In our privacy-preserving TRM, we define two distinct scores. First, trust score t_{sh} is the trust rating of transaction tx_{sh}. Second, reputation score R_o is an aggregation of multiple t_{sh} belonging to the same ro_o, which illustrates a global view of the behaviour of ro_o over a period of time. Note that we assign the trust rating t_{sh} to transaction tx_{sh} instead of ro_o, as both ru_u and ro_o use changeable public keys, i.e., no direct $ru_u \not\leftrightarrow ro_o$ relationship. Consequently, ru_u does not know the real owner of $cr_{o,c}$. We present the trust relationship in Figure 5.3.

In essence, t_{sh} represents a binary experience of tx_{sh}, which is assigned as follows:

$$t_{sh} = \begin{cases} e_{pos}, & \text{if } \sigma_u \text{ completion time } \leq d_u \\ e_{neg}, & \text{otherwise,} \end{cases} \tag{5.2}$$

where d_u denotes the prescribed task completion deadline (cf. (5.1)), while e_{pos} and e_{neg} are the weights for positive and negative experiences, respectively, such that $e_{pos} > e_{neg} > 0$. In other words, we assign $t_{sh} = e_{pos}$ when task σ_u is successfully completed within d_u, indicating a positive experience, while we assign $t_{sh} = e_{neg}$ when task σ_u is not completed within d_u, i.e., a negative experience.

Suppose ro_o has shared its resources p times in network $\mathbb{BC}$. We can then arrange a $4 \times p$ matrix

$$\mathbf{T}_o = \begin{bmatrix} tx_{sh,1} & tx_{sh,2} & \dots & tx_{sh,p} \\ pk_{o,1} & pk_{o,2} & \dots & pk_{o,p} \\ pk_{u,1} & pk_{u,2} & \dots & pk_{u,p} \\ t_{sh,1} & t_{sh,2} & \dots & t_{sh,p} \end{bmatrix}, \tag{5.3}$$

where the columns resemble the order of the interactions (i.e., $tx_{sh,2}$ happens after $tx_{sh,1}$). Here, $\mathbf{T}_o$ captures the track record of ro_o in sharing its resources using a changeable key $pk_{o,p}$ with user $pk_{u,p}$ recorded in transaction $tx_{sh,p}$ with a trust rating $t_{sh,p}$. Subsequently, we can calculate the value of R_o after p sharing instances as follows:

$$R_o(p) = (1-\alpha)\sum_{n=1}^{p} t_{sh,n}\alpha^{(p-n)}, \tag{5.4}$$

where α is the time discounting parameter ($0 < \alpha < 1$). We introduce α to give higher weights to recent experiences than the older ones.

Without loss of generality, let us assume $p \to \infty$ and all t_{sh} are positive experiences, then from (5.2) and (5.4) it follows that

$$\begin{aligned} R_o(\infty) &= (1-\alpha)e_{pos}\sum_{n=0}^{\infty}\alpha^n \\ &= e_{pos}, \end{aligned} \tag{5.5}$$

which is the upper boundary of R_o. Likewise, when all t_{sh} are negative experiences, we get $R_o(\infty) = e_{neg}$. Therefore we have the upper and lower boundary of R_o:

$$e_{neg} \leq R_o(p) \leq e_{pos}, \tag{5.6}$$

which indicates that ro_o is completely untrusted and unreliable when $R_o = e_{neg}$, but it is reliable when $R_o = e_{pos}$.

In addition, R_o can also be calculated by a simple recursion, as shown below:

$$\begin{aligned} R_o(p+1) &= (1-\alpha)\sum_{n=1}^{(p+1)} t_{sh,n}\alpha^{(p+1)-n} \\ &= \alpha(1-\alpha)\sum_{n=1}^{p} t_{sh,n}\alpha^{(p-n)} + (1-\alpha)t_{sh,(p+1)} \\ &= \alpha R_o(p) + (1-\alpha)t_{sh,(p+1)}. \end{aligned} \tag{5.7}$$

We utilise $\mathbb{S}_{sh}$ and $\mathbb{S}_{trm}$ contracts to provide automation and transparency on how the trust model is enforced. Note that, as ro_o uses changeable keys in each tx_{sh} (cf. (5.3)), it is not possible to calculate R_o by only using the evidence on B_m. We show how we address the challenge in calculating the reputation score in Section 5.3.2.

5.2.4 Threat Model and Assumptions

We consider the adversaries as the non-cooperative resource owners who are able to launch the following types of attacks:

- *Ballot-stuffing:* Adversaries may try to illegitimately increase their reputation by transacting with their own.
- *Bad-mouthing:* Adversaries may try to decrease others' reputation by submitting fake ratings.
- *Sybil attack:* To launch ballot stuffing or bad-mouthing attacks, adversaries may create multiple forged identities to increase the chance of the attack.
- *Selfish attack:* A resource owner may use less computation power or not execute the offloaded task only to get the incentives. A node may also give forged computation results.
- *De-anonymisation attack:* Adversaries trying to infer the original information about the resource owners to their benefits, such as resource ownership and transaction trails.

We presume that the authorities NA are partially trusted, meaning that there may be an unreliable node $na_a \in NA$ at any given time. However, we require that the number of reliable authority nodes are not less than $3i + 1$ to maintain the $\widehat{B}_s$ from byzantine nodes [88], where i is the number of unreliable authority nodes. We also presume that the network authorities have the ability to validate the identities and the resources during registration (cf. Section 5.3.1).

We assume that the offloaded tasks contain no privacy-sensitive information. We further presume that the offloaded task are hard to compute but easy to verify, such as those used in blockchain mining process [53] and in verifiable computing [102]. While there might be a small communication delay to offload the task, we presume these are negligible and thus not included in our model. All nodes are bound to the blockchain's cryptographic primitives.

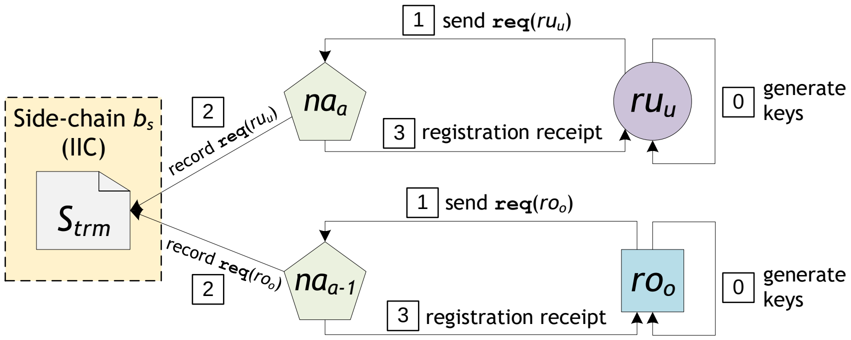

Figure 5.4: The overview of the on-boarding process, which illustrates both ru_u and ro_o create and register a set of public-private key pairs to the network authorities na_a and na_{a-1}. To record the *registration*, network authorities store the registration request on the IIC side-chain.

5.3 Resource Sharing Mechanism

In this section, we describe the mechanics of our proposed privacy-preserving resource sharing framework (cf. Section 5.2.3). We begin by explaining the on-boarding process and the reputation calculation. Then, we describe the procedures for our resource sharing framework, which illustrates the usage of reputation scores in practice.

5.3.1 On-boarding Process

The on-boarding process governs the *registration* and *revocation* of public-secret key pairs, through which resource owners and users obtain access to the MRC. Recall that each resource owner and user use changeable keys in each transaction. To avoid Sybil attacks, these keys should be signed by one of the network authorities during the on-boarding process. We allow the key pairs to be registered in batch to reduce the load of the network authorities in handling the on-boarding requests. We present an overview of the on-boarding process in Fig. 5.4.

Registration: Suppose ro_o is about to participate in the consortium network $\mathbb{BC}$. Initially, ro_o generates a set of w key pairs, denoted $KP_o = \{\langle pk_{o,1}, sk_{o,1}\rangle, \ldots, \langle pk_{o,w}, sk_{o,w}\rangle\}$. Next, ro_o sends a registration request $\texttt{req}(ro_o)$ over a secure private channel to any available NA, e.g., na_a. We formally define

$\texttt{req}(ro_o)$ as follows:

$$\texttt{req}(ro_o) = \langle \texttt{attr}, \langle pk_{o,1}, \ldots \rangle, CR_o, k, \texttt{tstmp}, \texttt{sig} \rangle, \tag{5.8}$$

where $\texttt{attr}$ is the attributes of ro_o (including its main public key pk_o^*), k is the amount of deposited funds, $\texttt{tstmp}$ is the timestamp, and $\texttt{sig}$ is the signature on $\texttt{hash}(\texttt{attr}, \langle pk_{o,1}, \ldots \rangle, CR_o, k, \texttt{tstmp})$ using signing key sk_o^*, which is used for authentication. We require ro_o to deposit the fund k for each w key pairs, to avoid Sybil attacks, as here the key registration requires some deposits. These deposits are returned when ro_o revokes the keys.

Network authority na_a validates and records the request $\texttt{req}(ro_o)$ onto side-chain $\widehat{b}_{s,z}$, so the process is auditable by other authorities. Upon successful validation, na_a returns the receipt to ro_o, which includes the details of the main-chain B_m and the signatures of na_a on each registered public key to ascertain the integrity and validity of the keys. Note that, ro_o can register more keys or resources at a later time by submitting another $\texttt{req}(ro_o)$.

Similarly, ru_u follows identical on-boarding procedure, without including the resource list in the request

$$\texttt{req}(ru_u) = \langle \texttt{attr}, \langle pk_{u,1}, \ldots \rangle, k, \texttt{tstmp}, \texttt{sig} \rangle.$$

Revocation: To revoke a subset of resources or keys, ro_o and ru_u send a revocation request $\texttt{rvk}(\cdot)$ over a secure private channel to any available NA, e.g., na_a. For instance the request

$$\texttt{rvk}(ro_o) = \langle \langle pk_{o,1}, \ldots \rangle, CR_o, \texttt{tstmp}, \texttt{sig} \rangle$$

revokes keys $\langle pk_{o,1}, \ldots \rangle$ and resources CR_o from the network. Upon validation, na_a indicates the revoked keys as obsolete and publishes the list of revoked keys to B_m, so that everyone can publicly see whether certain keys have been revoked. In addition, ro_o and ru_u receive the deposited funds back.

Algorithm 5.1 Trust and reputation score calculation.

```
Require: $tx_{sh}$, and $tx_{fl}$ or $tx_{rf}$
Output: updated $R_o$
$pk_o^* \leftarrow g_{\mathcal{M}}(pk_{o,x})$                ▷ obtain the main key
if $\sigma_u$ completion $\leq d_u$ then            ▷ sharing experience
    $t_{sh} \leftarrow e_{pos}$
else                         ▷ also for when $\sigma_u$ is not completed
    $t_{sh} \leftarrow e_{neg}$
end if
update $R_o \leftarrow R_o(p+1)$                 ▷ as defined in (5.7)
save_to_state($R_o$)                            ▷ write to $\mathbb{S}_{trm}$
return $R_o$
```

5.3.2 Trust and Reputation Calculation

The nature of changeable keys in each sharing transaction makes it impossible to calculate R_o directly from available information on the main chain B_m. To calculate the score R_o, we utilise function $g_{\mathcal{M}}(\cdot)$, which translates the changeable key $pk_{o,x}$ to its main key pk_o^*. We describe the calculation process in Algorithm 5.1.

We implement function $g_{\mathcal{M}}(\cdot)$ in $\mathbb{S}_{trm}$ contract deployed on $\widehat{b}_{s,z}$, which can only be accessed by NA. To calculate R_o, we need to gather the full history of trust evidence and subsequently construct matrix $\mathbf{T}_o$ (cf. (5.3)). However, as the score R_o can be calculated by a simple recursion (cf. (5.7)), we can update R_o by determining the latest t_{sh}, without re-constructing matrix $\mathbf{T}_o$. In our framework, R_o is updated when a sharing is completed, triggered by transaction tx_{fl} or tx_{rf} (cf (5.13) and (5.14)). $\mathbb{S}_{trm}$ contract keeps a private list of key-value pair $\mathbf{R} = \{\langle pk_1^*, R_1, \rangle, \langle pk_2^*, R_2 \rangle, \ldots\}$, using which tx_{lk} can conveniently look up the score R_o for pk_o^* when required.

5.3.3 Resource Sharing

Our resource sharing framework is comprised of three major steps, depicted in Figure 5.5. First, we list the resources to share (i.e., *listing*). Second, we define the tasks to offload (i.e., *offloading*). Third, we finalise the sharing by consolidating the payments (i.e., *finalisation*). We summarise all transactions (tx) in our resource sharing framework in Table 5.2.

Listing: Initially, resource owners list their computation resource on the broker

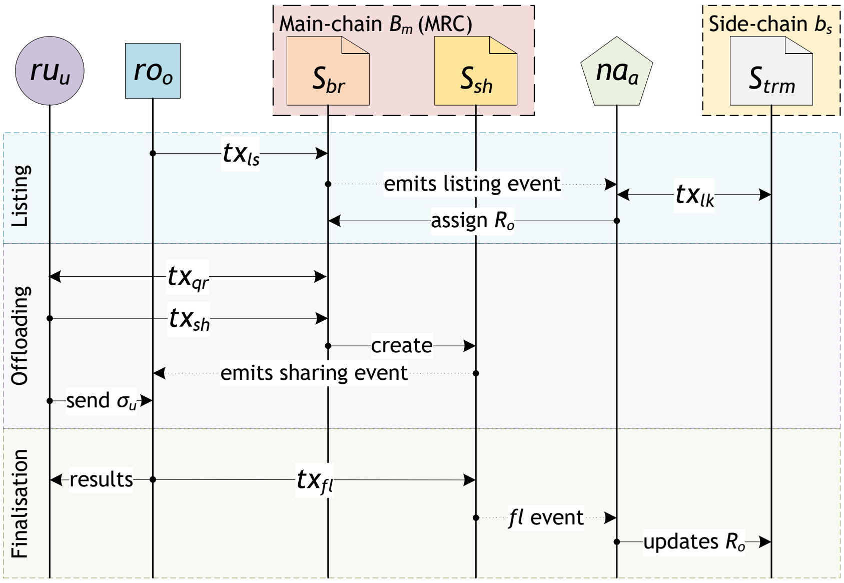

Figure 5.5: The sequence diagram for the proposed resource sharing framework, which is comprised of three steps. The authority na_a acts as the oracle for B_m-$\widehat{b}_s$ communication. Solid and dashed lines represent blockchain transactions and events, respectively.

contract to indicate that the resource is available and can be shared with others. Suppose ro_o is about to list $cr_{o,c}$ to $\mathbb{S}_{br}$ contract on B_m. We require ro_o to invoke listing transaction tx_{ls} using one of its keys $\langle pk_{o,x}, sk_{o,x}\rangle \in KP_o$. We define tx_{ls} as follows:

$$tx_{ls} = [pk_{o,x} \parallel cr_{o,c} \parallel \texttt{prf} \parallel \texttt{tstmp} \parallel \texttt{sig}], \tag{5.9}$$

where $\texttt{prf}$ is the proof of key registration (i.e., the signature of na_a) and $\texttt{sig}$ corresponds to the signature on $\texttt{hash}(pk_{o,x} \parallel cr_{o,c} \parallel \texttt{prf} \parallel \texttt{tstmp})$ using signing key $sk_{o,x}$, used for authentication.

Subsequently, $\mathbb{S}_{br}$ validates tx_{ls}, for instance, by checking if $cr_{o,c}$ has been previously listed and if $pk_{o,x}$ has not been revoked. $\mathbb{S}_{br}$ also initiates a cross-chain transaction tx_{lk} to $\mathbb{S}_{trm}$ contract on $\widehat{b}_{s,z}$ to obtain the score R_o for key $pk_{o,x}$. Essentially, $\mathbb{S}_{trm}$ calls function $g_{\mathcal{M}}(pk_{o,x})$ to get ro_o's main key pk^*_o, which is then used to look up the score R_o. Upon completion, resource $cr_{o,c}$ is listed on $\mathbb{S}_{br}$ with the key $pk_{o,x}$ and the reputation score R_o. Here, ro_o receives the receipt of tx_{ls} for confirmation.

On the other hand, to remove $cr_{o,c}$ from the list, ro_o invokes unlisting transaction

Table 5.2: Blockchain transactions vocabulary

Tx	***SC***	***Description***
tx_{ls}	$\mathbb{S}_{br}$	*listing* tx to list resource $cr_{o,c}$
tx_{lk}	$\mathbb{S}_{trm}$	*look up* tx to get the latest R_o score
tx_{ul}	$\mathbb{S}_{br}$	*unlist* tx to remove $cr_{o,c}$ from the list
tx_{qr}	$\mathbb{S}_{br}$	*query* tx to get $\mathbf{cr}_u$ as per σ_u and R_{min}
tx_{sh}	$\mathbb{S}_{br}$	*share* tx to bind sharing agreement
tx_{fl}	$\mathbb{S}_{sh}$	*finalisation* tx to finalise a sharing
tx_{rf}	$\mathbb{S}_{sh}$	*refund* tx to report if a sharing is failed

tx_{ul} to $\mathbb{S}_{br}$ using the same key $pk_{o,x}$ used for listing $cr_{o,c}$ previously. We formally define tx_{ul} as follows:

$$tx_{ul} = [pk_{o,x} \parallel cr_{o,c} \parallel \texttt{prf} \parallel \texttt{tstmp} \parallel \texttt{sig}]. \tag{5.10}$$

Offloading: Resource users can utilise the computation resources listed on the broker contract by offloading a computation task to a particular resource. Suppose ru_u is about to utilise a computation resource listed on $\mathbb{S}_{br}$. We require ru_u to firstly define a task σ_u to be offloaded (cf. Section 5.2.2). Using one of its keys $\langle pk_{u,y}, sk_{u,y}\rangle \in KP_u$, ru_u invokes a query transaction

$$tx_{qr} = [pk_{u,y} \parallel \sigma_u \parallel R_{min} \parallel \texttt{prf} \parallel \texttt{tstmp} \parallel \texttt{sig}], \tag{5.11}$$

to $\mathbb{S}_{br}$ contract, where R_{min} is the minimal reputation score and `sig` is the signature on $\texttt{hash}(pk_{u,y} \parallel \sigma_u \parallel R_{min} \parallel \texttt{prf} \parallel \texttt{tstmp})$ using signing key $sk_{u,y}$, for authentication.

The contract $\mathbb{S}_{br}$ gathers all idle resources with $R_o \geq R_{min}$. As a response, $\mathbb{S}_{br}$ returns a list of available computation resources $\mathbf{cr}_u = \{cr_{u,1}, cr_{u,2}, \ldots\}$, as per the preferred R_{min}, σ_u^λ and σ_u^δ. Note that, here we employ single-price auction, where ro_o has the right to define the access fee $f_{o,c}$, while ru_u has the liberty to choose which resource satisfies its needs. Access is given to ru_u in first-come-first-served basis [103].

Subsequently, ru_u inspects $\mathbf{cr}_u$ to select a suitable resource $cr_{o,c}$. Assuming ru_u has sufficient balance to pay for the fee $f_{o,c}$, ru_u then invokes transaction tx_{sh} to $\mathbb{S}_{br}$ contract, which temporarily holds the fee $f_{o,c}$ and binds the sharing agreement between ru_u and ro_o. Note that, here ro_u and ru_u use their changeable keys $pk_{o,x}$

and $pk_{u,y}$, respectively, thus no direct relation of $ro_o \leftrightarrow ru_u$. We define transaction tx_{sh} as follows:

$$tx_{sh} = [\,pk_{o,x} \parallel pk_{u,y} \parallel cr_{o,c} \parallel \sigma_u \parallel \texttt{tstmp} \parallel \texttt{sig}\,]. \tag{5.12}$$

Transaction tx_{sh} subsequently creates a new sharing contract $\mathbb{S}_{sh}(pk_{o,x}, pk_{u,y})$, which is later used to enforce the sharing agreement.

Finalisation: Recall that ro_o is expected to complete task σ_u within the deadline d_u (cf. (5.1)). Upon completion of the task execution, ro_o invokes transaction tx_{fl} to $\mathbb{S}_{sh}(pk_{o,x}, pk_{u,y})$, which finalises the sharing by notifying ru_u on the completion. We define tx_{fl} as follows:

$$tx_{fl} = [\,\texttt{hash}(\sigma_{rslt,u}) \parallel \texttt{tstmp} \parallel \texttt{sig}\,], \tag{5.13}$$

where $\texttt{hash}(\sigma_{rslt,u})$ is the hash of the computation results $\sigma_{rslt,u}$ to provide integrity and non-repudiability. When ro_o completes σ_u within the deadline d_u, contract $\mathbb{S}_{sh}(pk_{o,x}, pk_{u,y})$ transfers the fee $f_{o,c}$ to $pk_{o,x}$ to consolidate the payments. Otherwise, $\mathbb{S}_{sh}(pk_{o,x}, pk_{u,y})$ returns $f_{o,c}$ to $pk_{u,y}$, when ro_o is late in returning the computation results. Finally, contract $\mathbb{S}_{sh}(pk_{o,x}, pk_{u,y})$ also notifies $\mathbb{S}_{trm}$ to update R_o accordingly, as described in Algorithm 5.1.

In cases where ro_o never completes the computation by invoking tx_{fl}, ru_u could ask for refund to $\mathbb{S}_{sh}(pk_{o,x}, pk_{u,y})$ after the deadline d_u by invoking refund transaction

$$tx_{rf} = [\,\sigma_u \parallel \texttt{tstmp} \parallel \texttt{sig}\,]. \tag{5.14}$$

Contract $\mathbb{S}_{sh}(pk_{o,x}, pk_{u,y})$ examines whether the deadline d_u is passed and there is tx_{fl} recorded. Subsequently, $\mathbb{S}_{sh}(pk_{o,x}, pk_{u,y})$ returns the fee $f_{o,c}$ to $pk_{u,y}$ and notifies $\mathbb{S}_{trm}$ with regard to this negative experience.

5.4 Performance Evaluation

In this section, we evaluate our proposed resource sharing framework to study its feasibility. We first describe the details of our proof-of-concept implementation, which was deployed on lab-scale private Ethereum networks. We then present the experimental results in four parts, namely i) trust model evaluation, ii) blockchain evaluation, iii) trust evolution against baseline and iv) resource utilisation. We summarise the simulation parameters in Table 5.3.

5.4.1 Proof-of-concept

We implemented our framework on Ethereum blockchain, due to its support on smart contract execution with Ether as the native cryptocurrency. While Ethereum is initially developed as a permission-less blockchain, it can also be configured as a permissioned blockchain. We developed our proof-of-concept using two Ethereum networks on a Lenovo ThinkCentre mini (2.9 GHz quad-core Intel Core i5 CPU with 8GB memory). First, to realise the MRC, we constructed a network of docker containers[1], consisting of a network authority (miner), resource owner and resource user, each of which runs Geth[2] v1.10.17 as an Ethereum node. Second, we utilise Ganache as the IIC side-chain $\widehat{b}_s$, operated by the network authority. We developed our smart contracts in Solidity v0.8.0, and deployed the broker and sharing contracts on the MRC, while the TRM contracts is deployed on the IIC. In addition, we developed Python v3.8.10 scripts using web3py[3] v5.29.1 and py-solc[4] v3.2.0 libraries to evaluate the blockchain implementation and resource utilisation.

5.4.2 Experimental Results

Trust Model Evaluation: We study how the reputation scores grow and converge, by conducting an experiment in which we assign different values for $e_{pos} = [0.6, 1]$ and $\alpha = [0.8, 0.9]$. In this experiment, we simulate a reliable ro which always returns satisfactory task execution for $p = 50$ and plot the results in Figure 5.6. The experimental results indicate that assigning higher value to α would produce more gradual growth of R_o. However, regardless of the value of α, R_o converges to a similar value at different growth rate. On the other hand, assigning different values to e_{pos} would cause the score to converge to different values, which confirms the upper boundary of R_o, theoretically described in (5.5). In conclusion, e_{pos} and α

[1] https://www.docker.com/get-started/

[2] https://geth.ethereum.org/

[3] https://github.com/ethereum/web3.py

[4] https://github.com/ethereum/py-solc

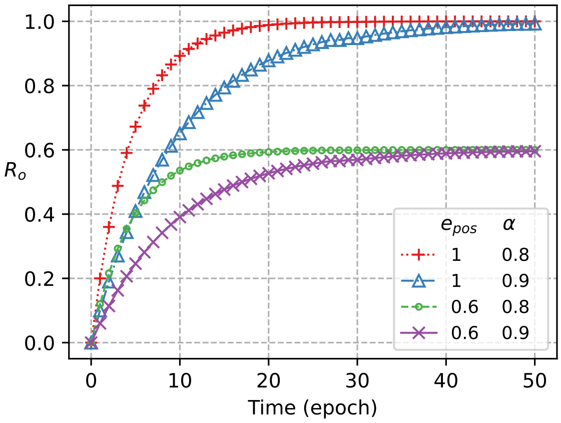

Figure 5.6: The evolution of reputation score R_o with regards to changing parameters. We give different values to α and e_{pos} to see the effect on the growth rate and convergence value.

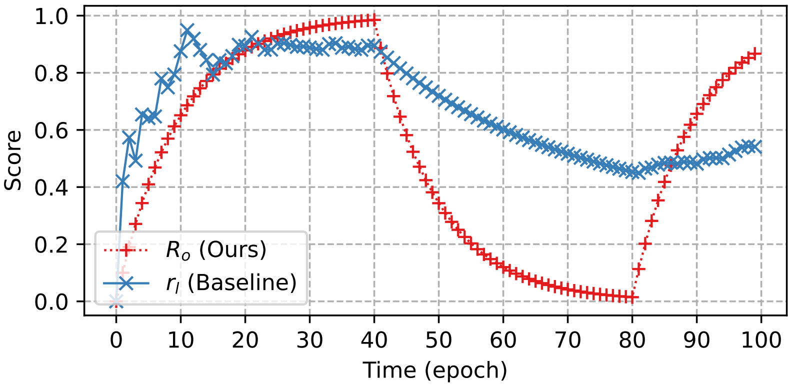

Figure 5.7: The evolution of the reputation score for a resource owner, which becomes unreliable during $40 < \text{time} < 80$. The baseline refers to the work in [76].

can be properly tuned according to the deployment settings, for instance, we can set higher value for α when we need more sharing experiences to achieve highly reputed ro, resulting in more gradual growth of R_o.

Trust Evolution against Baseline: We evaluate our trust model in non-ideal conditions, where the ro becomes unreliable for a certain period of time. We simulate $p = 100$ offloading tasks, in which the ro turns unreliable between $40 < p < 80$. We compare our trust model with a resource sharing framework for IIoT environment as the baseline [76], where the reputation of a node, denoted r_l, is determined by the average of the sharing rewards obtained after completing an offloaded task. We choose $\alpha = 0.9$, $e_{pos} = 1$ and $e_{neg} = 0$ as the simulation parameters; and we use

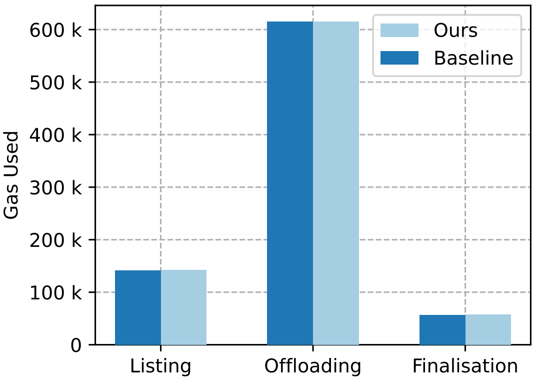

Figure 5.8: The comparison of the gas consumption for the execution of each step in our framework. Here, the baseline does not implement privacy preservation in the TRM.

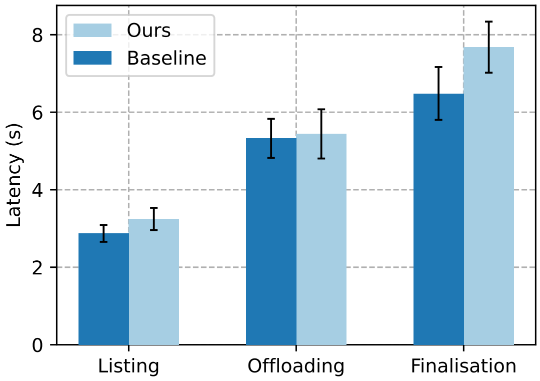

Figure 5.9: The comparison of latency in the execution of each step in our framework. The error bars indicate the standard error. The baseline refers to the same framework as in Fig. 5.8.

default parameters for the baseline [76]. We plot the results in Figure 5.7, where the epoch time indicates the time at which ro finalises an offloaded task.

The results indicate that for the first 40 offloaded tasks, both models show similar trends, where the scores gradually grow and plateau at approximately similar value, although the baseline model suffers from undesired fluctuations. However, when ro turns unreliable, both models exhibit different patterns. Our model degrades the score to the lower boundary, i.e., $R_o = 0$, which indicates that ro is now untrustworthy. On the other hand, the baseline model only degrades the score to approximately $r_l = 0.4$, giving an impression that the node is still fairly trustworthy. Then, when ro subsequently shows reliable behaviour, our model could return R_o score back to its desired maximum value, while r_l score of the baseline lingers at approximately $r_l = 0.6$, despite more tasks have been successfully offloaded.

Blockchain Evaluation: Our framework introduces unavoidable processing overheads as a result of incorporating changeable keys for privacy-preservation in TRM. We study the incurred overheads by comparing our solution against a TRM framework without privacy preservation. Consequently, the trust score can be directly calculated from the interaction history on the main chain, removing the need of a separate side-chain and some additional trust computations. In this experiment, we specifically examine the gas consumption and the latency for executing three steps of our framework (cf. Section 5.3.3); and plot the results in Figure 5.8

and 5.9. Relative to the baseline, the results indicate that our framework only introduces negligible additional gas consumption, while our framework requires slightly more time to complete the process. However, we notice that this increase in processing is insignificant, especially with regards to the added benefits of incorporating privacy-preservation. In addition, while using many key pairs incurs some costs, e.g., the deposits (cf. (5.8)), there is a balance which provides anonymity without significant key costs [104].

Utilisation: We study the implications of the deployment of our framework with regards to resource utilisation rate by conducting an experiment, wherein we vary the proportion of reliable ro in the network. In this experiment, we compare our framework against a baseline, which presents a network condition where TRM is not incorporated. We run the simulation using parameters in Table 5.3 and plot the result in Figure 5.10, where we also plot the convergence time at which maximum resource utilisation is reached. While our framework requires longer convergence, i.e., > 80 epoch, the results show that our framework can increase the resource utilisation rate relative to the baseline, especially when the number of reliable ro increases. In the baseline, rus are only willing to offload the tasks to known or trusted ro, which constitutes only 10% of total ro (cf. Table 5.3). There are, however, some ru that are willing to take the risk to offload the tasks to unknown ro, which increases the utilisation rate in the baseline, i.e., $> 10\%$. On the other hand, when a TRM is incorporated in the network, it introduces the reputation score R_o for each ro, which helps ru select the offloading target ro, based on its R_o scores. As there are more satisfactory task completions, there are more ro with high R_o scores, which consequently raises the trust in ro and increases the overall resource utilisation.

5.5 Chapter Summary

In this chapter, we proposed a privacy-preserving TRM framework for resource sharing in 6G-enabled IoT. In our framework, each participating node uses different keys in each sharing transaction, which conceals their actual pseudonyms, reducing the risk of de-anonymisation attacks. We design two interconnected blockchains, namely the IIC and MRC, where we store the trust scores and record the resource sharing

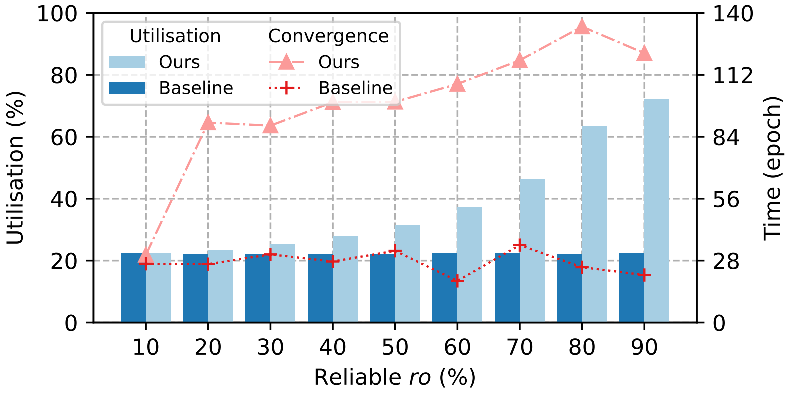

Figure 5.10: The comparison of total resource utilisation rate between our solution and the baseline, which refers to a network without a TRM implemented.

Table 5.3: Simulation parameters

Parameters	***Value***
α	0.9
e_{pos}	1
e_{neg}	0
Number of ro	500
Number of ru	600
Node type	[reliable, unreliable, faulty]
Trusted ro	10%
Lenient ru	15%
Iteration	100

transactions, respectively. The experimental results indicated that our framework only incurs insignificant overheads, signifying its feasibility.

Chapter 6

Conclusion and Future Work

This chapter summarises the overall contributions of the thesis in Section 6.1 and discusses the potential research directions for each of the contributions in Section 6.2.

6.1 Concluding Remarks

The main contribution of the thesis is the adoption of blockchain-based TRM to solve the trustworthiness issues in the core functionalities of IoT ecosystems, namely access control, Collaborative Intrusion Detection System (CIDS) and resource sharing, which are summarised below:

- Chapter 3 presented a blockchain-based TRM framework to address the issues in the enforcement of static predefined access control policies in IoT authorisation. The proposed framework exploits TRM to supply additional attributes to an Attribute Based Access Control mechanism. As such, the framework realises dynamic access control policies without overlooking the fact that access control requires a sensitive consideration of who can access a resource. The framework progressively quantifies the trust and reputation scores of each node in the network and incorporates the scores into the access control mechanism to achieve dynamic and flexible access control. The framework is designed to be blockchain agnostic, which can be implemented in any blockchain platforms that have adequate support for smart contract execution. The chapter presents the proof-of-concept implementation of the proposed framework in a public Rinkeby Ethereum test-network interconnected with a lab-scale testbed of Raspberry Pi computers. Experimental results show that our proposed framework achieves consistent processing latencies and is feasible for implementing effective access control in decentralised IoT networks.

- Chapter 4 proposed a decentralised CIDS framework that aims to build trust between CIDS nodes, with the help of a blockchain-based TRM. The proposed TRM framework evaluates and quantifies the quality of the contributed detection rules into trust scores, from which the trustworthiness of CIDS nodes are derived. A peer-to-peer decentralised storage is exploited to maintain a copy of the shared trustworthy detection rules, thus ensuring scalability. The framework divides the participating nodes into three categories, namely validator, contributor and regular nodes, each of which has a different role in the system. Two smart contracts are introduced, namely Trust and Reputation Management (TRM) and Storage smart contract, to quantify each node's trustworthiness and manage the decentralised storage, respectively. The framework offloads trust computation to the TRM smart contract which reduces the computation load for each CIDS node. A smart contract-based consensus algorithm is proposed to achieve collaborative detection rule validation and avoid an adversary from contributing deceptive detection rules. The chapter uses signature-based CIDS as an illustrative example. However, the TRM concept can be generalised to other types of CIDS. In addition, the proposed framework is blockchain agnostic and could be implemented on any blockchain platform that supports smart contracts. The proposed framework is evaluated on a lab-scale testbed of private Ethereum network. The experimental results indicate the feasibility of the proposed concept and show that the performance falls within the expected benchmarks of the Ethereum platform.

- Chapter 5 presented a TRM framework for resource sharing in 6G-enabled IoT to provide assurance in the security and trustworthiness of the underlying resource sharing process. The resource sharing mechanism follows task offloading approach in edge computing, where the resource users offload computation tasks to resource owners, after which the trustworthiness of the resource owners are determined. The framework allows for both resource owners and users to employ changeable keys to obfuscate their transaction traces in the network. To realise privacy preservation, the framework utilises two interconnected blockchains. First, an IIC is used to store the mapping between changeable keys and the main pseudonyms of a node. Second, a Main Resource sharing Chain (MRC) is utilised to record the resource sharing transactions between the resource owners and users. The framework exploits smart contracts to provide auditable trust calculation, where a recursive trust computation is proposed to meet the efficiency requirements of trust calculation in 6G networks. The framework achieves rater and ratee anonymity, where the identities of both resource users (rater) and owners (ratee) are completely concealed. The experimental results from a proof-of-concept implementation indicated that our framework only incurs insignificant overheads, signifying its feasibility.

6.2 Future Research Directions

Research in providing trusted and secure collaboration for IoT networks is an active research area, which poses various challenging open questions for further investigation. The potential research directions for each technical chapter in the present thesis are discussed in the following subsections.

6.2.1 Dynamic IoT Access Control

The proposed dynamic access control for IoT operates by incorporating trust and reputation scores as additional attributes in an ABAC scheme. Naturally, the trust and reputation scores grow over time, as more interactions occur in the network. In practice, one may encounter a bootstrapping problem, in which a new node with zero reputation score cannot participate in the network, as typically access policies require a node to have at least some level of trust and reputation scores. However, we can presume that some service providers would allow access to their resources for some nodes with zero scores, through which a newly joined node may initially build up its reputation score. Future research can be aimed to circumvent this boot strapping issue, for instance, by designing an endorsement-based access policy, in which new service consumers with zero reputation scores may request for an access to a resource with a prior endorsement from another service provider with an adequate reputation score.

In addition, while the proposed TRM captures attacks and violations against access control, violations in the attribute registration process, occurring on the sidechains cannot be mitigated. As an implication, malicious service consumers, which initially attempt to register themselves using invalid attributes, would obtain the same initial trust scores as honest service consumers, when they join the network for the first time. Future research should improve this single-dimensional trust model by developing a more robust multi-dimensional model with high efficiency. In the improved model, multiple metrics can be taken into account, including violations in the attribute registration process and intrusion related information from the deployed intrusion detection systems. The study should also investigate the performance under different malicious scenarios and compare the result with the previous work.

6.2.2 Trustworthy CIDS

The proposed trustworthy CIDS framework utilises signature-based IDS as an illustrative example, where the IDS compares network traffic against known attack signatures in the detection database. Future research may explore the possibility of incorporating the framework in the anomaly-based CIDS, where the IDS nodes apply machine learning algorithms to the network traffic for examining if certain attacks occur as per the detection model. In this type of CIDS, the collaboration follows the federated learning approach, where each node contributes a trained machine learning model to a blockchain aggregator, who validates the contributed model to determine the quality of the contributed models and assign trust and reputation scores to the node.

Another further direction for future research is the incorporation of economic model and game theory behind the fees and incentives mechanisms. As sharing detection rules or models may disclose some sensitive information, some CIDS nodes may be reluctant in sharing their knowledge of recent attacks. To encourage sharing, the CIDS framework may incorporate incentives mechanisms, where contributors are awarded certain amount of cryptocurrency as the incentives. The underlying TRM would play an important role in determining the amount of incentives to be awarded and to see if the contributed attack signatures or models are of poor quality, to which the TRM should apply punishments. The game theoretic approach would be instrumental in determining various aspects of financial motivations in the incentive mechanisms.

6.2.3 Resource Sharing for 6G-enabled IoT

The proposed TRM framework for resource sharing employs single-price auction, where resource owners have the right to define the fee to consume the resources, while resource users have the liberty to choose which resource to offload the computation tasks. Future research may study the effect on employing double auction mechanism, where both resource owners and users have the right to define their own asking price. Smart contracts would be critical in consolidating the requests from both sides. In addition, the results of implementing double auction mechanism can also be observed from the typical performance evaluation for blockchain systems, which investigate the blockchain performance in terms of overall latency, throughput and other overheads.

In addition, future research can be aimed to study the effective optimisation model to maximise the utilisation rate of the scare resources. An objective function with certain constraints needs to be properly defined to discover the optimum solution to allocate the computation tasks, which is typically an NP-hard problem.

There are also the needs for extensive research to address the challenges in the scalability, interoperability, and the security and privacy of blockchain-based TRM for 6G. Future research should provide rigorous measures for security and privacy preservation entailing the TRM model and blockchain design. Unprecedented scale in 6G network would enlarge the attack surface which may render the conventional defence strategy for TRM attacks infeasible. In addition, integration of autonomous networks in 6G highlights the need for reliable interoperability between networks. More research should be undertaken to investigate how a reliable and secure reputation score transfer between independent TRMs can be realised, which requires rigorous assessment of distinctive trust metrics in different autonomous networks.

Bibliography

[1] S. Bansal and D. Kumar, “IoT Ecosystem: A Survey on Devices, Gateways, Operating Systems, Middleware and Communication”, *International Journal of Wireless Information Networks*, vol. 27, no. 3, pp. 340–364, Sep. 2020, ISSN: 1068-9605, 1572-8129. DOI: 10.1007/s10776-020-00483-7.

[2] I. Butun, P. Österberg, and H. Song, “Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures”, *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 616–644, 2020, ISSN: 1553-877X. DOI: 10.1109/COMST.2019.2953364.

[3] I. Bisio, A. Fedeli, C. Garibotto, F. Lavagetto, M. Pastorino, and A. Randazzo, “Two Ways for Early Detection of a Stroke Through a Wearable Smart Helmet: Signal Processing vs. Electromagnetism”, *IEEE Wireless Communications*, vol. 28, no. 3, pp. 22–27, Jun. 2021, ISSN: 1536-1284, 1558-0687. DOI: 10.1109/MWC.001.2000401.

[4] M. Saleem, S. Abbas, T. M. Ghazal, M. Adnan Khan, N. Sahawneh, and M. Ahmad, “Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques”, *Egyptian Informatics Journal*, vol. 23, no. 3, pp. 417–426, Sep. 2022, ISSN: 1110-8665. DOI: 10.1016/j.eij.2022.03.003.

[5] W. Saad, M. Bennis, and M. Chen, “A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems”, *IEEE Network*, vol. 34, no. 3, pp. 134–142, May 2020, ISSN: 1558-156X. DOI: 10.1109/MNET.001.1900287.

[6] G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, *Towards Blockchain-based Trust and Reputation Management for Trustworthy 6G Networks*, Aug. 2022. DOI: 10.48550/arXiv.2208.07562. arXiv: 2208.07562 [cs].

[7] K. L. Lueth. (2020). State of the iot 2020: 12 billion iot connections, surpassing non-iot for the first time, [Online]. Available: https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/ (visited on 09/04/2022).

[8] M. Hasan. (2022). State of iot 2022: Number of connected iot devices growing 18% to 14.4 billion globally, [Online]. Available: https://iot-analytics.com/number-connected-iot-devices/ (visited on 09/04/2022).

[9] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai Botnet", in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1093–1110, ISBN: 978-1-931971-40-9.

[10] I. Ahmad, K.-L. A. Yau, M. H. Ling, and S. L. Keoh, "Trust and Reputation Management for Securing Collaboration in 5G Access Networks: The Road Ahead", *IEEE Access*, vol. 8, pp. 62 542–62 560, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2984318.

[11] S. M. Kerner. (2022). Colonial pipeline hack explained: Everything you need to know, [Online]. Available: https://www.techtarget.com/whatis/feature/Colonial-Pipeline-hack-explained-Everything-you-need-to-know/ (visited on 09/04/2022).

[12] M. binti Mohamad Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey", *Computer Networks*, vol. 148, pp. 283–294, 2019, ISSN: 13891286. DOI: 10.1016/j.comnet.2018.11.025.

[13] O. Hasan, L. Brunie, and E. Bertino, "Privacy-Preserving Reputation Systems Based on Blockchain and Other Cryptographic Building Blocks: A Survey", *ACM Computing Surveys*, vol. 55, no. 2, 32:1–32:37, Jan. 2022, ISSN: 0360-0300. DOI: 10.1145/3490236.

[14] A. Sharma, E. S. Pilli, A. P. Mazumdar, and P. Gera, "Towards trustworthy Internet of Things: A survey on Trust Management applications and schemes", *Computer Communications*, vol. 160, no. June, pp. 475–493, 2020, ISSN: 1873703X. DOI: 10.1016/j.comcom.2020.06.030. [Online]. Available: https://doi.org/10.1016/j.comcom.2020.06.030.

[15] Z. Lu, Q. Wang, G. Qu, and Z. Liu, "BARS: A Blockchain-Based Anonymous Reputation System for Trust Management in VANETs", *Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018*, pp. 98–103, 2018, ISSN: 2324-9013. DOI: 10.1109/TrustCom/BigDataSE.2018.00025. arXiv: 1807.06159.

[16] E. Bellini, Y. Iraqi, and E. Damiani, "Blockchain-Based Distributed Trust and Reputation Management Systems: A Survey", *IEEE Access*, vol. 8, pp. 21 127–21 151, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2969820.

[17] M. Swan, *Blockchain: Blueprint for a new economy.* " O'Reilly Media, Inc.", 2015.

[18] R. S. Sandhu and P. Samarati, “Access control: Principle and practice”, *IEEE Communications Magazine*, vol. 32, no. 9, pp. 40–48, 1994.

[19] A. Ouaddah, H. Mousannif, A. Abou Elkalam, and A. Ait Ouahman, “Access control in the Internet of Things: Big challenges and new opportunities”, *Computer Networks*, vol. 112, pp. 237–262, 2017, ISSN: 13891286. DOI: 10.1016/j.comnet.2016.11.007. [Online]. Available: https://doi.org/10.1016/j.comnet.2016.11.007.

[20] O. Novo, “Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT”, *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, Apr. 2018, ISSN: 2327-4662. DOI: 10.1109/JIOT.2018.2812239.

[21] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, “A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT”, *IEEE Access*, vol. 7, pp. 38 431–38 441, 2019, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2905846. [Online]. Available: https://ieeexplore.ieee.org/document/8668769/.

[22] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, “Smart contract-based access control for the internet of things”, *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2019. DOI: 10.1109/JIOT.2018.2847705.

[23] E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, “A Supervised Intrusion Detection System for Smart Home IoT Devices”, *IEEE IoT Journal*, vol. 6, no. 5, pp. 9042–9053, 2019, ISSN: 23274662. DOI: 10.1109/JIOT.2019.2926365.

[24] C. J. Fung, J. Zhang, I. Aib, and R. Boutaba, “Dirichlet-based trust management for effective collaborative intrusion detection networks”, *IEEE TNSM*, vol. 8, no. 2, pp. 79–91, 2011, ISSN: 19324537. DOI: 10.1109/TNSM.2011.050311.100028.

[25] W. Li, S. Tug, W. Meng, and Y. Wang, “Designing collaborative blockchained signature-based intrusion detection in IoT environments”, *Future Generation Computer Systems*, vol. 96, 2019, ISSN: 0167739X. DOI: 10.1016/j.future.2019.02.064.

[26] J. Liang and M. Ma, “Co-maintained database based on blockchain for idss: A lifetime learning framework”, *IEEE TNSM*, vol. 18, no. 2, pp. 1629–1645, 2021. DOI: 10.1109/TNSM.2021.3064607.

[27] T. Golomb, Y. Mirsky, and Y. Elovici, “CIoTA: Collaborative IoT Anomaly Detection via Blockchain”, in *Workshop on Decentralized IoT Security and Standards (DISS) 2018*, 2018, ISBN: 1891562517. DOI: 10.14722/diss.2018.23003. arXiv: 1803.03807.

[28] C. Yenugunti and S. S. Yau, "A Blockchain Approach to Identifying Compromised Nodes in Collaborative Intrusion Detection Systems", in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, Aug. 2020, pp. 87–93. DOI: `10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00029`.

[29] G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Towards scalable and trustworthy decentralized collaborative intrusion detection system for iot", in *5th IEEE/ACM IoTDI*, 2020, pp. 256–257. DOI: `10.1109/IoTDI49375.2020.00035`.

[30] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. M. Leung, "Enabling Massive IoT Toward 6G: A Comprehensive Survey", *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 11 891–11 915, Aug. 2021, ISSN: 2327-4662. DOI: `10.1109/JIOT.2021.3063686`.

[31] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, and H. V. Poor, "6G Internet of Things: A Comprehensive Survey", *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, Jan. 2022, ISSN: 2327-4662. DOI: `10.1109/JIOT.2021.3103320`.

[32] S. Hu, Y.-C. Liang, Z. Xiong, and D. Niyato, "Blockchain and Artificial Intelligence for Dynamic Resource Sharing in 6G and Beyond", *IEEE Wireless Communications*, vol. 28, no. 4, pp. 145–151, Aug. 2021, ISSN: 1558-0687. DOI: `10.1109/MWC.001.2000409`.

[33] J. Wang, X. Ling, Y. Le, Y. Huang, and X. You, "Blockchain-enabled wireless communications: A new paradigm towards 6G", *National Science Review*, vol. 8, no. 9, nwab069, Sep. 2021, ISSN: 2095-5138, 2053-714X. DOI: `10.1093/nsr/nwab069`.

[34] A. Biryukov and S. Tikhomirov, "Deanonymization and Linkability of Cryptocurrency Transactions Based on Network Analysis", in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, Jun. 2019, pp. 172–184. DOI: `10.1109/EuroSP.2019.00022`.

[35] C. Kang, C. Lee, K. Ko, J. Woo, and J. W.-K. Hong, "De-Anonymization of the Bitcoin Network Using Address Clustering", in *Blockchain and Trustworthy Systems*, Z. Zheng, H.-N. Dai, X. Fu, and B. Chen, Eds., ser. Communications in Computer and Information Science, Singapore: Springer, 2020, pp. 489–501. DOI: `10.1007/978-981-15-9213-3_38`.

[36] A. Dorri, C. Roulin, S. Pal, S. Baalbaki, R. Jurdak, and S. S. Kanhere, "Device identification in blockchain-based internet of things", *IEEE Internet of Things Journal*, pp. 1–1, 2022. DOI: `10.1109/JIOT.2022.3194671`.

[37] R. Di Pietro, X. Salleras, M. Signorini, and E. Waisbard, "A Blockchain-based Trust System for the Internet of Things", in *Proceedings of the 23Nd ACM on Symposium on Access Control Models and Technologies*, ser. SACMAT '18, ACM, 2018, pp. 77–83, ISBN: 978-1-4503-5666-4. DOI: `10.1145/3205977.3205993`. [Online]. Available: `http://doi.acm.org/10.1145/3205977.3205993`.

[38] S. Malik, N. Gupta, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "TradeChain: Decoupling Traceability and Identity inBlockchain enabled Supply Chains", *arXiv:2105.11217 [cs]*, May 2021. arXiv: `2105.11217 [cs]`.

[39] A. Batwa and A. Norrman, "Blockchain technology and trust in supply chain management: A literature review and research agenda", *Operations and Supply Chain Management: An International Journal*, vol. 14, no. 2, pp. 203–220, 2021. DOI: `http://doi.org/10.31387/oscm0450297`.

[40] D. Gambetta *et al.*, "Can we trust trust", *Trust: Making and breaking cooperative relations*, vol. 13, pp. 213–237, 2000.

[41] S. Malik, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Trustchain: Trust management in blockchain and iot supported supply chains", in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 184–193. DOI: `10.1109/Blockchain.2019.00032`.

[42] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services", *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 429–445, 2019. DOI: `10.1109/TSC.2018.2823705`.

[43] Q. Li, A. Malip, K. M. Martin, S.-L. Ng, and J. Zhang, "A Reputation-Based Announcement Scheme for VANETs", *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 4095–4108, Nov. 2012, ISSN: 1939-9359. DOI: `10.1109/TVT.2012.2209903`.

[44] G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Blockchain for Trust and Reputation Management in Cyber-Physical Systems", in *Handbook on Blockchain*, ser. Springer Optimization and Its Applications, D. A. Tran, M. T. Thai, and B. Krishnamachari, Eds., Cham: Springer International Publishing, 2022, pp. 339–362, ISBN: 978-3-031-07535-3. DOI: `10.1007/978-3-031-07535-3_10`.

[45] A. Javaid, M. Zahid, I. Ali, R. J. U. H. Khan, Z. Noshad, and N. Javaid, "Reputation system for iot data monetization using blockchain", in *Advances on Broad-Band Wireless Computing, Communication and Applications*, L. Barolli, P. Hellinckx, and T. Enokido, Eds., Cham: Springer International Publishing, 2020, pp. 173–184, ISBN: 978-3-030-33506-9.

[46] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision", *Decision support systems*, vol. 43, no. 2, pp. 618–644, 2007.

[47] O. Hasan, L. Brunie, and E. Bertino, "Privacy Preserving Reputation Systems based on Blockchain and other Cryptographic Building Blocks: A Survey", *University of Lyon Research Report*, pp. 1–65, 2020.

[48] R. Levien and A. Aiken, "Attack-resistant trust metrics for public key certification.", in *Usenix security symposium*, 1998, pp. 229–242.

[49] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks", in *Proceedings of the 12th international conference on World Wide Web*, 2003, pp. 640–651.

[50] A. Whitby, A. Jøsang, and J. Indulska, "Filtering out unfair ratings in bayesian reputation systems", in *Proc. 7th Int. Workshop on Trust in Agent Societies*, Citeseer, vol. 6, 2004, pp. 106–117.

[51] R. C. Merkle, "A digital signature based on a conventional encryption function", in *Conference on the theory and application of cryptographic techniques*, Springer, 1987, pp. 369–378.

[52] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain", in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2017, pp. 2567–2572. DOI: 10.1109/SMC.2017.8123011.

[53] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger", *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[54] Ethereum. (2022). The great renaming: What happened to eth2?, [Online]. Available: https://blog.ethereum.org/2022/01/24/the-great-eth2-renaming (visited on 09/04/2022).

[55] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains", in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18, Porto, Portugal: Association for Computing Machinery, 2018, ISBN: 9781450355841. DOI: 10.1145/3190508.3190538. [Online]. Available: https://doi.org/10.1145/3190508.3190538.

[56] M. P. Andersen, S. Kumar, M. AbdelBaky, G. Fierro, J. Kolb, H.-S. Kim, D. E. Culler, and R. A. Popa, "WAVE: A decentralized authorization framework with transitive delegation", in *USENIX Security*, 2019.

[57] O. J. A. Pinno, A. R. A. Gregio, and L. C. De Bona, “ControlChain: Blockchain as a Central Enabler for Access Control Authorizations in the IoT”, *2017 IEEE GLOBECOM*, pp. 1–6, 2018. DOI: 10.1109/GLOCOM.2017.8254521.

[58] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “Lsb: A lightweight scalable blockchain for iot security and anonymity”, *Journal of Parallel and Distributed Computing*, vol. 134, pp. 180–197, 2019.

[59] M. P. Andersen, J. Kolb, K. Chen, G. Fierro, D. E. Culler, and R. A. Popa, *WAVE: A Decentralized Authorization System for IoT via Blockchain Smart Contracts*, 2017. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-234.html.

[60] S. Pal, T. Rabehaja, A. Hill, M. Hitchens, and V. Varadharajan, “On the integration of blockchain to the internet of things for enabling access right delegation”, *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2630–2639, 2020.

[61] Y. Zhang and X. Wu, “Access Control in Internet of Things: A Survey”, *DEStech Transactions on Engineering and Technology Research*, no. apetc, 2018. DOI: 10.12783/dtetr/apetc2017/11295. eprint: arXiv:1610.01065v1.

[62] I. R. Chen, F. Bao, and J. Guo, “Trust-Based Service Management for Social Internet of Things Systems”, *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 6, pp. 684–696, 2016, ISSN: 19410018. DOI: 10.1109/TDSC.2015.2420552.

[63] I. R. Chen, J. Guo, and F. Bao, “Trust Management for SOA-Based IoT and Its Application to Service Composition”, *IEEE Transactions on Services Computing*, vol. 9, no. 3, pp. 482–495, 2016, ISSN: 19391374. DOI: 10.1109/TSC.2014.2365797.

[64] J. Bernal Bernabe, J. L. Hernandez Ramos, and A. F. Skarmeta Gomez, “TACIoT: Multidimensional trust-aware access control system for the Internet of Things”, *Soft Computing*, vol. 20, no. 5, pp. 1763–1779, May 2016, ISSN: 1433-7479. DOI: 10.1007/s00500-015-1705-6.

[65] B. Gwak, J. H. Cho, D. Lee, and H. Son, “TARAS: Trust-Aware Role-Based Access Control System in Public Internet-of-Things”, in *Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018*, IEEE, 2018, pp. 74–85, ISBN: 9781538643877. DOI: 10.1109/TrustCom/BigDataSE.2018.00022.

[66] G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Trust management in decentralized iot access control system", in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2020, pp. 1–9. DOI: `10.1109/ICBC48266.2020.9169481`.

[67] I. R. Chen, J. Guo, D. C. Wang, J. J. Tsai, H. Al-Hamadi, and I. You, "Trust-Based Service Management for Mobile Cloud IoT Systems", *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 246–263, 2019, ISSN: 19324537. DOI: `10.1109/TNSM.2018.2886379`.

[68] H. Al-Hamadi, I. R. Chen, and J. H. Cho, "Trust Management of Smart Service Communities", *IEEE Access*, vol. 7, pp. 26 362–26 378, 2019, ISSN: 21693536. DOI: `10.1109/ACCESS.2019.2901023`.

[69] D. E. Kouicem, Y. Imine, A. Bouabdallah, and H. Lakhlef, "A decentralized blockchain-based trust management protocol for the internet of things", *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2020. DOI: `10.1109/TDSC.2020.3003232`.

[70] W. B. Daoud, M. S. Obaidat, A. Meddeb-Makhlouf, F. Zarai, and K.-F. Hsiao, "TACRM: Trust access control and resource management mechanism in fog computing", *Human-centric Computing and Information Sciences*, vol. 9, no. 1, p. 28, Jul. 2019, ISSN: 2192-1962. DOI: `10.1186/s13673-019-0188-3`.

[71] V. Dedeoglu, R. Jurdak, G. D. Putra, A. Dorri, and S. S. Kanhere, "A trust architecture for blockchain in iot", in *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, ser. MobiQuitous '19, Houston, Texas, USA: Association for Computing Machinery, 2019, pp. 190–199, ISBN: 9781450372831. DOI: `10.1145/3360774.3360822`. [Online]. Available: `https://doi.org/10.1145/3360774.3360822`.

[72] N. Kolokotronis, S. Brotsis, G. Germanos, C. Vassilakis, and S. Shiaeles, "On blockchain architectures for trust-based collaborative intrusion detection", *IEEE SERVICES 2019*, pp. 21–28, 2019. DOI: `10.1109/SERVICES.2019.00019`.

[73] W. Li, W. Meng, J. Parra-Arnau, and K.-K. R. Choo, "Enhancing Challenge-based Collaborative Intrusion Detection Against Insider Attacks using Spatial Correlation", in *2021 IEEE Conference on Dependable and Secure Computing (DSC)*, Jan. 2021, pp. 1–8. DOI: `10.1109/DSC49826.2021.9346232`.

[74] W. Fan, Y. Park, S. Kumar, P. Ganta, X. Zhou, and S.-y. Chang, "Blockchain-enabled Collaborative Intrusion Detection in Software Defined Networks", in *19th IEEE TrustCom*, 2020, pp. 967–974. DOI: `10.1109/TrustCom50675.2020.00129`.

[75] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "A Deep Blockchain Framework-enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks", *IEEE IoT-J*, vol. 4662, 2020. DOI: `10.1109/jiot.2020.2996590`.

[76] S. Iqbal, R. M. Noor, A. W. Malik, and A. U. Rahman, "Blockchain-Enabled Adaptive-Learning-Based Resource-Sharing Framework for IIoT Environment", *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14 746–14 755, Oct. 2021, ISSN: 2327-4662. DOI: `10.1109/JIOT.2021.3071562`.

[77] B. Bellaj, A. Ouaddah, E. Bertin, N. Crespi, A. Mezrioui, and K. Bellaj, "BTrust: A New Blockchain-Based Trust Management Protocol for Resource Sharing", *Journal of Network and Systems Management*, vol. 30, no. 4, p. 64, Oct. 2022, ISSN: 1064-7570, 1573-7705. DOI: `10.1007/s10922-022-09674-4`.

[78] H. Chai, S. Leng, K. Zhang, and S. Mao, "Proof-of-Reputation Based-Consortium Blockchain for Trust Resource Sharing in Internet of Vehicles", *IEEE Access*, vol. 7, pp. 175 744–175 757, 2019, ISSN: 2169-3536. DOI: `10.1109/ACCESS.2019.2956955`.

[79] L. Xiao, Y. Ding, D. Jiang, J. Huang, D. Wang, J. Li, and H. Vincent Poor, "A Reinforcement Learning and Blockchain-Based Trust Mechanism for Edge Networks", *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5460–5470, Sep. 2020, ISSN: 1558-0857. DOI: `10.1109/TCOMM.2020.2995371`.

[80] H. J. Jo and W. Choi, "BPRF: Blockchain-based privacy-preserving reputation framework for participatory sensing systems", *PLOS ONE*, vol. 14, no. 12, e0225688, Dec. 2019, ISSN: 1932-6203. DOI: `10.1371/journal.pone.0225688`.

[81] A. Schaub, R. Bazin, O. Hasan, and L. Brunie, "A Trustless Privacy-Preserving Reputation System", in *ICT Systems Security and Privacy Protection*, J.-H. Hoepman and S. Katzenbeisser, Eds., vol. 471, Cham: Springer International Publishing, 2016, pp. 398–411. DOI: `10.1007/978-3-319-33630-5_27`.

[82] J. Ye, X. Kang, Y.-C. Liang, and S. Sun, "A Trust-Centric Privacy-Preserving Blockchain for Dynamic Spectrum Management in IoT Networks", *IEEE Internet of Things Journal*, pp. 1–1, 2022, ISSN: 2327-4662. DOI: `10.1109/JIOT.2022.3142989`.

[83] Z. Lu, W. Liu, Q. Wang, G. Qu, and Z. Liu, "A Privacy-Preserving Trust Model Based on Blockchain for VANETs", *IEEE Access*, vol. 6, pp. 45 655–45 664, 2018, ISSN: 2169-3536. DOI: `10.1109/ACCESS.2018.2864189`.

[84] TechCrunch, *Elk, a blockchain dev board for decentralized IoT, launches on Kickstarter*, `https://techcrunch.com/2019/07/25/elk-a-blockchain-dev-board-for-decentralized-iot-launches-on-kickstarter/`, [Online; accessed 26-July-2019], 2019.

[85] V. A. Siris, P. Nikander, S. Voulgaris, N. Fotiou, D. Lagutin, and G. C. Polyzos, "Interledger Approaches", *IEEE Access*, vol. 7, pp. 89 948–89 966, 2019, ISSN: 21693536. DOI: `10.1109/ACCESS.2019.2926880`.

[86] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain technologies for the internet of things: Research issues and challenges", *IEEE IoT Journal*, vol. 6, no. 2, pp. 2188–2204, 2019.

[87] K. L. Huang, S. S. Kanhere, and W. Hu, "On the need for a reputation system in mobile phone based sensing", *Ad Hoc Networks*, vol. 12, pp. 130–149, 2014, ISSN: 1570-8705. DOI: `https://doi.org/10.1016/j.adhoc.2011.12.002`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S1570870511002174`.

[88] M. Castro, B. Liskov, *et al.*, "Practical byzantine fault tolerance", in *OSDI*, vol. 99, 1999, pp. 173–186.

[89] Solidity, *Solidity - Solidity 0.6.6 documentation*, `https://solidity.readthedocs.io/en/v0.6.6/`, [Online; accessed 4-June-2020], 2020.

[90] S. Pal, T. Rabehaja, M. Hitchens, V. Varadharajan, and A. Hill, "On the Design of a Flexible Delegation Model for the Internet of Things Using Blockchain", *IEEE Transactions on Industrial Informatics*, 2019, ISSN: 1551-3203. DOI: `10.1109/TII.2019.2925898`.

[91] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey", *IEEE Access*, vol. 8, pp. 16 440–16 455, 2020. DOI: `10.1109/ACCESS.2020.2967218`.

[92] M. S. Ali, K. Dolui, and F. Antonelli, "IoT Data Privacy via Blockchains and IPFS", in *Proceedings of the Seventh International Conference on the Internet of Things*, ser. IoT '17, New York, NY, USA: ACM, 2017, 14:1–14:7, ISBN: 978-1-4503-5318-2. DOI: `10.1145/3131542.3131563`. [Online]. Available: `http://doi.acm.org/10.1145/3131542.3131563`.

[93] H. Debar, D. Curry, and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)", RFC Editor, RFC 4765, Mar. 2007, pp. 1–157. DOI: `10.17487/RFC4765`.

[94] J. Benet, *Ipfs - content addressed, versioned, p2p file system*, 2014. arXiv: `1407.3561 [cs.NI]`.

[95] G. D. Putra, V. Dedeoglu, S. S. Kanhere, R. Jurdak, and A. Ignjatovic, "Trust-Based Blockchain Authorization for IoT", *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1646–1658, Jun. 2021, ISSN: 1932-4537, 2373-7379. DOI: 10.1109/TNSM.2021.3077276.

[96] S. Leonardos, D. Reijsbergen, and G. Piliouras, "Weighted voting on the blockchain: Improving consensus in proof of stake protocols", *International Journal of Network Management*, vol. 30, no. 5, 2020. DOI: https://doi.org/10.1002/nem.2093. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nem.2093.

[97] Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, G. K. Karagiannidis, and P. Fan, "6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies", *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 28–41, Sep. 2019, ISSN: 1556-6080. DOI: 10.1109/MVT.2019.2921208.

[98] A. Kalla, C. De Alwis, G. Gur, S. P. Gochhayat, M. Liyanage, and P. Porambage, "Emerging Directions for Blockchainized 6G", *IEEE Consumer Electronics Magazine*, pp. 1–1, 2022, ISSN: 2162-2256. DOI: 10.1109/MCE.2022.3164530.

[99] P. Robinson, R. Ramesh, and S. Johnson, "Atomic Crosschain Transactions for Ethereum Private Sidechains", *Blockchain: Research and Applications*, vol. 3, no. 1, p. 100 030, Mar. 2022, ISSN: 2096-7209. DOI: 10.1016/j.bcra.2021.100030.

[100] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator", *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.

[101] S. B. Prathiba, G. Raja, S. Anbalagan, K. Dev, S. Gurumoorthy, and A. P. Sankaran, "Federated Learning Empowered Computation Offloading and Resource Management in 6G-V2X", *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021, ISSN: 2327-4697. DOI: 10.1109/TNSE.2021.3103124.

[102] G. Ramachandran, D. Nemeth, D. Neville, D. Zhelezov, A. Yalçin, O. Fohrmann, and B. Krishnamachari, "Whistleblower: Towards a decentralized and open platform for spotting fake news", in *2020 IEEE International Conference on Blockchain (Blockchain)*, IEEE, 2020, pp. 154–161.

[103] K. Kotobi and S. G. Bilen, "Secure Blockchains for Dynamic Spectrum Access: A Decentralized Database in Moving Cognitive Radio Networks Enhances Security and User Access", *IEEE Vehicular Technology Magazine*, vol. 13, no. 1, pp. 32–39, Mar. 2018, ISSN: 1556-6080. DOI: 10.1109/MVT.2017.2740458.

[104] A. Dorri, F. Luo, S. S. Kanhere, R. Jurdak, and Z. Y. Dong, “Spb: A secure private blockchain-based solution for distributed energy trading”, *IEEE Communications Magazine*, vol. 57, no. 7, pp. 120–126, 2019. DOI: 10.1109/MCOM.2019.1800577.