CONSORTIUM BLOCKCHAIN MANAGEMENT WITH A PEER REPUTATION SYSTEM FOR CRITICAL INFORMATION SHARING

A Thesis presented to
the Faculty of the Graduate School
at the University of Missouri

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
SOUMYA PUROHIT
Dr. Prasad Calyam , Thesis Supervisor
December 2020

The undersigned, appointed by the Dean of the Graduate School, have examined the thesis entitled:

“CONSORTIUM BLOCKCHAIN MANAGEMENT” WITH A PEER REPUTATION SYSTEM FOR CRITICAL INFORMATION SHARING

presented by Soumya Purohit, a candidate for the degree of Master of Science and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Prasad Calyam

Dr. Praveen Rao

Dr. Praveen Edara

ACKNOWLEDGMENTS

I would like to thank Dr. Prasad Calyam for his great support throughout my entire Master's degree. I appreciate the outstanding opportunity given to me by Dr. Calyam to work in Virtualization, Multimedia and Networking (VIMAN) Lab with a talented group of individuals on a state-of-art research area and lab facilities. I would like to express my gratitude towards Dr. Praveen Rao and Dr. Praveen Edara for their interest to be a part of my thesis committee. Next, I would like to thank my research partner Songjie Wang for helping me throughout this research.

I would like to thank my family for supporting and showing their confidence in me. Finally, I would like to thank my friends for making this journey exciting and joyful for me.

Soumya Purohit

Contents

List of Tables

Table Page

List of Figures

Figure Page

ABSTRACT

Blockchain technology based applications are emerging to establish distributed trust amongst organizations who want to share critical information for mutual benefit amongst their peers. There is a growing need for consortium based blockchain schemes that avoid issues such as false reporting and free riding that impact co-operative behavior between multiple domains/entities. Specifically, customizable mechanisms need to be developed to setup and manage consortiums with economic models and cloud-based data storage schemes to suit various application requirements.

In this MS Thesis, we address the above issues by proposing a novel consortium blockchain architecture and related protocols that allow critical information sharing using a reputation system that manages co-operation amongst peers using off-chain cloud data storage and on-chain transaction records. We show the effectiveness of our consortium blockchain management approach for two use cases: (i) threat information sharing for cyber defense collaboration system viz., DefenseChain, and (ii) protected data sharing in healthcare information system viz., HonestChain. DefenseChain features a consortium Blockchain architecture to obtain threat data and select suitable peers to help with cyber attack (e.g., DDoS, Advance Persistent Threat, Cryptojacking) detection and mitigation. As part of DefenseChain, we propose a novel economic model for creation and sustenance of the consortium with peers through a reputation estimation scheme that uses 'Quality of Detection' and 'Quality of Mitigation' metrics. Similarly, HonestChain features a consortium Blockchain architecture to allow protected data sharing between multiple domains/entities (e.g., health data service providers, hospitals and research labs) with incentives and in a standards-compliant manner (e.g., HIPAA, common data model) to enable predictive healthcare analytics. Using an OpenCloud testbed with configurations with Hyperledger Composer as well as a simulation setup, our evaluation experiments for DefenseChain and HonestChain

show that our reputation system outperforms state-of-the-art solutions and our consortium blockchain approach is highly scalable

Chapter 1

Introduction

1.1 Background on Information Sharing

Cloud-hosted services are targeted by ever-growing Distributed Denial of Service (DDoS) attacks that aim to disrupt the service of major industries, conglomerates and community organizations [47]. Attacks such as Advanced Persistent Threats (APTs) also cause economic damage and leakage of sensitive information through sophisticated malicious attack codes [19]. Another targeted attack type can be seen in the cryptojacking attacks [48], where criminals compromise enterprise resources for illegal bitcoin mining revenue gains.

To defend against such targeted attacks, a co-operative and collaborative attack threat intelligence sharing platform can help raise the situational awareness and foster mechanisms to protect targeted assets through pertinent detection and mitigation of attacks. The platform can produce proactive measurements and actionable information that can be available to multiple domains/entities in a federation [49].

Moreover, organizations under a line of attack in close proximity can leverage the platform to form alliances for collaborative defense by sharing the burden [52]. The need of information sharing is also seen in applications of healthcare where there is the increase in data-driven methods to create healthcare innovations that requires utilization of voluminous, high-variety and standards-compliant (e.g., HIPAA [27], common data model [46]) distributed datasets that enable predictive analytics.

The authorization step in health information systems can cause data access bottlenecks due to trust issues among the data custodians and requesters. Such trust issues lead to the fear of "Loss of Value" for the data being provided by the data custodians. To cope with risks associated with "Loss of Value" and to ensure assurance/auditability in data access [5], data custodians use high-touch methods that require a governance committee to *manually* approve data requests. Consequently, manual approvals in the data access transactions cause long queues of data requests. In addition, human error in the forms filled out by data requesters can prolong the deliberation of the governance committee and cause over-provisioning or under-provisioning of data queries for data analytics/visualization. These factors cause frictions in the innovation process and causes delayed patient care decisions, which ultimately leads to a "Loss of Opportunity" [23] in the requested data. A trustworthy cooperation can allow faster accessibility and help in building trust in protected healthcare records as well as in cyber threat data.

1.2 Need for Blockchain for Information Sharing

Creation of threat intelligence sharing platforms requires overcoming other substantial challenges that include issues such as: why should one domain share its threat intelligence information with another domain? How can we opt-in the domains/entities that are *proximal* (i.e., in geographic distance or units that are distributed but belong to the same organization) or *distant* (i.e., relatively far in geographic distance or belonging to different organizations) for collective attack defense? How can the platform be used for co-ordinated threat detection and attack impact mitigation in a timely manner with distributed trust? A subset of these issues have been addressed in prior works using methods such as crowdsourcing with incentives [26] [55]. Reputation systems also have been proposed with algorithms to counter the impact of having false reporting and free riding peers [53] [54] [14]. However, there is a lack of works that use Blockchain solutions that can potentially be used to establish distributed trust, integrate reputation systems and create automated access control for threat intelligence data sharing in a scalable and transparent manner.

In the field of healthcare, *Blockchain* technology can be a promising solution for creating effective techniques that can help in minimizing the above Loss of Value and the Loss of Opportunity issues in health information systems. However, a solution with Blockchain technology should address the following research questions: How to convey diverse datasets and overcome the lack of trust between parties involved in health data sharing? How can the health records be tamper-proof and distributed to allow for a faster data accessibility? How to enable a consistent representation of authorization to access health information in a secured network platform? How to create a smart platform equipped with a conversational agent i.e., a *chatbot* that interacts with the data consumer with a knowledge-driven capability to have a better consumer-platform interaction? How can the quality of health information service be improved by incentives for data consumers and providers to cooperate?

1.3 Thesis Outline

The remainder of this thesis is organized as follows: In Chapter 2, we describe one of the usecase "DefenseChain" Solution Approach and evaluation. In Chapter 3, we elaborate on our second usecase on HonestChain Solution Approach and evaluation. Adviser with system design and evaluation. Same way, Chapter 4 conclude thesis with future work.

Chapter 2

“DefenseChain” Solution Approach

In this chapter, We discuss the various literature work that have led to the idea and implementation of this research and then we discuss solution design for “DefenseChain”.

2.1 Related Work

2.1.1 Threat Intelligence Sharing

Due to the constant increase in the number and complexity of cyber attack incidents, organizations are eager to have proactive and actionable knowledge for efficiently defending their valuable assets i.e., cloud-hosted applications. Towards this end, they need to develop the practice to share threat intelligence information amongst their peers in order to effectively and collectively detect cyber attacks, and stand up robust defenses that mitigate the attack impact on their assets.

Several works have been performed to enable cyber defenders to explore threat intelligence sharing capabilities and construct effective defenses against the ever-changing cyber threat landscape. The authors in [38] and [18] identify gaps in existing technologies and introduce the Cyber Threat Intelligence model (CTI) and a related cyber threat intelligence ontology approach, respectively. The work in [9] details a novel approach based on Structured Threat Information eXpression (STIX) to deal with system diversity during threat information sharing. An encryption strategy for threat intelligence sharing is proposed in [10] in the form of a privacy preserving protocol. The CYBEX work in [50] details an incentivized approach and uses the concept of an admission fee, as well as interaction models organizations for cybersecurity information exchange to defend against attackers in a dynamic game. fThe novelty of our work is in the design of a threat intelligence sharing platform using consortium Blockchain in order to implement a 'defense by pretense' paradigm for cyber defense as detailed in the work on the Dolus system [40]. We adapt the two-stage ensemble learning scheme to trigger co-operation between multiple domains who collectively provide detection and impact mitigation to defend a domain targeted by attackers through DDoS, APTs and cryptojacking.

2.1.2 Reputation Systems

Several different reputation systems have been proposed in prior works that address the issues of false reporting and free riding [55], [53], [54]. The work in [55] proposed the design of a crowdsourcing tournament to maximize a service provider's utility in crowdsourcing and provide continuous incentives for users by rewarding them based on the rank achieved. The authors in [53] presented schemes to eliminate dishonest behavior with the help from a trusted third party. In a related effort [21], a reputation system is developed that overcomes the limitations in decentralized systems and quantifies the reputation by removing human opinion from the transactions. E-commerce applications [45] have also adopted reputation systems that use Blockchain solutions for implementation of privacy-preserving mechanisms involving Proof of Stake for determining any new block to be accepted instead of accepting the highest difficulty block. The authors in [36] designed a trust model that evaluates trust based on the reputation built up on historical interactions and indirect opinions about the sender. The work in [29] introduces a proxy to transfer reputation values between anonymous contributions, and a reputation anonymization scheme is shown to prevent the inadvertent leakage of privacy.

The closest related work to our work is in [25]. Therein, a reputation and reward scheme is proposed that considers potential information frauds and allows automatic smart contract execution based on malicious peers. We adapt their Beta reputation that is used for probabilistic rating and to identify and reward honest participants. Our work also borrows the idea of using a InterPlanetary File System (IPFS) [11] for creation of the reputation system and to store device attributes as well as threat data in an off-chain manner in our Blockchain architecture. We include the concept of a deposit, and a request/response deadline to eliminate free-riding cases and false reporting similar to the work in [25]. Furthermore, we propose a novel objective evaluation of attack detection and impact mitigation through real-time threat intelligence sharing using novel QoD and QoM metrics. Our reputation system also features a trust-based model implemented using threat detection and attack impact mitigation protocols that are motivated by prior work in [54] for incentivizing domains in a federation to co-operate and trust each other.

2.1.3 Blockchain for Building Trust

There have been several studies that utilize Blockchain as a solution in order to solve the problems inherent in traditional transactional models. CrowdBC [33] is an exemplar work that implements a reward/penalty scheme using smart contracts, and explores the ability to abstract a user's real world identity for providing a unique method to ensure data privacy. In the area of IoT and sensor networks, works such as [39] proposed security models based on Blockchain to ensure the validity and integrity of cryptographic authentication data. A Blockchain-based security model is proposed for forensic evidence preservation [17] in order to allow storage of metadata e.g., pieces of evidence using smart contracts amongst the different entities involved in an investigation process. The authors in [42] use

Blockchain to perform edge computing resource allocation to IoT devices using a policy-based security model to regulate malicious requests. Similarly, iShare [44] features a security model that leverages Blockchain to collect cyber attack information and shares it across organizations in an anonymous fashion. The anonymity afforded by this approach serves as inspiration to our approach to threat intelligence sharing across a federation of proximal/distant domains. Anonymity issues have also been tackled in [35], where Blockchain is used to enable anonymous reputation estimation as part of establishing privacy-preserving trust for vehicular ad hoc networks.

Our work on DefenseChain is motivated by the above works in the context of designing our reputation system using Blockchain technologies, and for incentivizing federation peers via an economic model based on a deposit fee received from potential detector and mitigator peers.

2.2 Solution approach of “DefenseChain”

2.2.1 DefenseChain Platform Overview

Fig. 2.1 illustrates our proposed reference architecture in a federation where a cloud service provider is hosting several servers belonging to different organization peers that may be vulnerable to cyber attack threats. Roles of the peers involved in the federation area defined in Section III-C. The central part of our DefenseChain architecture is the consortium Blockchain-based trust setup created on top of the Dolus defense by pretense implementation as outlined in [40]. Within this federation, we assume that there are organization peers requesting for a detection and mitigation service from cooperating domains. Furthermore, each domain can perform their service using a suitable mitigation strategy such as e.g., moving target defense, defense by pretense, network firewall defense using blacklisting, etc. Our DefenseChain rates the detection and mitigation service quality of the peer(s), and provides the requesting peer(s) with the flexibility to choose the domain that can provide the higher levels of service quality measured through the QoD and QoM metrics that are detailed in Section III-D. Furthermore, through our economic model described in Section III-E, we implement an incentivized approach that allows the mitigator domains to collaborate and also eliminates the issues of free riding and false reporting. Additionally, our platform design includes on-chain and off-chain components for storage, processing and sharing of threat intelligence information. We elaborate on these components in the following:

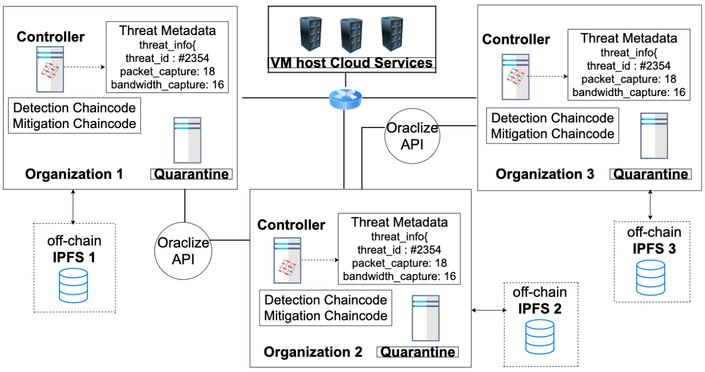

Figure 2.1: Proposed DefenseChain reference architecture that features on-chain/off-chain components within a federation of peers involving a cloud-hosted application, dedicated controllers with Hyperledger configurations, IPFS and QVMs integration.

On-Chain: this component fetches and displays the details such as e.g., attacker IP, source IP, number of packets, spoofed IP, blacklisted IP from the IPFS. These details are fed into the detection and mitigation chaincodes that initialize and manage ledger state through transactions submitted by applications. In our DefenseChain, they help in calculation of QoD and QoM in a federation of peers, respectively.

Off-Chain: this component stores information such as the packet capture, bandwidth capture and device attributes data. Depending on the number of transactions and the attacks encountered, the storage of the related data will require large amounts of storage (in the order of tera bytes or even peta bytes in core network domain scenarios). For this purpose, we utilize the IPFS concept from [11] as an off-chain storage that interacts periodically through the Oraclize [?] service. The hashes of the IPFS data are referenced in our chaincodes. This approach allows us to deliver a dynamic and efficient data retrieval in a peer-to-peer manner.

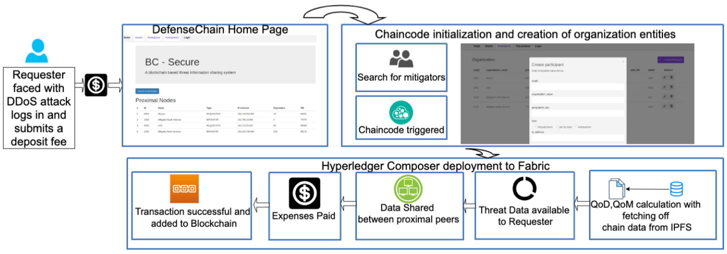
Requester faced with DDoS attack logs in and submits a deposit fee
DefenseChain Home Page
BC - Secure
A blockchain based threat information sharing system
Proximal Nodes
Chaincode initialization and creation of organization entities
Search for mitigators
Chaincode triggered
Create participant
Hyperledger Composer deployment to Fabric
Transaction successful and added to Blockchain
Expenses Paid
Data Shared between proximal peers
Threat Data available to Requester
QoD,QoM calculation with fetching off chain data from IPFS

Figure 2.2: DefenseChain workflow with requester peer(s) and detector/mitigator peer(s) sharing threat information co-operatively.

2.2.2 User Interface

We utilized the Hyperledger Composer playground for configuration, deployment, and testing of our business network i.e., federation. A new federation can be created by a requesting peer organization. The business networks are a combination of identity and profile, and hence they are viable for permission and access control. We created userIds and secret passwords for the peers to connect into a business network. In our security policy, we use a Federated Identity and Access Management scheme, where mapping of peers in the Hyperledger Composer matches with the real-world identity of the peers. Initially, the requester, detector and mitigator roles pay a deposit fee that initiates the transaction process, as shown in Fig. 2.2. Upon successful validation by a detection or mitigation chaincode, the requester can search and view the detectors/mitigators.

We share the threat information such as assets affected, attack tools, QVM IP addresses, blacklisted IP addresses, attack duration times, etc. The attack data is stored on the controller of the peer node, which is then fetched through the IPFS as shown in Fig. 2.2. This dynamic fetching off-chain occurs rapidly. At this step, some delays can be experienced depending upon the network performance. Furthermore, proximal peers which are likely to be attacked can benefited from this information by our mitigation protocol. We allow the requester to choose the proximal peers, which are acceptable in the chance of getting attacked based on their geographical location or domain affinity.

2.2.3 System Roles

Requester

Requester is an actor who is affected by cyber attacks and submits the detection/mitigation requests to the federation. Once a threat is identified, the requester has the option to search for detector(s) and mitigator(s), and can decide which of the peer service providers are ideal to trust. All the actors have to pay a deposit fee, which includes the transaction fee and a collateral. The transaction fee is refunded in the case a mitigation strategy could not be effectively devised; otherwise, the fee is credited to the detector(s)/mitigator(s) providing the services.

Detector

Detector is an actor who provides cyber attack detection services to the attack defense requests from a Requester. Upon providing the service, the Detector receives the transaction fee as payment for a successful detection. To incentivize Detectors to provide high-quality services, a reward v and a monetary penalty π_{D} are required as deposit ($deposit_D = v + \pi_D$) in the DefenseChain. This deposit cannot be redeemed before the detection deadline.

Mitigator

Mitigator is an actor who provides mitigation services to all the attack defense requests from Requesters. Upon providing the service, the Mitigator receives the transaction fee as payment for successful mitigation. Each Mitigator must make a deposit of π_{M_j} in the DefenseChain, which serves to significantly reduce the possibility of Sybil and Collusion attacks (see Section III-G for definitions). Efficient strategies by $\mathrm{M_j}$ will result in a corresponding reward v_R.

Watchdog

The Watchdog is a system Daemon which is essentially an admin role that is used to: (a) analyse the detection and mitigation data, and (b) rate whether the detection or mitigation has been successful. The Watchdog will also flag false reports by analyzing the data that it monitors. False reporters will be penalized in their transactions and could loose out on their collateral and in their ability to perform future transactions. The output given by the Watchdog determines the rating and reward that will be received by a Detector/Mitigator who claims successful service completion.

2.2.4 QoD and QoM Protocols for Decision Making

Our reputation system allows objective rating of the detectors/mitigators after the threat data has been shared by the requesters. We devise two protocols to govern the detection and mitigation performed within DefenseChain. These protocols allow threat data transmission sequentially through a software-defined network (SDN) infrastructure.

Quality of Detection (QoD)

Fig. 2.3 describes the process of the attack detection protocol. When an attack is active on a federation peer, the traffic within a cloud provider's network through the SDN switches can be monitored using a Frenetic run-time enabled monitoring sub-component [40]. Next, in order to learn and classify the attacks, the DefenseChain employs a two-stage ensemble learning scheme used in the Dolus system on the incoming traffic, both from the attackers' side and from the benign users' side. In order to differentiate attackers from benign users, the first stage handles outlier detection to identify salient events of interest (e.g., connection exhaustion), whereas the second stage handles outlier classification to distinguish different attack event types.

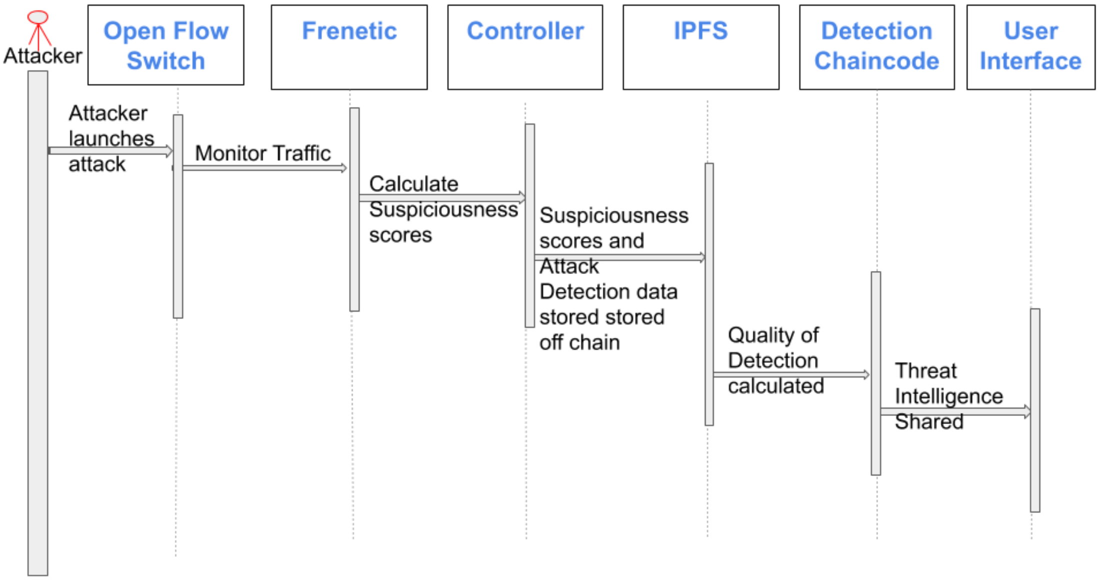
Attacker
Open Flow Switch
Frenetic
Controller
IPFS
Detection Chaincode
User Interface
Attacker launches attack
Monitor Traffic
Calculate Suspiciousness scores
Suspiciousness scores and Attack Detection data stored stored off chain
Quality of Detection calculated
Threat Intelligence Shared

Figure 2.3: Sequence diagram of the DefenseChain detection protocol that involves attack detection, IPFS data retrieval, and detection chaincode to calculate QoD.

The controller calculates the "suspiciousness score" of a domain node as detailed in [40] and summarized in Section IV-B. It stores the calculated score along with the attack detection data (e.g., attacker ID, source IP, number of packets, attack start time, attack end time, response time and deadline time) in the IPFS. From the IPFS, the related data and score information is consumed by the detection chain code. Using the data and the suspiciousness score information, the QoD score is calculated and displayed on the user interface.

We present the following mathematical formulation to determine the QoD values (see Table I for notation details). Let $a_i \in [0, 1]$ be the accuracy used to determine the closeness of measurement of detection. $S_{norm} \in [0, 5]$ is the suspiciousness score which is calculated on the basis of number of unique destinations, total number of connections and the total number of bytes transferred for different type of attacks encountered. Let k represent the number of types of attacks encountered on the requester side. The time for QoD estimation is divided into response and detection times. Let t_r be the response time defined by -

$$t_r = e^{\frac{t_{deadline} - t_r}{t_{deadline}}} \tag{2.1}$$

and t_d be the detection time defined by -

$$t_d = \frac{t_d - t_{min}}{t_{max} - t_{min}} \tag{2.2}$$

$$A = \tfrac{1}{n}\sum_{i=1}^{n} \tfrac{a_i * \sum_{j=1}^{k} \frac{S_{norm}}{k}}{t_d} \tag{2.3}$$

Our final QoD calculation considers the response time as a deciding factor in assigning the scores. Let y be a Watchdog defined penalty variable that is charged when the response time is greater than the deadline time. For our use case, we consider y as 0.8 that is a 20 percent reduction in the score. The QoD is thus given by:

$$QoD = \begin{cases} A, if \mathrm{t}_r \leq \mathrm{t}_{deadline} \\ A \cdot y, if \mathrm{t}_r > \mathrm{t}_{deadline} \end{cases} \tag{2.4}$$

Quality of Mitigation (QoM)

Fig. 2.4 describes the process of the attack mitigation protocol. Once attack detection is done, the Requester then submits the request for attack mitigation to the mitigator(s) based on trust considerations. The mitigation chaincode has the script that triggers the appropriate mitigation policy (i.e., to automate the mitigation mechanism). Once the appropriate mitigation policy is set, the controller redirects the attack traffic to the QVMs. Meanwhile the mitigation mechanism for a particular attack checks the resource availability and submits a response time to the mitigation chaincode. The mitigation chaincode takes into account the availability of resources, the service response time and detection effectiveness in order to calculate the QoM.

We propose the following mathematical formulation to determine the QoM. Let $t_m \in [1, 10]$ be the time taken to mitigate an attack impact, which is given by

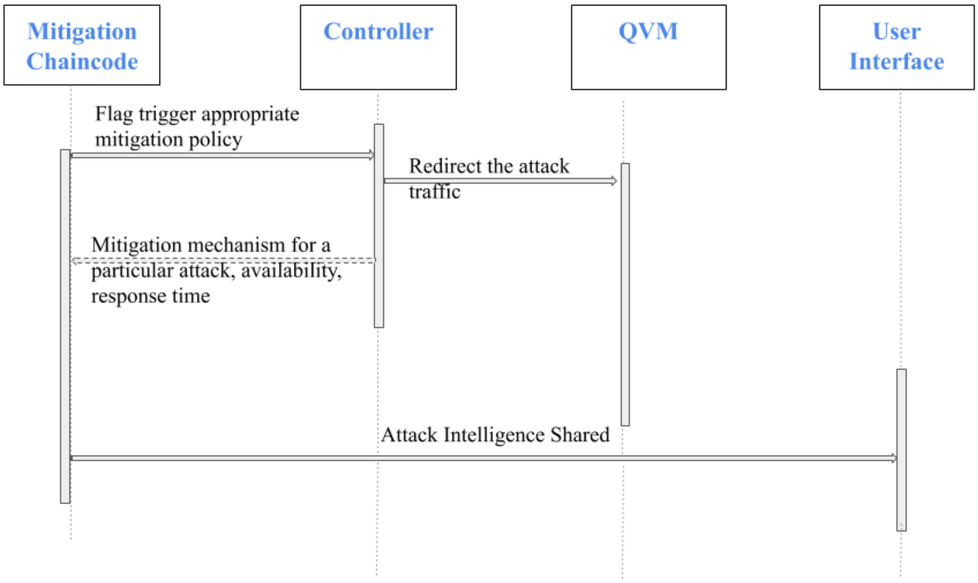
Mitigation Chaincode
Controller
QVM
User Interface
Flag trigger appropriate mitigation policy
Redirect the attack traffic
Mitigation mechanism for a particular attack, availability, response time
Attack Intelligence Shared

Figure 2.4: Sequence diagram of the DefenseChain mitigation protocol involving interactions of pretense initiation, redirection of attack traffic and mitigation chaincode to calculate QoM.

-

$$t_m = \frac{t_m - t_{min}}{t_{max} - t_{min}} \tag{2.5}$$

Let $r \in [0,1]$ be the attack reocurrence rate and $S_r \in [10,100]$ be the success rate of the Mitigator. The QoM is thus given by:

$$QoM = \frac{\sum_{i=1}^{n} \cdot S_r}{e^r \cdot t_m} \tag{2.6}$$

2.2.5 Incentives for Sharing

We also consider domain reputation as a factor in performing decision making. It allows the DefenseChain to choose a detector or a mitigator based on their respective historic reputations. We follow a semi-legal approach, where we focus on determining the reputation of a detector/mitigator based on their service performance and deposit fee factors. With the historic reputation information, and owing to the design of the detection/mitigation protocols in our scheme, we enable a trust building platform in DefenseChain for threat detection and mitigation. A

Table 2.1: Notations used in this paper.

Notation	**Description**
t_d	time taken to detect an attack
t_m	time taken to mitigate an attack
t_{min}	assumption made about minimum time taken to detect/mitigate an attack
t_{max}	assumption made about maximum time taken to detect/mitigate an attack
t_r	service response time
S_{norm}	Suspiciousness Score for each type of attack
y	penalty factor
S_r	successful rate to mitigate attack impact
e^r	attack reoccurence rate
n	number of interactions
β	initial reputation
β_k	average reputation
β_w	overall reputation
f	fee

higher reputation score leads to a higher probability of a peer being selected for providing detection/mitigation services in the future.

We categorize β_W into three intervals, where if the QoD/QoM value lies in the range of [0,3], a score of -1 is assigned. If QoD/QoM value is in the range of [3,7], then a score of 0 is assigned; similarly, a score of 1 is assigned if QoD/QoM is in the range of [7,10]. Additionally, β_k represents the mean reputation value of the detector(s)/mitigator(s). If there is a case of no deposit submission, and if the reputation is less than the mean reputation value, then the β_W is set to negative one. If the deposit fee and the reputation is higher than the mean reputation, then the β_W is set to zero. Following this logic, we define our β_{W} as follows:

$$\beta_{\mathrm{W}} = \begin{cases} -1, \ if\ QoD/QoM \ \varepsilon \ [0,3], \ \beta \leq \beta_k \text{ and } f = 0 \\ \ \ 0, \ if\ QoD/QoM \ \varepsilon \ [3,7], \ \beta \geq \beta_k \text{ and } f = 0 \\ \ \ 1, \ if\ QoD/QoM \ \varepsilon \ [7,10] \end{cases}$$

2.3 Performance Evaluation

In this section, we evaluate our DefenseChain platform by performing real-time threat sharing of DDoS, APTs and cryptojacking in realistic experiments. We show how DefenseChain effectively allows a federation of peers to leverage the attack threat information that is shared across multiple domains. Further, we compare the performance of our DefenseChain solution with state-of-the-art schemes proposed in related works such as [13] [34] [30] and [25].

2.3.1 Experiment Testbed Setup

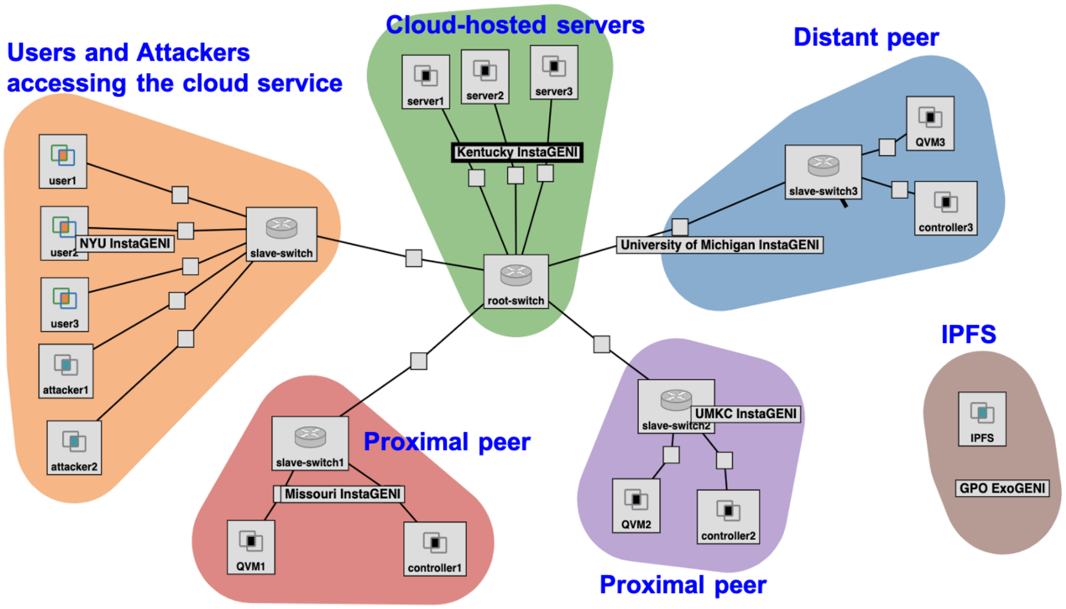

Figure 2.5: NSF Cloud [12] testbed setup used to evaluate DefenseChain performance with experiments involving SDN for testing platform capabilities.

We implemented our DefenseChain using the NSF Cloud [12] infrastructure as shown in the Fig. 2.5. In this testbed, we created a peer federation network, where each peer organization is connected through a central root switch. Each peer organization has a dedicated QVM and a Controller to perform detection and mitigation protocols. We installed the Hyperledger Composer platform on the controllers of each organization. We introduced different channels on the Blockchain by having the concept of inter and intra-organization and deploying them on virtual machines. All of these components were connected via a network switch that facilitated interactions between the federation of users.

The Missouri InstaGENI and UMKC InstaGENI nodes Fig. 2.5 act as proximal peers, whereas the Michigan InstaGENI acts as a distant peer. We leverage the IPFS deployed on a GPO ExoGENI node that is connected through the Oraclize service via a REST API. The interaction between on-chain and off-chain components is done through REST API calls and the Oraclize service.

Reputation scores and threat metadata can also be queried using REST API calls. Moreover, peers can query the transaction history, which includes fields such as: reputation score, number of interactions, and source IP.

2.3.2 Network Feature Analysis

We use Frenetic (an open-source SDN controller platform [40]) to execute Python scripts to identify suspicious packets, gather attack patterns in order to direct switches via SDN to redirect packets to pertinent QVMs. We then broadcast this information to the neighboring switches where the IP addresses of the attackers are blacklisted by updating a corresponding network policy. We randomize the attack data for DDoS, APTs and cryptojacking by changing e.g., the total bytes transferred, rate of transfer, connections made, and attack duration. This allows us to get dynamic suspiciousness scores of domain nodes for different targeted attacks. For example, in case of a DDoS attack, we exhaust the targeted application using a SlowHTTPTest and thereby cause random changes in e.g., number of packets, and attack times. We also perform event-based simulations to get different suspiciousness scores for different types of targeted attacks by:
Destination suspiciousness for trace t:

$$dst_i = w_{dst} \times \frac{numDst_i - numDstMin_i}{numDstMax_i - numDstMin_i} \tag{2.7}$$

Flow suspiciousness for trace t:

$$flows_i = w_{flows} \times \frac{numFlows_i - numFlowsMin_i}{numFlowsMax_i - numFlowsMin_i} \tag{2.8}$$

Bytes suspiciousness for trace t:

$$\begin{gathered} bytes_i = w_{bytes} \times \frac{numBytes_i - numBytesMin_i}{numBytesMax_i - numBytesMin_i}; \\ w_{dst} \in [0.0, 1.0]; w_{flows} \in [0.0, 1.0]; w_{bytes} \in [0.0, 1.0] \end{gathered} \tag{2.9}$$

Device suspiciousness for trace t:

$$ss_i = \sqrt{\frac{dst_i^2 + flows_i^2 + bytes_i^2}{3}} \tag{2.10}$$

We assume the weight parameters i.e., w_{dst}, w_{flows},w_{bytes} to be equal to 1 in a general case of suspiciousness score calculations. Also, the Min and Max values are assumptions made per device type based on the current knowledge of the device, as well as the network and traffic expectations.

2.3.3 Evaluation Results

We evaluate our DefenseChain through experiments using metrics such as detection time, mitigation time, attack reoccurence and peers' reputation. We compare the performance of DefenseChain with state-of-the-art schemes i.e., First Come First Serve (FCFS) [13], Random [34], Distance-based [30] and the Social Reputation models [25]. We performed simulation experiments by choosing a different set of Detectors and Mitigators over 25 iterations. This is to simulate real-world situations that allow us to create a fair chance of interactions. Each Mitigator and Detector have different values of data corresponding to the detection and mitigation strategies that they employ. Our DefenseChain picks a Detector or Mitigator from a non empty set and a two-stage simulation experiment is conducted. In these experiments, we consider the DDoS attack as our exemplar cyber attack scenario.

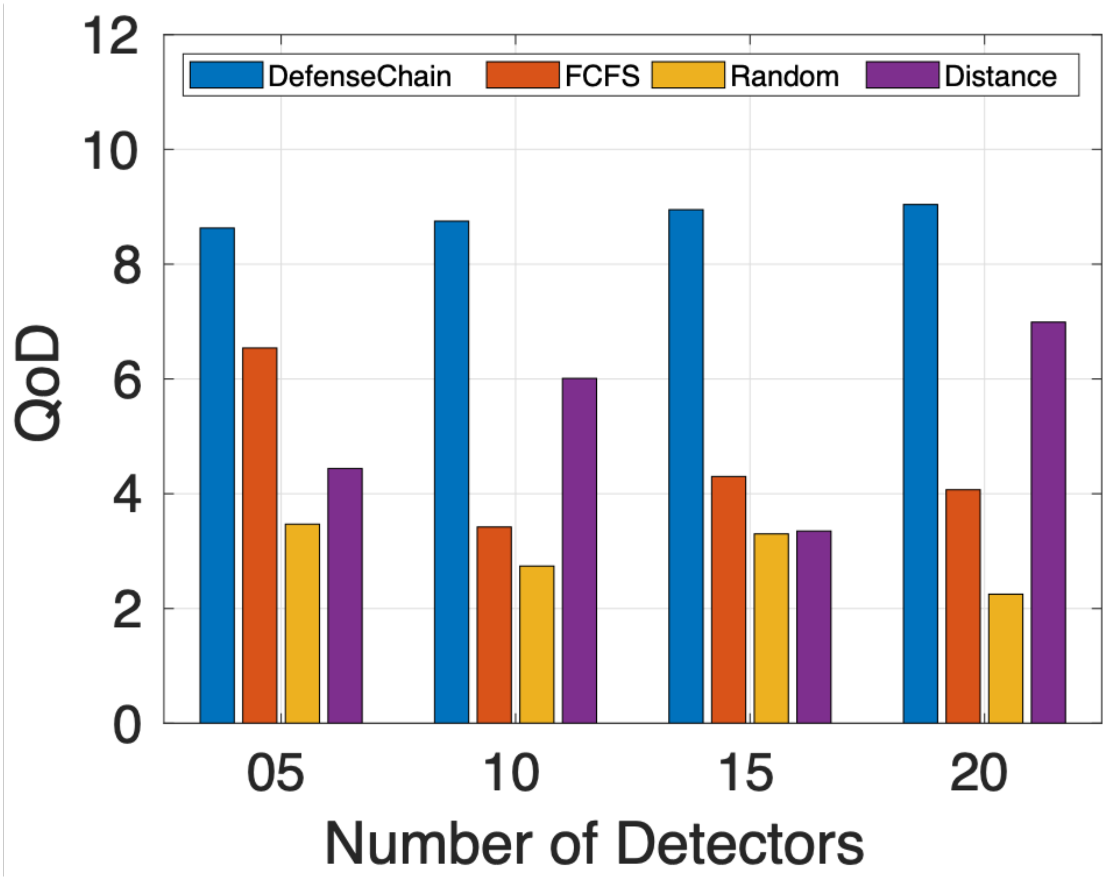

Figure 2.6: Performance comparison of DefenseChain with First Come First Serve, Random, and Distance-based schemes for detectors chosen to determine QoD

Our first experiment was to evaluate the decision making process in choosing a Detector and Mitigator, when a request arrives from a Requestor who is under a targeted attack. We simulated a total set of 20 Detectors and Mitigators. From this total set, we randomly generated a subset of 5, 10, 15, 20 Detectors and Mitigators for evaluating the various decision making schemes. With our DefenseChain, the Requestor ended up choosing the Detector(s)/Mitigator(s) based on their calculated QoD and QoM scores. The other schemes uses different algorithms for choosing the Detector and Mitigator. For instance, the Random scheme will randomly pick one DetecXtor/Mitigator, the FCFS scheme will choose the first peer who has responded to the request submitted by the Requestor, and the Distance-based scheme will choose the nearest Detector/Mitigator for detection/mitigation of the cyber attack. Our results show that Mitigators and Detectors chosen by our DefenseChain have better overall QoD/QoM scores when compared to the state-of-

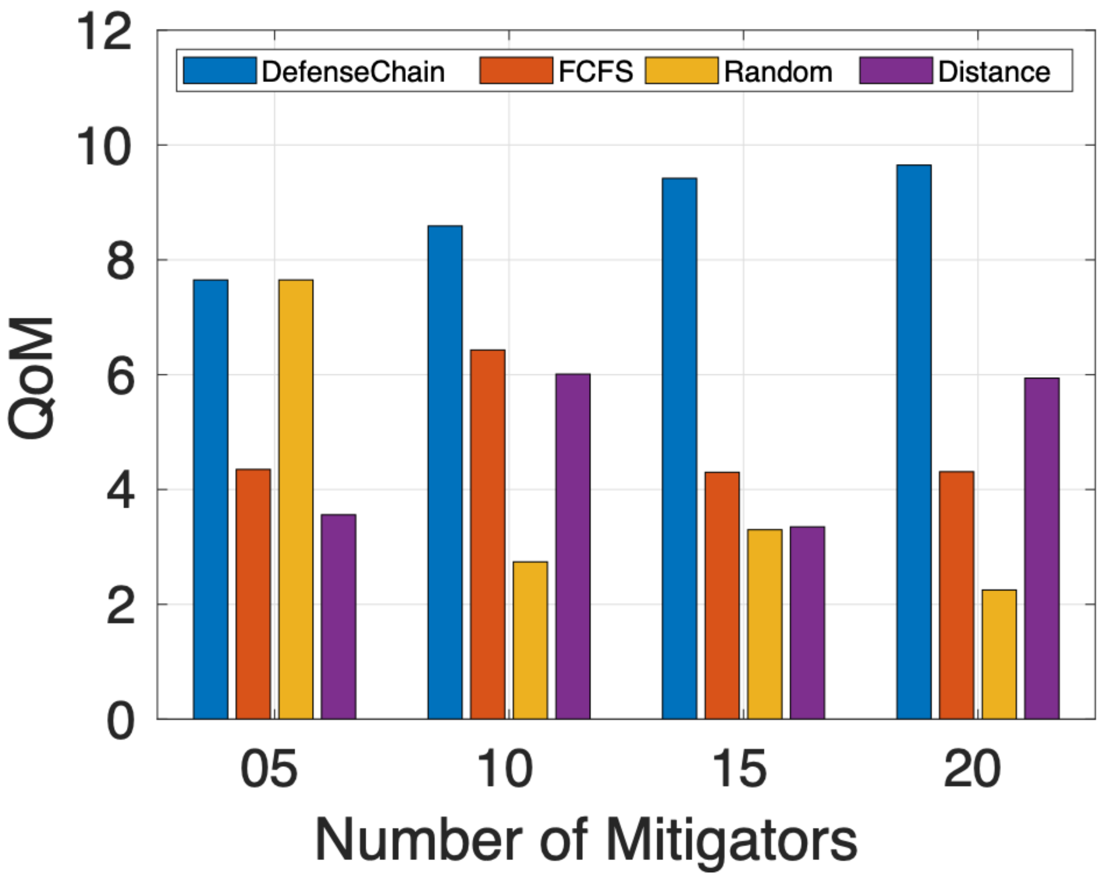

Figure 2.7: Performance comparison of DefenseChain with First Come First Serve, Random, and Distance-based schemes for mitigators chosen to determine QoM.

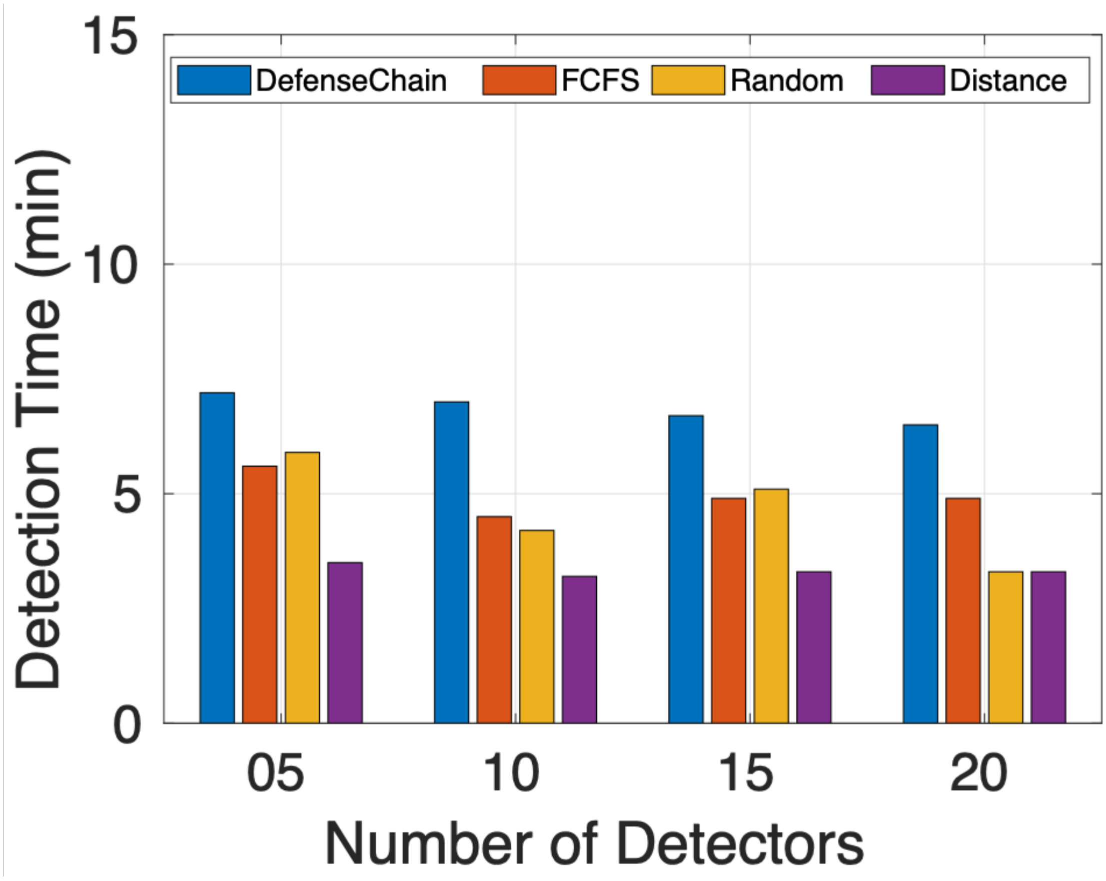

Figure 2.8: Performance comparison of DefenseChain with First Come First Served, Random, and Distance-based schemes for evaluating detectors on the basis of the time taken for detection

the-art schemes. This can be seen in Fig. 2.7 (a, b), where DefenseChain improved performance ranges from 1.3x - 4x times higher in terms of QoD/QoM values. This improvement obtained by our DefenseChain is due to the fact that we consider a comprehensive set of parameters to determine the Detection and Mitigation capabilities, rather than randomly choosing from a set of Detectors/Mitigators or using simplistic decisions considering only the order in response to the request or the distance from the Requestor, as in the other FCFS, Random, and Distance-based schemes being compared, respectively.

Upon choosing the Detector/Mitigator using our DefenseChain, we analyze the performance trade-offs in the detection time, mitigation time with the attack reoccurence metric. As shown in Figs. 3.3.1 (a, b, c), DefenseChain takes up to 2 times more time in detection and mitigation of cyber attacks as compared to the

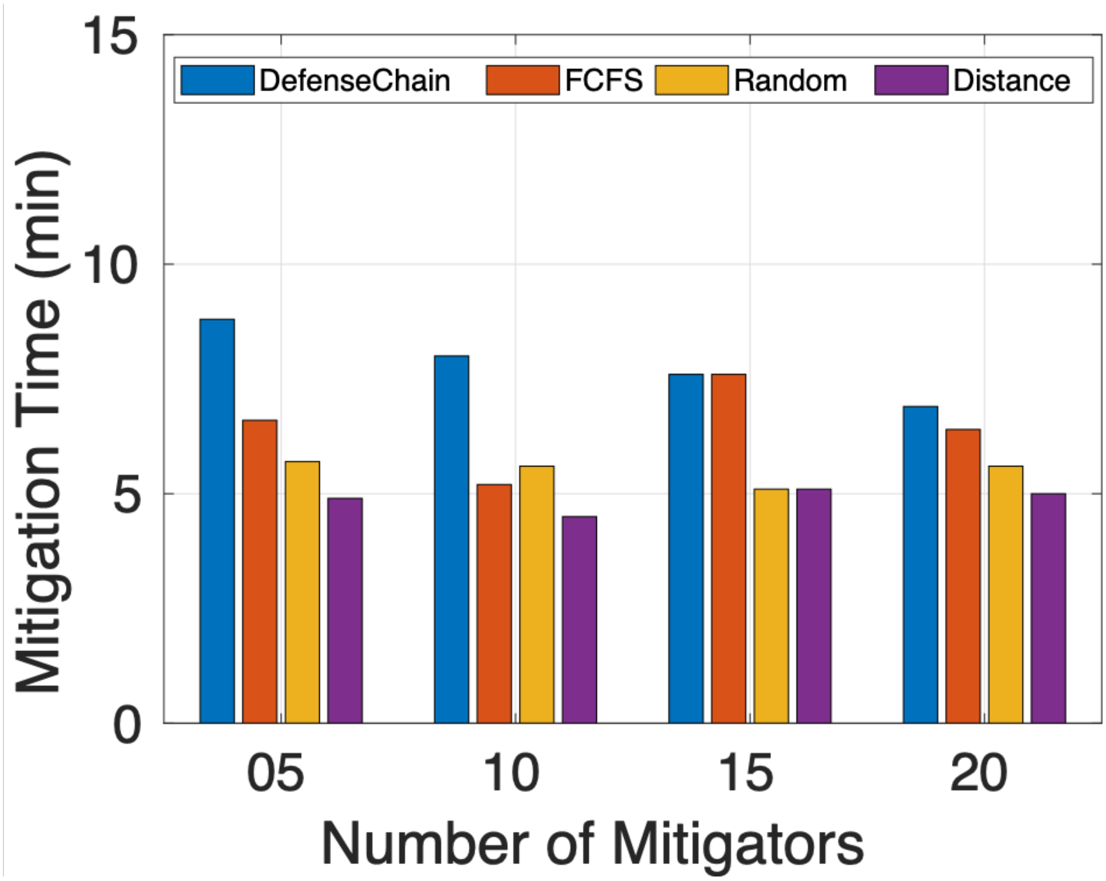

Figure 2.9: Performance comparison of DefenseChain with First Come First Served, Random, and Distance-based schemes for evaluating mitigators on the basis of the time taken for mitigation

other schemes, due to its multiple stages, i.e., policy update, attack traffic redirection and spoofing of the IP address during the detection/mitigation processes. However, these processes only consume a few minutes and these overhead times can be compensated by using the defense by pretense strategies that buy time for federation peers to create a robust cyber defense solution.

It is however important to note that our DefenseChain produces the least attack reoccurence rate, which is 10-100 times lower than other schemes, as shown in Fig. 3.3.1 (c). This is because of our policy enforcement approach to mitigate attacks with more secure mechanisms using the Dolus system. The reoccurence of the cyber attack is an important measurement of the quality of detection and the effectiveness of the mitigation services, and all Requestors will inherently give a much higher weight in real-world scenarios. Thus, we show that our DefenseChain

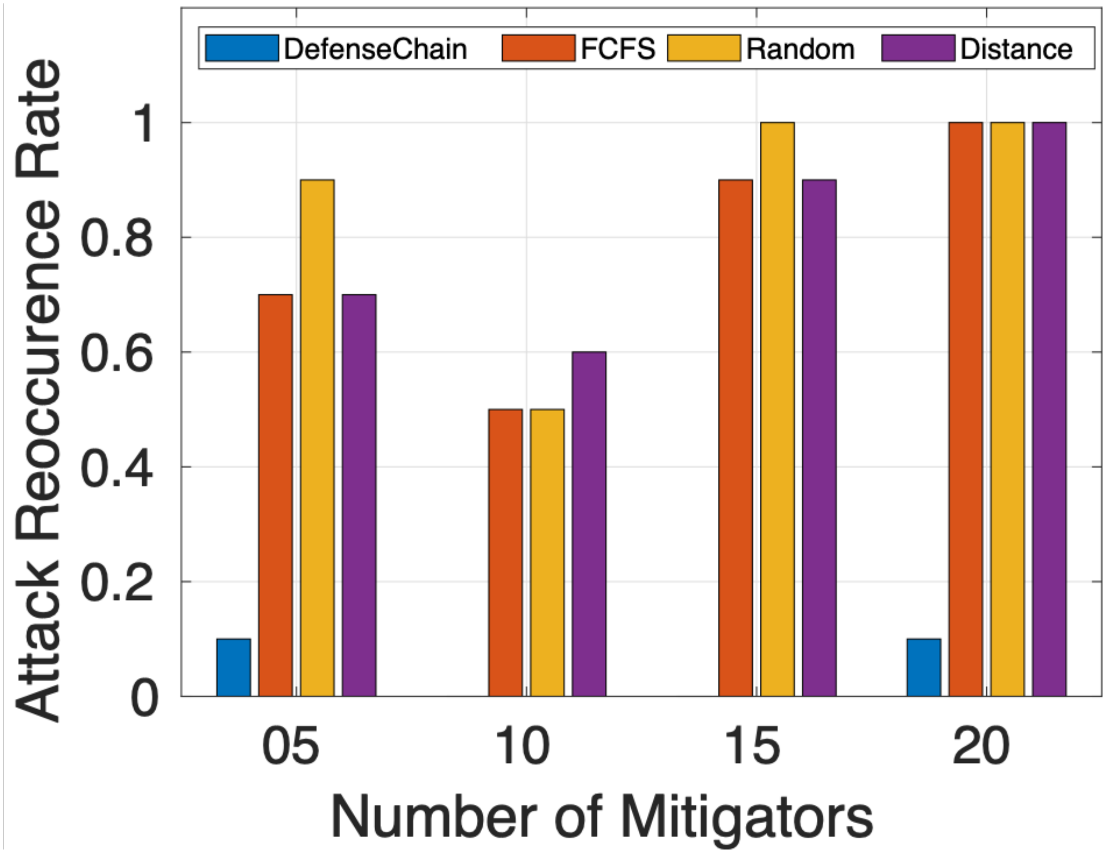

Figure 2.10: Performance comparison of DefenseChain with First Come First Served, Random, and Distance-based schemes for studying the their performance trade-offs in the context of the attack reoccurence rate.

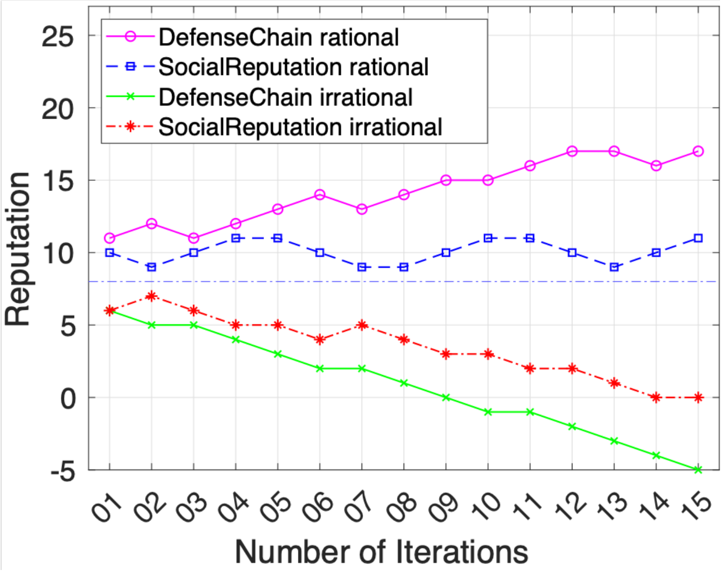

Figure 2.11: Performance comparison of DefenseChain with a SocialReputation model proposed in [25] in order to evaluate reputation values for rational and irrational mitigators.

has much higher performance considering trade-offs in comparison to other state-of-the-art decision-making schemes in choosing the appropriate Detectors and Mitigators in a federation.

Lastly, we compared our reputation system implementation with the Social-Reputation based model [25] to evaluate its efficacy in determining and dealing with rational and irrational mitigators. In this evaluation, we initialized reputation scores of rational and irrational mitigators as 11 and 6. Choosing a baseline helped us to get results comparable to a real-world setting. For each iteration performed, we show the cumulative reputation scores of both DefenseChain and SocialReputation. Using our feedback from our Watchdog service described in Section III-E, our DefenseChain identifies the rational/irrational Detectors/Mitigators and provides them with pertinent feedback based on their historic data and social data in the Blockchain. As shown in Fig. 2.11, with the increase in number of iterations, our DefenseChain shows a faster increase in reputation for rational mitigators due to its ability to choose the most capable and reliable Detectors/Mitigators using our comprehensive QoD/QoM scoring. Additionally, the reputation of irrational mitigators decreases at a much faster rate as our DefenseChain is more capable of identifying false reporters and free riders, who are assigned negative scores.

Chapter 3

HonestChain

3.1 Related Work

3.1.1 Health Information Sharing Systems

Health data sharing challenges have been widely approached by organizations for many years around efficient and secure ways to conduct the data brokering process. The data brokering process includes: data integration, data protection, Institutional Review Board (IRB) approval, brokering auditability, data assurance, and data request for analysis/visualization.

The data integration issue has been an extensively studied topic in prior works. Related efforts focus on defining frameworks to consolidate healthcare data from disparate sources into a unified platform. Authors in [16] proposed a common data model to serve as data hub for data brokering processes, which improves data accessibility and availability. Works such as [22] that aim to improve data protection automate the data de-identification process and centralize the IRB request evaluation.

While above works expedite the data brokering function, all of them however require human intervention to evaluate requests and manually approve or deny them. Moreover, they don't systematically address the issues related to auditability and assurance of the brokering process, and don't devise methods to minimize human intervention. In contrast, there have been recent efforts such as the work in [51] to semi-automate the honest broker processes through automation in compliance checking to expedite the data sharing.

Our goal in the current work is to extend the above prior works and implement a fully automated honest broker solution using Blockchain and chatbot technologies with minimal custodian-in-the-loop intervention. Our approach establishes distributed trust by improving efficiency in compliance checking, incorporating auditability, and including a common data model to improve data accessibility. Thus, we address long delays in data accessibility due to human intervention based on automation of assurance and auditability steps during requester-custodian cooperation in protected data access.

3.1.2 Guided Data Brokering with Trust

The work in [51] proposed a semi-automated honest broker that partially addresses the lack of trust between data custodians and data consumer in order to improve the data sharing process. Their methodology does not focus on improving the consumer interactions within a trusted health information sharing platform. Trust can be established by having data custodians use guided interfaces such as conversational agents i.e., chatbots to avoid human errors in over/under-provisioning of data requests or enable quick submission of protected complex data requests. Advantages of using chatbots have been presented in works such as [20], where mobile health care services have been improved using relevant knowledge bases to provide fast requirement analysis and quick response to address conditions of patients impacted by accidents. However, the design of chatbot guided systems need to be built in a manner that ensures maximum service, component re-usability and scalability. In addition, chatbots need to be designed with suitable natural language processing (NLP) techniques as detailed in [28] based on ranking and sentence similarity calculations.

The novelty of our approach is in the development of an automated brokering system using a chatbot incorporated with the necessary NLP techniques that helps in improving the data custodians' and data consumers' reputation. Our chatbot development follows the best practices for chatbot creation as outlined in [3]. We leverage chatbot technology to minimize the service times caused by human errors and its integration with the Blockchain technology minimizes trust bottlenecks.

3.1.3 Blockchain in Broker Systems

Several prior works address the problem of lack of trust in sharing protected healthcare data. One of the exemplar works in [32] proposes a trust-building brokering architecture that fosters patient-centric cloud eHealth services. This model seeks user feedback and enables auditability by tracking transactions through a Blockchain solution. Additionally, brokering systems with Blockchain technology can both improve the quality of patient care and reduce the cost of care with targeted safe sharing of healthcare data as shown in [6]. To overcome the limitations in a centralized architecture of health information sharing such as high dependence on network connectivity and a single point of failure, authors in [43] propose the use of a Blockchain solution. Their approach uses distributed ledger technologies to facilitate multi-site, collaborative studies in the data-intensive sciences such as genetics/genomics, and enables auditability through single institutional ethics review in their Blockchain platform. The work in [37] uses technologies such artificial intelligence, machine learning and Blockchain to enable researchers to access medical data by transforming simple facial pictures and videos into powerful sources of data via predictive analytics. Blockchain technology when integrated with an online machine learning platform as shown in [8] can further help in distribution of partial models, and design new proof-of-information algorithms.

Our HonestChain is inspired by the above works and is a platform that takes into account both the objective and subjective reputation attributes. Our reputation scheme with an automated risk assessment technique ensures that data requests comply with health information sharing standards. Through our Blockchain based platform, we incentivize the consortium of peers through rewards that use the reputation of data custodians and data consumers. This interaction between data custodians and data consumers in our HonestChain builds a trusted network of peers e.g., data custodians automate data requests via audit log notifications to data custodians, and serve data consumers with a faster data access decision process.

3.2 Solution approach of HonestChain

3.2.1 HonestChain Platform Overview

Figure 3.1 illustrates our proposed reference architecture in a consortium where our HonestChain is hosted on a cloud infrastructure that is accessible by different peers that want to leverage the service. The key component in our HonestChain is the consortium Blockchain-based trust setup built on top of our reputation scheme, and incentives for information sharing as detailed later in this section. Within this consortium, we assume that there are peers (Requesters) requesting for protected health data, and peers (Providers) providing the records from cooperating domains. Furthermore, peers in each domain are assigned with a reputation value based on their contributions to the other consortium peers. Our HonestChain rates the Requester and Provider peers using metrics such as compliance score, dataset risk, and user's feedback. Using our reputation scheme, we minimize the issues of Loss of Value and Loss of Opportunity in enabling protected data access. HonestChain platform design includes on-chain and offchain components for storage, processing and sharing of health information:

On-Chain: this component fetches and displays the details such as e.g.,`user_id`, `dataset_id`, `risk_level`, `decision`, `reputation` from the CDM and the related au-

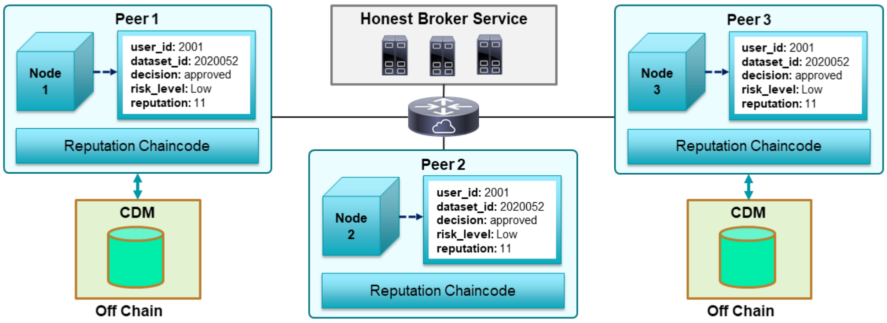

Figure 3.1: Proposed Honestchain reference architecture that features on-chain/off-chain components within a consortium of peers involving an honest broker service, dedicated blockchain nodes with Hyperledger configurations, chaincodes and CDM.

tomated honest broker services in HonestChain. These details are fed into the chaincode that helps in calculation of User and Provider Reputations.

Off-Chain: this component stores information such as the details about the domain form filled by requester, the requester details, compliance score, and dataset details. Depending on the number of requests submitted and the heterogenity of requested data, the storage of the related data will require large amounts of storage (in the order of tera bytes or even peta bytes in core network domain scenarios) and a homogenized data format. For this purpose, we utilize the CDM as an off-chain storage that interacts periodically through the related honest broker services in HonestChain. The `dataset_id` from the CDM is fetched from honest broker services and is referenced in our chaincode. This approach allows us to deliver a dynamic, standards-compliant and efficient protected data retrieval process in a peer-to-peer manner.

3.2.2 HonestChain System Roles

There are three different roles i.e., Healthcare Data Requester, Healthcare Data Provider, Healthcare Brokering System Administrator that serve as central actors in our HonestChain implementation. In the following, we provide their definitions:

Healthcare Data Requester

Requester is an actor who needs the health data and submits the requests to a consortium of peers. Once the request is identified, our honest broker service in HonestChain determines the parameters to send to the Blockchain network, and determines the Requester's reputation. The transaction gets submitted when all the parameters are sent via REST API calls and the Provider is notified.

Healthcare Data Provider

Provider is an actor who provides health records to the requests from a Requester if the decision is approved from the Blockchain chaincode. To incentivize Providers to provide high-quality services, a reputation R_p is given to the Provider so that their service is regarded as a trustworthy and reputable service.

Healthcare Brokering System Administrator

Admin is an actor who: (a) analyzes the data request, and (b) rates whether the transaction has been successful. The Admin will also analyze and monitor the submitted request data. The Admin is responsible for taking care of the 'manual approval required' decisions by passing the details of the requester and requested data to the honest broker governance process (manual or automated) for further evaluation. Admin here is also the point of contact for further assistance for the request related queries by the Requesters.

3.2.3 Reputation based Healthcare Data Brokering Protocols

User Reputation

Prior works in [1, 2, 41] determine HIPAA compliance in order to consider functionality of healthcare systems based on predefined regulations. Our proposed HonestChain solution features an automated HIPAA [27] compliance checking method to establish the trust and ensures auditability through Blockchain of data request transactions. To determine the trust of a Requester, we first calculate if the data requested by the Requester is compliant or not. This process is done by comparing the policies of requested data with the answers from the data domain request form filled by Requester. A Requester needs to fill the domain form that consists of 18 questions. Each domain fields include *Yes* (*Y*), and *No* (*N*). Additionally, each type of data (aggregated, de-identified, limited, or identified) also maintains its own policies based on the sensitivity. The data policies are also stored in the form of a list of 18 respective *categories*. Furthermore, we categorize the policies based on the Requester role that is internal and external. The data policies include 1 and 0, where 1 means compliant and 0 means not-compliant.

HonestChain retrieves the answers from the form and compares them to the policies of the requested data. The recorded comparison outcomes are assigned a compliance score as shown in Equation 3.1. The equation first computes the score and then the resulting value is normalized in the range from [1, 10] as done by Equation 3.2. We compute the risk on the basis of output of the average function as shown in Table I. This function takes as an input compliance and dataset risk. Once the average is computed, the risk levels are computed and are described as Low (L), Medium (M) and High (H) risks as shown in Equation 3.3. Our reputation scheme is based on two modules that is reputation of Requester and reputation of Provider. The reputation of the Requester is computed based on the risk levels. We describe our reputation as 0, -1 and +1 based on the risk levels as shown in Equation 3.4. The base reputation default is set to 10 for a new Requester, and this value is updated based on new requests submitted.

Table **??** shows the notations used in the remainder of this section. The Score calculation (S_i), Compliance score (C_s), Risk evaluation (R_u), and Reputation value calculation (B_w) can be given as:

Score calculation:

$$S_i = \begin{cases} 1 & \text{if } (a_i = r_i) \ or \ (a_i = 1 \ and \ r_i = 0) \\ 0 & \text{if } (a_i = 0) \ and \ (r_i = 1) \end{cases} \tag{3.1}$$

where $(0 < i < n)$, and n = number of terms in domain form.

Compliance score normalization from 1 to 10:

$$C_s = \left\lceil \left(\sum_{i=1}^{n} (S_i * 10) \right) / n \right\rceil \tag{3.2}$$

User Risk Level definition:

$$R_u = \begin{cases} L: & C_s \in [1, 3] \\ M: & C_s \in [4, 6] \\ H: & C_s \in [7, 10] \end{cases} \tag{3.3}$$

User Reputation calculation:

$$B_u = \begin{cases} -1 & \text{if } (R_u = H); \ request \ is \ denied \\ 0 & \text{if } (R_u = M); \ manually \ evaluated \\ 1 & \text{if } (R_u = L); \ request \ is \ approved \end{cases} \tag{3.4}$$

Data Provider Reputation

Chatbot guidance in the request form helps the Requesters to fill the data more accurately, which in return increases the reputation of the Requester. After the request is handled successfully and protected data access is granted, the Requester will fill out a feedback form where they subjectively give responses on Provider's performance. In this way, we account for subjective opinions in increasing Providers' reputation in addition to using objective metrics such as number of requests handled, service time per request, number of feedback received, etc. We calculate reputation of the Provider based on Equation 3.5 borrowed from a related work [4]:

$$B_p = \begin{cases} 1 & \text{if } (\textit{positive});\ \textit{trust worthy} \\ 0 & \text{if } (\textit{neutral});\ \textit{no assessment} \\ -1 & \text{if } (\textit{negative});\ \textit{not trust worthy} \end{cases} \tag{3.5}$$

In our consortium Blockchain, we ensure that the providers are incentivized to provide the accurate data and the information is shared across the trusted peers. We incorporate an optimistic approach where we provide rewards to Providers as an incentive to share the protected healthcare data. This reward is the increase in reputation that helps them by allowing more Requesters to be paired to use the Provider's data. Our reputation value calculation is given by:

$$B_p = K + \left(\sum_{i=1}^{m} (S_p) \right) \tag{3.6}$$

where: $(0 < i < m)$; K is a constant that represents the initial reputation of provider p; m is the number of feedback values received on Provider p; S_p is the value given to Provider p on each feedback.

Initially, we assign a base reputation of 10 by default that is given by constant K to the Provider. Through our reputation value calculation, Providers can increase/decrease their reputation. The determination of reputation value is given by S_p, which is obtained through a feedback form provided to the Requester after the user receives the requested data. Requester can rate the Provider and the value goes into the -1, 0 or 1 category. The overall feedback given by the Requester results in a cumulative value calculation for the Provider. In this optimistic model, our goal is to allow Provider and Requesters to gain reputation by contributing honestly and sharing the healthcare data in a trusted, automated manner via an immutable ledger.

3.2.4 Incentives for Sharing

We consider risk and reputation as the primary HonestChain factors in performing the decision making to authorize protected data access. This allows the HonestChain to rate a Requester and a Provider based on their respective historic feedback and risks. We follow a semi-legal approach including both objective and subjective ratings as detailed above, where we focus on determining the reputation of a Requester and Provider(s) based on their service performance and data request parameters. With the historic reputation information, and owing to the design of the reputation protocols in our HonestChain, we enable a trustworthy platform in HonestChain for health information sharing. A higher reputation value leads to a higher probability of a Provider peer being selected for delivering service to the Requester, and a higher reputation value leads to a higher probability of Requester peer data being approved and delivered in a fast manner in the future.

3.2.5 Exception Handling

1) Prevention against Sybil attacks: Sybil attacks occur when the attacker disguises as an authorized user and generates multiple illegitimate and fake identities in order to disrupt the functioning of the service and to take undesired control over the peers within the consortium. Our HonestChain platform allows an inbuilt trust creation through certificates. When Hyperledger Composer is deployed to the Fabric, all the Hyperledger Fabric Certificate Authority servers share the same database for keeping track of identities and certificates. The identity management here is centralized and helps in protection against Sybil attacks.

2) Replay attacks: Replay attack occurs when a user tries to delay or repeat the data transmission in a network. Our scheme is immune to replay attacks due to the incorporation of Admin authorization. It allows us to associate a certificate and a private key. The ability to deploy a network is only given to the trusted authority and furthermore, each transaction has a unique timestamp id that can be traced back to the user's identity. Additionally, due to automatic generation of a valid private key, a user without a key is unable to modify the content and the transaction will not reach consensus.

3) Authorization: In our scheme, we first authorize the Requester through our UI and then we map his/her identity to the Blockchain chaincode. Our scheme only allows the trusted authorities to execute transactions. If the connection profile matches with the peer's details, then the chaincode is executed. This mapping allows us to trace-back the real identity of Requesters to avoid potential frauds and thefts.

3.3 Performance Evaluation

3.3.1 Experiment Testbed Setup

To evaluate our Honestchain, we implemented our solution using a realistic simulation testbed on a public cloud infrastructures as shown in Figure 3.2. In this testbed setup, we included a node dedicated for the User Interface (UI). Our UI is built using the Flask framework and is integrated with a chatbot created using DialogFlow following the best practices [3]. Requester fills the protected data access request in "request form" with the guidance of the chatbot. The Blockchain implementation is hosted on an AWS EC2 instance and involves an Hyperledger Composer installation that utilizes 20 GB memory. Additionally, we utilize an AWS RDS instance for hosting the CDM service. To allow CDM service to handle large datasets, we configure 40 GB memory on the RDS. The request details are send to the HonestChain services for risk assessment calculations, and the peer reputation calculations. Based on the calculated values, HonestChain allows data custodians to automate decisions on approvals or denials of the data requests.

We considered state-of-the-art healthcare requester reputation schemes to compare performance and evaluate reputation values of our HonestChain. Specifically, we compare the performance of HonestChain with Recency-Based [15],

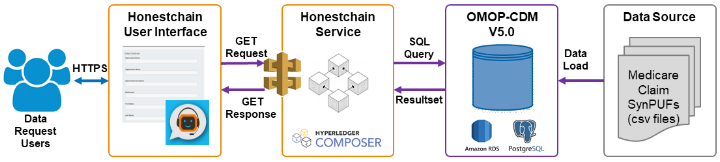

Figure 3.2: Cloud testbed used to evaluate HonestChain platform performance with experiments involving a distributed network.

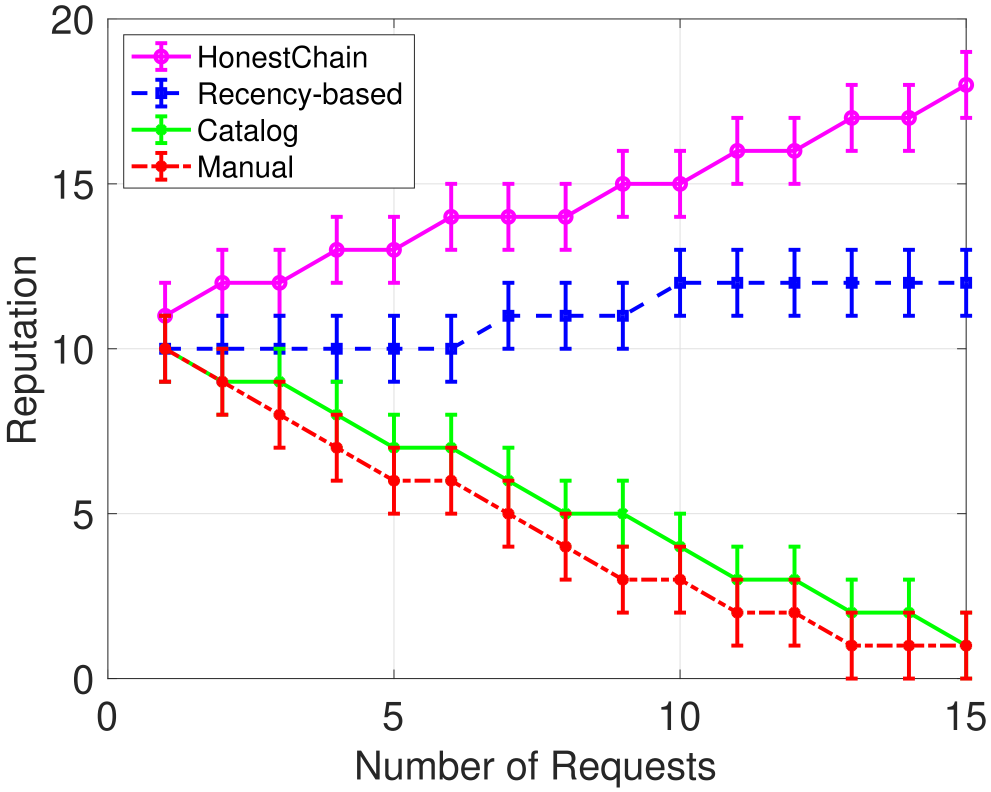

Figure 3.3: HonestChain performance comparison with state-of-the-art schemes for reputation values for requesters

Catalog [31], and Manual [7] schemes. The Recency-based scheme includes the information about the last two requests submitted, whereas the Catalog includes requests that prompt users to choose particular dataset(s) from a limited catalog list; the Manual scheme involves filling out paper forms for data request and authorization/acess processes. We choose these schemes because they help us in deriving close-to-real results based on the types of request submissions.

We performed simulation experiments by choosing a different set of Requesters and Providers over 25 iterations. The goal here was to simulate real-world situations that allow us to create a fair chance of interactions. Each Requester and

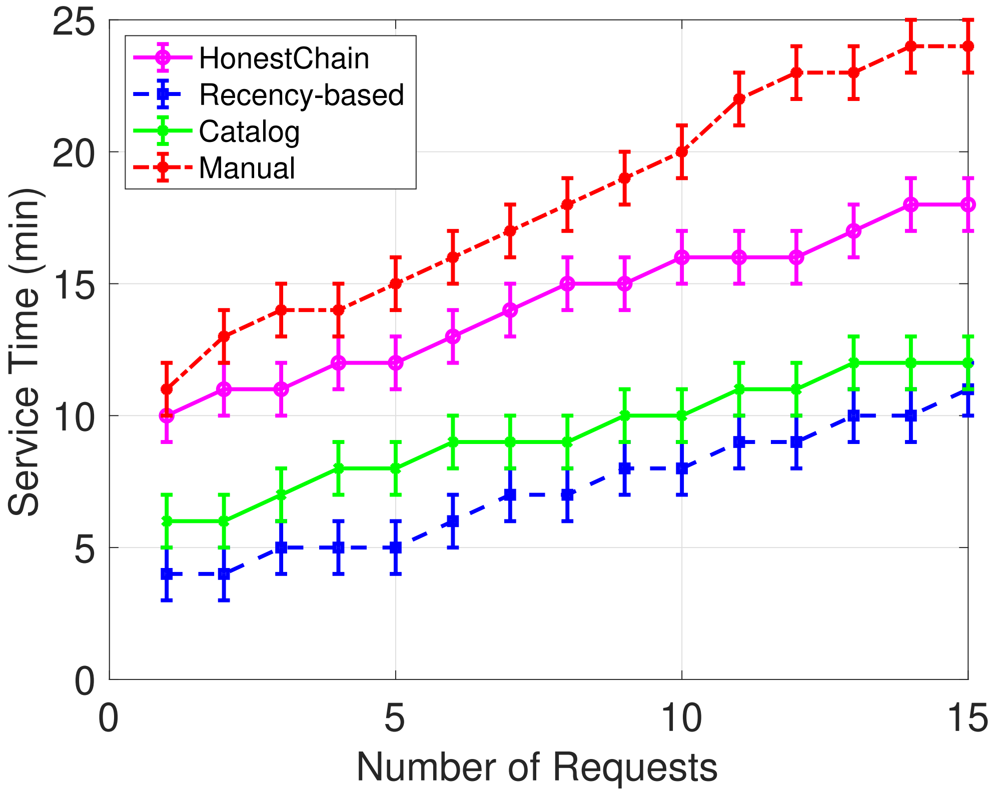

Figure 3.4: HonestChain performance comparison with state-of-the-art schemes for service times

Provider have different values of data corresponding to the dataset that they request, or the dataset that they provide. We performed simulations of the request process by taking into account parameters such as: type of dataset required, compliance score, and dataset risk levels to determine the reputation values. Each request associated with these schemes have different values of data corresponding to the data request parameters that they employ. We evaluate our HonestChain in the testbed experiments using metrics such as: reputation values, service time and request resubmission rate. Reputation value is the cumulative score assigned to the Providers/Requesters based on the approval/denial of a series of requests. Service time is the total time that our HonestChain takes to process each request. Request resubmission rate is the number of requests that are resubmitted upon denial in the previous transaction.

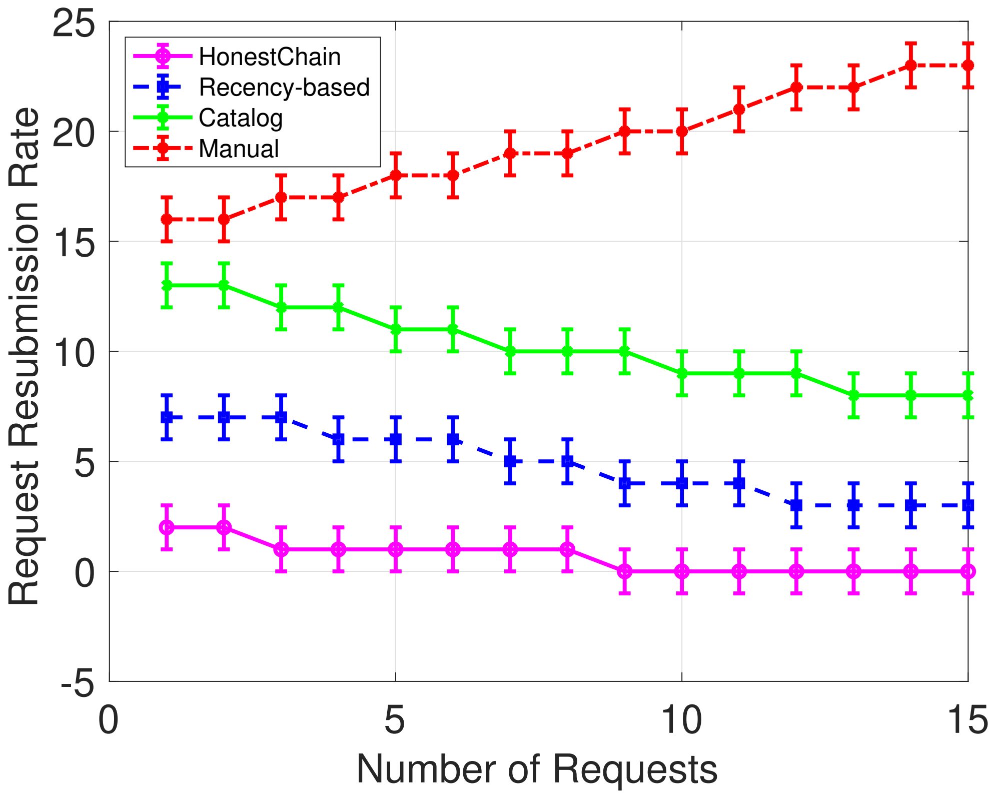

Figure 3.5: HonestChain performance comparison with state-of-the-art schemes for studying their performance trade-offs in terms of request resubmission rate

3.3.2 Reputation Scheme Results

Our first experiment was to evaluate the decision making process in pairing a Provider, when a request arrives from a Requester for a particular dataset. We simulated a total set of 20 Requesters and Providers. From this total set, we randomly generated a subset of 15 requests evaluating the various decision making schemes. With our HonestChain, the Requestors ended up pairing with Provider(s) based on their calculated Reputation values. The other schemes use different algorithms for pairing the Requesters and Providers. For instance, the Recency-based scheme will pair one Requester or Provider based on the last couple of requests, whereas the Catalog scheme will pair the Requester/Provider peers based on the request submission involving the completion of a pre-defined form (without guidance); the Manual-based scheme will pair the Requester/Provider through a long tedious form submission process. Our results show that Requesters and Providers paired by our HonestChain have better overall reputation values when compared to the state-of-the-art schemes. This can be seen in Figure 3.3.1(a), where the HonestChain's performance improvement ranges from 1.3x - 4x times higher in terms of reputation values. This improvement in reputation in our HonestChain is due to the fact that we consider a comprehensive set of parameters to determine the Reputation values, rather than pairing Requesters/Providers based on a predefined form filling criteria in Catalog or using last couple of submitted requests in Recency-based approach.

Upon choosing the Provider using our HonestChain, we analyze the performance trade-offs in the service time, with the request resubmission metric. As shown in Figures 3.3.1 b and c, HonestChain takes up to 1.5 times more service time in the worst case as compared to the Recency-based [15], Catalog [31], and Manual [7] schemes. This is due to the rigorous multiple stages involved in the HonestChain process of compliance checking, risk determination and decision making process and data retrieval process using the CDM module. However, these steps only consume a few minutes and the related overhead times can be compensated by using automation to make the risk assessment strategies more efficient.

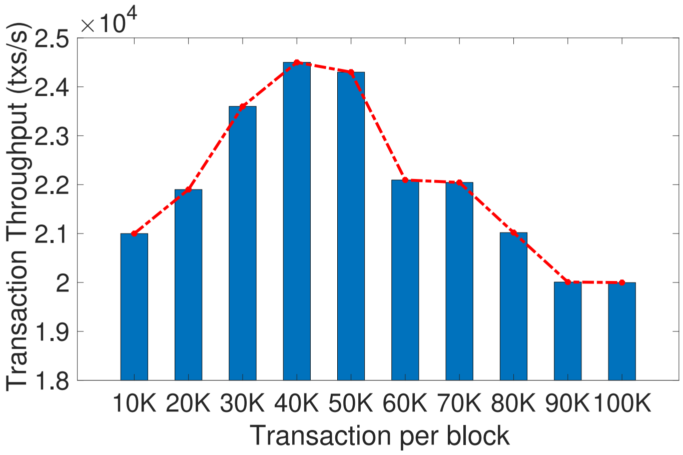

Figure 3.6: HonestChain Throughput results involving measurement of transaction rate in terms of block sizes ranging from 10,000 to 100,000.

More importantly, we should note that our HonestChain produces the least request resubmission rate compared to the other schemes. It is due to our chatbot guided interface, consoritum Blockchain architecture and the automation for efficient risk enforcement policy implementation that together ensure that protected data requests are compliant with the security standards (thus minimizing Loss of Value) and have inherent auditability that avoids manual intervention (thus reducing the chance of Loss of Opportunity). The request resubmission rate is an important measurement of the healthcare broker systems, and all Requestors will inherently get a much higher weight or reputation in real-world scenarios if their resubmission rate is low. Thus, we show that our HonestChain has much higher performance overall considering trade-offs in comparison to the state-of-the-art decision-making schemes in pairing the appropriate Requesters and Providers in a consortium. any anomalies.

3.3.3 Scalability Results

Lastly, we evaluate our HonestChain based on the scalability performance as shown in Figure 3.6. Specifically, we compare HonestChain throughput results involving measurement of transaction rate (throughput rate per second) in terms of increasing block sizes ranging from 10,000 to 100,000. For a fair comparison, we used the same transaction chaincode for all experiments. Each transaction process involves a series of independent read/write operations focusing on I/O, caching and parallelism. We get transaction throughput of 210,000 (tx/sec) for a block size of 10,000 transactions. A smaller block size of 10,000 transactions corresponds to a lower batch size as per [24], which produces a lower throughput. In contrast, a higher block size of 40,000 transactions corresponds to a higher batch size, which produces a larger throughput rate. As the block size further increases up to 40,000, we see the highest transaction throughput. When the block size goes beyond and reaches to a level of 60,000, there is a sharp decrease in the transaction throughput. This decrease can be attributed to the delay in accepting the committed transactions by the endorsing peers in the Hyperledger Fabric, and the limited number of endorsing peers to accept the consensus mechanism. We also note that this delay occurs due to the limited resources available in our cloud testbed setup to process transactions above 60,000. It is possible however to minimize this time delay by allowing participation of higher number endorser peers, and by increasing the resource availability in real-world HonestChain deployments to meet service demands. In summary, the scalability experiment results show that by achieving a highest throughput on 40,000 block size, our HonestChain platform is practical and is able to scale for a reasonably large number of protected health data requests in a consortium of Requesters and Providers.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

In this thesis, we provided two applications : DefenseChain and HonestChain. In the DefenseCahin, we developed a novel DefenseChain platform that leverages advancements in Blockchain technology for providing threat intelligence sharing platform capabilities to defend against cyber attacks such as DDoS, APTs and cryptojacking. DefenseChain can be used to perform attack detection/mitigation via threat intelligent sharing among a federation of domains using distributed trust principles. We devised novel QoD and QoM metrics to determine which Detector and Mitigator can be selected by a Requester in a trustworthy manner, based on factors such as e.g., accuracy, suspiciousness score, service time, attack type and attack reoccurence. Our consortium Blockchain reference architecture implementation allows threat data sharing before and after attacks, so that the Requester is able to request for Detector(s) and Mitigator(s) services to effectively defend targeted attacks in a timely and robust manner. Our evaluation results from a realistic experimental testbed and from simulation results show that our DefenseChain is effective in choosing Detector(s) and Mitigator(s) based on QoD and QoM values, and outperforms state-of-the-art schemes such as SocialReputation model [25] in identifying and handling rational/irrational Detectors and Mitigators within a federation of co-operating peers/domains.

In HonestChain platform, it leverages Blockchain and chatbot technologies to enable health information sharing in a secured, expedited and standards-compliant manner. Using a consortium Blockchain based approach, protected data sharing is efficiently facilitated in HonestChain by using reputation value calculations of the peers (both Requesters and Providers) and by performing risk assessment of each transaction using automation to ensure auto-assurance and auto-auditability. HonestChain platform automation of distributed trust and chatbot-based requester guidance minimizes the Loss of Value and Loss of Opportunity issues, and thus allows Providers to perform faster data decision making when processing protected data requests from Requesters. These Provider benefits for multi-source data sharing and analysis can support rapid innovations for clinical research informatics and engender next-generation decision support for researchers/clinicians in the cure of diseases.

Our evaluation results from a realistic experimental cloud testbed of a health information system show that our HonestChain is effective in increasing reputation of both Providers and Requesters. Consequently, HonestChain reduces the request re-submission rate in comparison to state-of-the-art requester reputation schemes such as Recency-based, Catalog and Manual schemes that allow for secure and speedy access to protected data for authorized Requesters. Lastly, we showed that our HonestChain is practical and scalable to handle tens of thousands of transactions per block with high-performance.

In the HonestChain, we developed a novel HonestChain platform that leverages Blockchain and chatbot technologies to enable health information sharing in a secured, expedited and standards-compliant manner. Using a consortium Blockchain based approach, protected data sharing is efficiently facilitated in HonestChain by using reputation value calculations of the peers (both Requesters and Providers) and by performing risk assessment of each transaction using automation to ensure auto-assurance and auto-auditability. HonestChain platform automation of distributed trust and chatbot-based requester guidance minimizes the Loss of Value and Loss of Opportunity issues, and thus allows Providers to perform faster data decision making when processing protected data requests from Requesters. These Provider benefits for multi-source data sharing and analysis can support rapid innovations for clinical research informatics and engender next-generation decision support for researchers/clinicians in the cure of diseases.

4.2 Future Work

The future work scope for this solution is extension of it to more usecases to allow distributed trust and information sharing. We plan to collaborate with regional network service providers to integrate our Blockchain-based solutions in their infrastructures. We will identify additional real-world scenarios where distributed trust principles can be applied to authorize access of protected threat data access to defend against targeted cyber attacks. Additionally, our work can be integrated and optimization strategies and related Blockchain policies for higher-scale workloads. Towards this aim, one can use high-performance computing back-ends in cloud platforms as well as Jupyter notebook front-ends to enable faster data analysis/visualization for the requested protected datasets.

Bibliography

[1] Community cloud architecure for salesforce health care applications.) [online]. Available at https://www.salesforce.com/products/community-cloud/faq.

[2] Ibm services: Getting your data ready for precision medicine.(2020) [online]. Available at https://www.ibm.com.

[3] Lise embley, technical writer, national center for state courts(2020) [online]. Available at https://www.ncsc.org/ /media/Files/PDF/About.

[4] Karl Aberer and Zoran Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 310–317, 2001.

[5] H Afshari and Q Peng. Challenges and solutions for location of healthcare facilities. *Industrial Engineering and Management*, 3(2):1–12, 2014.

[6] Cornelius Chidubem Agbo and Qusay H Mahmoud. Blockchain in healthcare: Opportunities, challenges, and possible solutions. *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, 15(3):82–97, 2020.

[7] Majed Al Dogether, Yahya Al Muallem, Mowafa Househ, Basema Saddik, and Mohamed Khalifa. The impact of automating laboratory request forms on the quality of healthcare services. *Journal of infection and public health*, 9(6):749–756, 2016.

[8] Abdullah Al Omar, Md Zakirul Alam Bhuiyan, Anirban Basu, Shinsaku Kiyomoto, and Mohammad Shahriar Rahman. Privacy-friendly platform for

healthcare data in cloud based on blockchain environment. *Future generation computer systems*, 95:511–521, 2019.

[9] Jean Andrian, Charles Kamhoua, Kevin Kiat, and Laurent Njilla. Cyber threat information sharing: A category-theoretic approach. In *2017 Third International Conference on Mobile and Secure Services (MobiSecServ)*, pages 1–5. IEEE, 2017.

[10] Shahriar Badsha, Iman Vakilinia, and Shamik Sengupta. Privacy preserving cyber threat information sharing and learning for cyber defense. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0708–0714. IEEE, 2019.

[11] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

[12] Mark Berman, Jeffrey S Chase, Lawrence Landweber, Akihiro Nakao, Max Ott, Dipankar Raychaudhuri, Robert Ricci, and Ivan Seskar. Geni: A federated testbed for innovative network experiments. *Computer Networks*, 61:5–23, 2014.

[13] Mark Berman, Jeffrey S Chase, Lawrence Landweber, Akihiro Nakao, Max Ott, Dipankar Raychaudhuri, Robert Ricci, and Ivan Seskar. Geni: A federated testbed for innovative network experiments. *Computer Networks*, 61:5–23, 2014.

[14] Xiaoming Bi, Wenan Tan, and Ruohui Xiao. A ddos-oriented distributed defense framework based on edge router feedbacks in autonomous systems. In *2008 International Multi-symposiums on Computer and Computational Sciences*, pages 132–135. IEEE, 2008.

[15] Lubomir D Bourdev. Autocompleting form fields based on previously entered values, March 11 2008. US Patent 7,343,551.

[16] Andrew D Boyd, Dale A Hunscher, Adam J Kramer, Charles Hosner, Paul Saxman, Brian D Athey, John F Greden, and Dan C Clauw. The "honest broker" method of integrating interdisciplinary research data. In *AMIA Annual Symposium Proceedings*, volume 2005, page 902. American Medical Informatics Association, 2005.

[17] Sotirios Brotsis, Nicholas Kolokotronis, Konstantinos Limniotis, Stavros Shiaeles, Dimitris Kavallieros, Emanuele Bellini, and Clément Pavué. Blockchain solutions for forensic evidence preservation in iot environments. In *2019 IEEE Conference on Network Softwarization (NetSoft)*, pages 110–114. IEEE, 2019.

[18] Eric W Burger, Michael D Goodman, Panos Kampanakis, and Kevin A Zhu. Taxonomy model for cyber threat intelligence information exchange technologies. In *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*, pages 51–60, 2014.

[19] Junho Choi, Chang Choi, Htet Myet Lynn, and Pankoo Kim. Ontology based apt attack behavior analysis in cloud computing. In *2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, pages 375–379. IEEE, 2015.

[20] Kyungyong Chung and Roy C Park. Chatbot-based heathcare service with a knowledge base for cloud computing. *Cluster Computing*, 22(1):1925–1937, 2019.

[21] Richard Dennis and Gareth Owen. Rep on the block: A next generation reputation system based on the blockchain. In *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 131–138. IEEE, 2015.

[22] Alex S Felmeister, Aaron J Masino, Tyler J Rivera, Adam C Resnick, and Jeffrey W Pennington. The biorepository portal toolkit: an honest brokered,

modular service oriented software tool set for biospecimen-driven translational research. *BMC genomics*, 17(4):434, 2016.

[23] Thanassis Giannetsos, Tassos Dimitriou, and Neeli R Prasad. People-centric sensing in assistive healthcare: Privacy challenges and directions. *Security and Communication Networks*, 4(11):1295–1307, 2011.

[24] Christian Gorenflo, Stephen Lee, Lukasz Golab, and Srinivasan Keshav. Fastfabric: Scaling hyperledger fabric to 20,000 transactions per second. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 455–463. IEEE, 2019.

[25] Andreas Gruhler, Bruno Rodrigues, and Burkhard Stiller. A reputation scheme for a blockchain-based network cooperative defense. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 71–79. IEEE, 2019.

[26] Andreas Gruhler, Bruno Rodrigues, and Burkhard Stiller. A reputation scheme for a blockchain-based network cooperative defense. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 71–79. IEEE, 2019.

[27] Joan Hash, Pauline Bowen, Arnold Johnson, CD Smith, and DI Steinberg. *An introductory resource guide for implementing the health insurance portability and accountability act (HIPAA) security rule*. US Department of Commerce, Technology Administration, National Institute of . . . , 2005.

[28] D. S. Hormansyah, E. Amalia, Luqman Affandi, D. Wibowo, and Indinabilah Aulia. N-gram accuracy analysis in the method of chatbot response. *International journal of engineering and technology*, 7:152, 2018.

[29] Kuan Lun Huang, Salil S Kanhere, and Wen Hu. A privacy-preserving reputation system for participatory sensing. In *37th Annual IEEE Conference on Local Computer Networks*, pages 10–18. IEEE, 2012.

[30] Xiaohong Huang, Cheng Xu, Pengfei Wang, and Hongzhe Liu. Lnsc: A security model for electric vehicle and charging pile management based on blockchain ecosystem. *IEEE Access*, 6:13565–13574, 2018.

[31] Jacqueline C Kirby, Peter Speltz, Luke V Rasmussen, Melissa Basford, Omri Gottesman, Peggy L Peissig, Jennifer A Pacheco, Gerard Tromp, Jyotishman Pathak, David S Carrell, et al. Phekb: a catalog and workflow for creating electronic phenotype algorithms for transportability. *Journal of the American Medical Informatics Association*, 23(6):1046–1052, 2016.

[32] Heba Kurdi, Shada Alsalamah, Asma Alatawi, Sara Alfaraj, Lina Altoaimy, and Syed Hassan Ahmed. Healthybroker: A trustworthy blockchain-based multi-cloud broker for patient-centered ehealth services. *Electronics*, 8(6):602, 2019.

[33] Ming Li, Jian Weng, Anjia Yang, Wei Lu, Yue Zhang, Lin Hou, Jia-Nan Liu, Yang Xiang, and Robert H Deng. Crowdbc: A blockchain-based decentralized framework for crowdsourcing. *IEEE Transactions on Parallel and Distributed Systems*, 30(6):1251–1266, 2018.

[34] Qun Lin, Hongyang Yan, Zhengan Huang, Wenbin Chen, Jian Shen, and Yi Tang. An id-based linearly homomorphic signature scheme and its application in blockchain. *IEEE Access*, 6:20632–20640, 2018.

[35] Zhaojun Lu, Wenchao Liu, Qian Wang, Gang Qu, and Zhenglin Liu. A privacy-preserving trust model based on blockchain for vanets. *IEEE Access*, 6:45655–45664, 2018.

[36] Zhaojun Lu, Qian Wang, Gang Qu, and Zhenglin Liu. Bars: a blockchain-based anonymous reputation system for trust management in vanets. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On*

Big Data Science And Engineering (TrustCom/BigDataSE), pages 98–103. IEEE, 2018.

[37] Polina Mamoshina, Lucy Ojomoko, Yury Yanovich, Alex Ostrovski, Alex Botezatu, Pavel Prikhodko, Eugene Izumchenko, Alexander Aliper, Konstantin Romantsov, Alexander Zhebrak, et al. Converging blockchain and next-generation artificial intelligence technologies to decentralize and accelerate biomedical research and healthcare. *Oncotarget*, 9(5):5665, 2018.

[38] Vasileios Mavroeidis and Siri Bromander. Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pages 91–98. IEEE, 2017.

[39] Axel Moinet, Benoît Darties, and Jean-Luc Baril. Blockchain based trust & authentication for decentralized sensor networks. *arXiv preprint arXiv:1706.01730*, 2017.

[40] Roshan Lal Neupane, Travis Neely, Nishant Chettri, Mark Vassell, Yuanxun Zhang, Prasad Calyam, and Ramakrishnan Durairajan. Dolus: cyber defense using pretense against ddos attacks in cloud platforms. In *Proceedings of the 19th International Conference on Distributed Computing and Networking*, pages 1–10, 2018.

[41] Sungyoung Oh, Jieun Cha, Myungkyu Ji, Hyekyung Kang, Seok Kim, Eun-Young Heo, Jong Han, Hyunggoo Kang, Hoseok Chae, Hee Hwang, and Sooyoung Yoo. Architecture design of healthcare software-as-a-service platform for cloud-based clinical decision support service. *Healthcare Informatics Research*, 21:102, 04 2015.

[42] Jianli Pan, Jianyu Wang, Austin Hester, Ismail Alqerm, Yuanni Liu, and Ying Zhao. Edgechain: An edge-iot framework and prototype based on blockchain and smart contracts. *IEEE Internet of Things Journal*, 6(3):4719–4732, 2018.

[43] Vaso Rahimzadeh. Ethics governance outside the box: Reimagining blockchain as a policy tool to facilitate single ethics review and data sharing for the'omics' sciences. *Blockchain in Healthcare Today*, 1:1–10, 2018.

[44] Danda B Rawat, Laurent Njilla, Kevin Kwiat, and Charles Kamhoua. ishare: Blockchain-based privacy-aware multi-agent information sharing games for cybersecurity. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, pages 425–431. IEEE, 2018.

[45] Alexander Schaub, Rémi Bazin, Omar Hasan, and Lionel Brunie. A trustless privacy-preserving reputation system. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 398–411. Springer, 2016.

[46] Yuqi Si and Chunhua Weng. An omop cdm-based relational database of clinical research eligibility criteria. *Studies in health technology and informatics*, 245:950, 2017.

[47] Gaurav Somani, Manoj Singh Gaur, Dheeraj Sanghi, Mauro Conti, and Rajkumar Buyya. Ddos attacks in cloud computing: Issues, taxonomy, and future directions. *Computer Communications*, 107:30–48, 2017.

[48] Rashid Tahir, Sultan Durrani, Faizan Ahmed, Hammas Saeed, Fareed Zaffar, and Saqib Ilyas. The browsers strike back: Countering cryptojacking and parasitic miners on the web. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 703–711. IEEE, 2019.

[49] Deepak Tosh, Shamik Sengupta, Charles Kamhoua, Kevin Kwiat, and Andrew Martin. An evolutionary game-theoretic framework for cyber-threat information sharing. In *2015 IEEE International Conference on Communications (ICC)*, pages 7341–7346. IEEE, 2015.

[50] Iman Vakilinia, Deepak K Tosh, and Shamik Sengupta. 3-way game model for privacy-preserving cybersecurity information exchange framework. In *MIL-*

COM 2017-2017 IEEE Military Communications Conference (MILCOM), pages 829–834. IEEE, 2017.

[51] Samaikya Valluripally, Murugesan Raju, Prasad Calyam, Mauro Lemus, Soumya Purohit, Abu Mosa, and Trupti Joshi. Increasing protected data accessibility for age-related cataract research using a semi-automated honest broker. *Journal for Modeling in Ophthalmology*, 2(3):115–132, 2019.

[52] Jisheng Wang, SHEN Min-Yi, Prasad Palkar, and Sriram Ramachandran. Collaborative and adaptive threat intelligence for computer security, November 5 2019. US Patent 10,469,514.

[53] Xiang Zhang, Guoliang Xue, Ruozhou Yu, Dejun Yang, and Jian Tang. Keep your promise: Mechanism design against free-riding and false-reporting in crowdsourcing. *IEEE Internet of Things Journal*, 2(6):562–572, 2015.

[54] Xiang Zhang, Guoliang Xue, Ruozhou Yu, Dejun Yang, and Jian Tang. Keep your promise: Mechanism design against free-riding and false-reporting in crowdsourcing. *IEEE Internet of Things Journal*, 2(6):562–572, 2015.

[55] Yanru Zhang, Chunxiao Jiang, Lingyang Song, Miao Pan, Zaher Dawy, and Zhu Han. Incentive mechanism for mobile crowdsourcing using an optimized tournament model. *IEEE Journal on Selected Areas in Communications*, 35(4):880–892, 2017.