ADAPTIVE SECURITY FOR IOT IN E-HEALTHCARE ENVIRONMENT USING BLOCKCHAIN

By

Zeeshan Zulkifl

A thesis submitted to the faculty of Information Security Department,
Military College of Signals, National University of Sciences and Technology,
Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS in
Information Security

September 2019

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by **Zeeshan Zulkifl Shah** Registration No. **00000240992**, of Military College of Signals has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have been also incorporated in the said thesis.

Signature: _____

Supervisor: Dr. Mehreen Afzal

Date: _____

Signature (HoD): _____

Date: _____

Signature (Dean): _____

Date: _____

ABSTRACT

Authentication, Authorization and Audit Logs are soul features of Network Security. These services are achieved in legacy systems through application of Access Control mechanisms like MAC, DAC, RBAC et cetera, coupled with Authentication mechanisms like Auth 2.0, Kerberos, LDAP and RADIUS. IoT is a fresh domain in networks which require due security considerations and these classic mechanisms are not optimized for such devices due to various aspects such as heterogeneity, resource constrained processing, storage and multiple factors. Moreover, the legacy methods discussed above are mostly centralized in nature and thus introduce a single point of failure. In this thesis, a novel approach using fuzzy logic and blockchain technology is adopted to achieve AAA services (Authentication, Authorization and Audit Logs) through utilization of computing capability of blockchain using Dapps and foolproof logs which are built-in feature of DLT (Distributed-Ledger Technology) due to their intrinsic immutable property.

In this research work we have explored Blockchain technology to its fullest, examining various Blockchain based solutions appertaining to attributes like scalability, trust, heterogeneity and resource constrained environment. Hyperledger was found to be best suited for the HealthCare environment which requires privacy as well as fast response environment. Furthermore, Adaptive security mechanism for authentication and access control are achieved through behavior driven fuzzy logic to achieve security parameters for healthcare IoTs.

DEDICATION

This thesis is dedicated to

MY FAMILY AND TEACHERS

for their love, endless support and encouragement

ACKNOWLEDGEMENTS

I am grateful to Allah, the Almighty, for His mercy and benevolence who has bestowed me with the strength and the passion to complete this thesis. Without his consent I could not have indulged myself in this task.

I am also thankful to my supervisor especially and committee members who have always guided me with their keen and useful counseling in achieving my research objectives.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

Mandatory Access Control	MAC
Discretionary Access Control	DAC
Role Based Access Control	RBAC
Lightweight Directory Access Protocol	LDAP
Remote Authentication Dial-In User Service	RADIUS
Internet of things	IoT
Authentication, Authorization and Audit Logs	AAA
Decentralized Applications	Dapps
Distributed-Ledger Technology	DLT
Electrocardiogram	ECG
Byzantine fault tolerance	BFT
Electronic Medical Records	EMR
Denial of Service	DOS
Man in the middle	MITM
Terminal Access Controller Access-Control System	TACACS
Extended Terminal Access Controller Access-Control System	XTACACS
Transmission Control Protocol	TCP
Network Access Server	NAS
Message Digest algorithm 5	MD5
Internet Engineering Task Force	IETF
Internet service providers	ISPs
Digital Subscriber line	DSL
Virtual Private Networks	VPN
User Datagram Protocol	UDP
Pseudo Random Number Generator	PNRG
Massachusetts Institute of Technology	MIT
Ticket Granting Ticket	TGT
Key Distribution Center	KDC
Ticket Granting Service	TGS
Service Principal Name	SPN

Datagram Transport Layer Security	DTLS
Reference Integrity Metrics	RIM
Artificial Intelligence	AI
Certificate Authority	CA
Peer to Peer	P2P
Requesting Entity	RE
Service Provider	SP
Identity	ID
Fuzzy Inference System	FIS
Internet Protocol	IP
Media Access Control	MAC
One Time Passcode	OTP
Multi Factor Authentication	MFA
Direct Knowledge	Kd
Experience of SP with RE	REESP
Knowledge of SP with RE	REKSP
Reputation of RE	RRE
Body Area Network	BAN
Transaction per second	tps

Chapter 1

INTRODUCTION

The world is rapidly moving towards Cyber age of 5G supplemented by AI and IoTs. Thus, entering an era of Contextual connectivity and highly personalized experiences powered by sensor data of IoTs. The IoT devices already installed across the world are projected to 75.44 billion worldwide by 2025 [1]. These devices will form the base of the "Intelligent Connectivity" [2]. However, the security of IoT is still flabby even being the most lucrative target of Hackers. The IoT based cyber-attacks have been on the rise and only during 2018 these attacks were 32.7 million [3]. The malicious actors have started exploiting the weakly implemented security controls by manufacturers and thus large number of botnets across the world are formed by IoT devices. These devices have the basic feature of internet connectivity which is not secured by the manufacturing companies thus remote login attacks are on the top of the list. Moreover, due to the limited processing capability of such devices the Access Control mechanisms are hardly implemented in such devices which further worsens the situation. According to SonicWall, Windows 10 IoT Core devices were found to vulnerable to remote command execution attacks enabling malicious actors to run arbitrary code with system privileges in dearth of authentication mechanisms by Safe Breach. Mostly only password-based authentication mechanism is used in these devices which are prone to dictionary attacks, guessing and variety of other attack vectors. The network authentication mechanisms used in these devices are Kerberos, RADIUS, OAuth 2.0, LDAP etc. All of these methods are centralized in nature and pose single point of failure. Further trust is not intrinsic in these scenarios and third-party breach can compromise the device in particular and network in general. Moreover, there have been number of Ransomware attacks across the globe in which the network was breached through a feeble secured IoT device, thus these devices not only compromise themselves but the whole network.

Blockchain is another technology which recently found its way in various aspects of human life ranging from cryptocurrencies to voting, healthcare, logistics and many more. The

biggest challenge for IoT device security was limited processing capability which can now be overcome with cutting edge blockchain technology which ensures trust through immutability and distributed environment. The basic need in security is that the device or the owner must be in complete control of it at every step of every process and this has been a challenge until now. We believe that we can deploy blockchain for IoT security management using multi-factor authentication and access control without reliance on third parties such as Certifying Authorities, OAuth 2.0, and other Security providing platforms. In this thesis, we have explored the Blockchain technology for achieving intrinsic trust and security management of IoT devices in a lightweight and distributed manner. Healthcare IoT devices being most crucial to Human life are taken as test environment for our research however this framework can be extended to other IoT deployment environments in future.

1.1 Motivation

In healthcare, amalgamation of IoTs is propitious. Let's cogitate are all IoT devices secure enough distinctively in healthcare domain? [1] This signifies the concern of IoT deployment in sensitive scenarios such as healthcare and is a major hindrance in the adoption of IoT devices. Most of these devices provide partial security with simple authentication mechanisms such as username and password, they don't have identity management and access control and most of the times are being used with default settings which have negligible security. The manufacturer companies tend to beguile security by applying these simple mechanisms which are easily compromised by even script kiddie kind of hackers. For Instance, in the recent past Baby monitors being a sensitive and private data device were being compromised by hackers on large scale [2]. The hacks included monitoring and storage of live video feeds, changing camera configurations and amending authorized user lists. This doesn't end here, sensitive machines like Cardiovascular imaging devices from Philips were found with privilege escalation and code execution bugs [3]. The IoT devices used in healthcare range from smart wearables to specific medical equipment such as ECG machines and smart ventilators. Induction of such devices is making it possible to shift healthcare services from hospital centric to Home centric thus merging the lines of Smart homes with healthcare eco systems. These modern healthcare services offer smart diagnosis, disease risk-profiling and online medical support, providing swift and intelligent healthcare services to masses. Environmen-

tal sensors, body sensors, fetal monitors, electrocardiograms, blood glucose monitors etc. collect the health-related data, which is synthesized through smart gateways and actuators and made available for decision making by medical professionals. Thus, the transaction flow starting from the sensor till medical practitioners require extensive security for this sensitive health information. Most of these occasions are critical and any dereliction will be calumnious. For example, a coma patient on smart ventilator system may lose life due to glitch in machine. Thus, securing these devices is imperative. The motivation comes in play when these devices have completely different security challenges apart from legacy devices, in terms of Heterogeneity, Computing power, Network topologies, identity management etc.

Blockchain is another promising futuristic technology which has found its way in many decentralized application domains. This technology is already revolutionizing IoT as well as healthcare by providing intrinsic trust based on anonymous yet trustful transactions. Moreover, The BFT provides fault tolerant systems and any kind of malicious transaction can be easily detected through Blockchain. Blockchain is being used for secure handling of EMR (Electronic Medical Records), Identity Management of IoT based systems, logistic and Business transactions etc. As the blockchain has capability of running across multiple platforms, it is robust against many traditional security threats and most of all it provides a mechanism of secure resource sharing thus has great potential in solving security concerns of IoT. In this Thesis, we have explored this potential of Blockchain for providing an Adaptive security framework comprising identity management, access control and adaptive authorization using chaincode/ Smart contract in e-healthcare. The aim is to design a security framework using Blockchain which is better in performance and management.

1.2 Problem Statement

The emergence of Intelligent connectivity based on 5G, Cloud computing, AI and IoT has diversified the technology balance being interweaved together. Heterogeneity is a challenge as well as vital characteristic of these technologies. Sensor networks in an e-Healthcare environment constitute of multitude of devices ranging from temperature and humidity sensors to ECG and ventilator machines, and each device data belonging to a particular patient and a particular environment. Thus, identity management and authentication are prerequisite yet

a dilemma in pretext of computation, power and other resources. The IoT devices have decentralized topologies however most of security mechanisms are of centralized architecture, thus a single point of failure through DOS on authenticating devices, MITM attacks, impersonation and other IoT specific attacks. Traditional authentication mechanisms such as passwords and tokens involve human interaction and other mechanisms requiring heavy processing. Therefore, **there is a need for a smart lightweight security solution for managing authentication in IoT devices beyond dependence on primitive methods like passwords.**

1.3 Research Objectives

The main objectives of thesis are:

- Analyzing security techniques (Identity management, authentication and access control) in healthcare IoT.
- Proposing a Blockchain-based adaptive identity management and authentication mechanism based on contextual and role-based hybrid access control for resource constrained healthcare IoT devices.
- Analysis of proposed scheme in terms of security and efficiency.

1.4 Contribution

Following are the contributions of this research work:

- This thesis discusses existing authentication mechanisms in healthcare by highlighting problems that include password-based authentication, trust issues and centralized architecture inducing single point of failure.
- This thesis presents a novel approach for Adaptive authentication and access control in healthcare environment to ensure patient privacy.
- This thesis provides analysis for applicability of proposed framework in practical environment.

1.5 Thesis Outline

The research work has been organized in following chapters:

- Chapter 1: A brief introduction is given, problem statement is highlighted, followed by motivation behind research and research objectives are explained. Furthermore, the contributions made through this research are highlighted.

- Chapter 2: A Birdseye view of existing state of the art authentication systems followed by the recent research already carried out in IoT authentication and authorization.

- Chapter 3: An Introduction to Blockchain technology, core elements of blockchain, its types and applications followed by Challenges.

- Chapter 4: This chapter starts with threat model. System model is presented followed by preliminaries and blockchain transaction flow. In the end Transaction Logic of framework is discussed.

- Chapter 5: This chapter discuss System specifications and tools used for practical implementation. Comparative analysis of our thesis work with existing research. Results based on performance parameters.

- Chapter 6: This chapter summarizes the research with conclusion drawn and provides objectives for future work.

Chapter 2

Authentication and Authorization Protocols

2.1 Introduction

In this chapter various authentication and authorization standards used in healthcare environment and their vulnerabilities are highlighted. Followed by authentication and authorization protocols for IoTs using blockchain. Various aspects of these protocols are discussed in context of utilization of blockchain, drawbacks and strong points.

2.2 AUTHENTICATION, AUTHORIZATION AND ACCOUNTING/ AUDITING (AAA)

Authentication, Authorization and Accounting is a term for security frameworks implementing access control to computer resources, enforce policy, maintaining audit Logs and Authenticating users for authorized access. These processes are considered in synergy for effective network management and security. Typically, the first one in these systems is Authentication process which identifies a user usually through username and password. These services are usually deployed in a centralized client server architecture where the server compares user provided authentication credentials with those stored in database and if they match the user is deemed as authentic. If the credentials do not match the authentication fails and no more controls applied. Next is the part where the user after identifying asks for access to certain resource, thus Authorization comes into play and it determines whether the user is authorized to perform a certain task or action according to enforced policy. Usually, Authorization occurs in context of authentication, because once a user is authenticated and provided access, it is according to the enforced policies. But sometimes a user requires additional access rights for a particular task and in such cases the Authorization mechanism is segregated from Authentication and may provide user with those special privileges for a particular session or time.

Last but not the least is the Accounting or Auditing which is used for authorization control, resource utilization monitoring, any kind of network abuse by a legitimate user and analysis of malicious activities. Thus, Auditing plays an important part in making system robust and impenetrable. Different type of AAA service mechanisms exists for network security and we discuss them one by one in succeeding paras. One thing is highlighted here that these protocols are not in particular for IoT but are used at network level for the security, the IoT are provided security through these protocols may be in terms of through gateways or the protocols such as OAuth2.0 are also discussed which are deployed directly for IoT security.

2.3 TACACS

Terminal Access Controller Access-Control System (TACACS) is a family protocols designed to handle remote authentication and other related services for network access control through a centralized server. The original TACACS protocol, introduced in 1984 for communicating with an authentication server was commonly used in older UNIX networks; its successor protocols are:

2.3.1 Extended TACACS (XTACACS)

Which is a commercial extension to TACACS by Cisco, this protocol is not compatible with original protocol. TACACS and XTACACS both use a remote access server and an authentication server which communicate with each other to decide if the user can be granted access.

2.3.2 Terminal Access Controller Access-Control System Plus (TACACS+)

is a protocol which was designed by Cisco as an open standard in 1993. It is derived from TACACS but TACACS+ is completely a different protocol that handles authentication, authorization, and accounting (AAA) services. The TACACS+ protocol provides adjustable managerial control over the AAA process. TACACS+ allows a TACACS+ detailed access control to clients allowing the Access server to respond to each component of that request. TACACS+ is TCP based and it provides security by encrypting all traffic between the NAS and the process. Encryption relies on a shared secret key known to both TACACS+ server and the client.

2.3.3 Security Analysis

TACACS+ protocol lacks modern day security mechanisms. MD5-based encryption support even fails to provide any sort of transport integrity, which presents following risks:

- Accounting information can be rendered discrepant and unreliable through the man-in-the-middle attack, for auditing purposes.

- Header fields are not encrypted and are subject to man-in-the-middle attack. Man-in-the-middle attacker can manipulate header fields at known by false insertion to taint the authentication or authorization checks even though encrypted.Even though the protocol provides some kind of encryption privacy through MD5 still following attacks are very much possible:

- Brute force attacks can be used to exploit the increased efficiency of MD5 digest computation.

- Known plaintext attacks can be used to decrease the cost of brute force attacks.Known plaintext imply that an attacker would know with certitude that which is the target of the attack. In sequence to the known plaintext attack, the attacker can further determine with confidence the value of the octet used to conceal the original octet. Likewise, Chosen plaintext attacks may be used to decrease the cost of a brute force attacks as well.

- There is no forward secrecy mechanism.

Attackers who can guess or crack md5 can gain unbounded and undiscovered access to all TACACS+ traffic. The security hazard of such attack succeeding against a centralized AAA system like TACACS+ cannot be overruled.Due to type of symmetric encryption used in TACACS+ there is a possibility of attack on the protocol itself. The server must be able to differentiate between known and unknown client request which it fails to and due to this the protocol is exposed to remote brute force attacks.This protocol requires other solutions such as VPNs to be deployed along with protocol in order to secure the channel otherwise the session as well protocol itself can be hijacked as discussed above.

2.4 RADIUS

Remote Authentication Dial-In User Service is a network security protocol, which provides centralized services for users connecting through it for network services like TACACS+. RADIUS was developed in 1991 as a server AAA protocol and brought under umbrella of Internet Engineering Task Force (IETF) standards.

Due to its extensive support and pervasive features of the RADIUS protocol, it has found wide acceptance by ISPs and corporate ventures to manage access to the Internet as well as internal networks, wifi networks, and official e-mail services. These applications include digital subscriber line (DSL), modems, network ports, web servers and access points etc.

RADIUS is an application layer protocol based on client/ server architecture, and compatible with both TCP and UDP at transport layer. NAS (Network Access Server) and gateways controlling access to a network usually act as a RADIUS client to communicate with the RADIUS server. RADIUS is generally the most popular choice for 802.1X authentication. The RADIUS server runs as a background process on a server most of the times.

2.4.1 Vulnerabilities

RADIUS is a widely used protocol which works across many platforms and devices across the globe, this poses a double edge weapon as a little short coming will affect thousands of devices and it's difficult to remove core flaws as this will lend into compatibility issues of many devices. There are some fundamental flaws found in RADIUS which may allow an attacker to compromise the integrity of a transaction as well as system. Mainly, the User-Password authentication mechanism is inherently delphic, due to improper deployment of encryption and cryptographic techniques. The response authenticator conept in RADIUS is productive, but the implementation is weakly designed. The Access-Request packet is not authenticated by any machine involved in the transaction. The request authenticators is not really random enough. And finally, the shared secret is a historical method of securing RADIUS client-to-server transactions. Following are the attacks possible on RADIUS [4] which are summarized for consumption:

- **Response Authenticator Attack**

The Response Authenticator utilizes MD5-based hash. The shared secret is vulnerable to offline attack where the attacker closely observes an Access-Request|Access-Reject|Access-Accept packet sequence and as the majority of information is unobscured therefore, attacker is able to compute MD5 hash for (Code+ID+Length+RequestAuth+Attributes), and he can follow the same for guessing every shared secret.

- **Attribute-Based Shared Secret Attack**

  If authentication attempts are monitored the attacker can guess the shared secret and as the request authenticator is also known the brute force dictionary attacks are very much possible.

- **Password Guessing Attack**

  As to previous the attackers can launch exhaustive online search-based password guessing attack because there are no authentication limits set in the protocol. Thus, limit authentication requests can cater for this attack easily.

- **Attacks based on Request Authenticator**

  As discussed earlier the request authenticator is poorly generated thus resulting compromise of protocol. If a strong PRNG is used for random number generation with a long cycle it can eradicate many security vulnerabilities in RADIUS.

- **Replay Attack based on Server Response**

  The adversary can fake server responses by network traffic. When she finds a request with a matching Request Authenticator to the adversariesâĂŹ dictionary, she can masquerade as the server and replay the previously captured server response. Similarly, she can replay the Access-Accept server response by changing few parameters and successfully authenticate to the client with invalid credentials.

- **Shared Secret Issues**

  The RADIUS standard allows the use of the same shared secret to connect to several machines thus a malicious device can compromise whole network. Shared secret value of each client must be different to make it secure.

2.5 Kerberos

It is a network authentication protocol devised by Massachusetts Institute of Technology (MIT) for provision of encryption-based client/server authentication. As we have seen the protocols only relying on username-password based authentication are prone to sniffing, brute-force and password guessing attacks. Furthermore, trust is anchored to client to be honest about its identity, thus these protocols donâĂŹt cater for insider attackers. Kerberos is one of the earliest protocols which utilized cryptography for client identification and authentication over an insecure channel. Kerberos authentication takes place in following manner [5]:

- An authentication ticket which is known as Ticket Granting Ticket (TGT) is requested by client from the Key Distribution Center (KDC).
- The credentials are verified by KDC which responds with an encrypted TGT and session key after verification.
- The TGT is encrypted by KDC using the Ticket Granting Service (TGS) secret key.
- The client stores the TGT until its expired and a new TGT can be requested by local session manager.
- When client needs to access a certain network service or resource it provides current TGT to the TGS along with the Service Principal Name (SPN) of the resource.
- TGT of the user is verified by KDC after that the user is granted access to the service.
- A valid session key is provided by TGS to the client for that particular service.
- Client provides the session key to the service provider, the service provider grants access according to assigned privileges.

2.5.1 Vulnerabilities

- Hackers have introduced a way known as "Pass-the-ticket" in which they forge a session key and presenting that forgery to the resource as credentials.

- "Golden Ticket" which grants a user with KRBTGT account privileges which have access to encrypting all authentication tokens for domain controller thus granting the hacker with access to all services of the network.

- Kerberos allows some Operating system level services to log in without double checking their credentials so if a hacker is able to crack a user account and uses it to generate authentication tokens they are known as "Silver Ticket".

- Brute force attack using third party software through automated attempts to guess a password.

- A malware that can bypass Kerberos by downgrading encryption with a Skelton key, but the attack must have Admin access

- An attack known as "DC Shadow attack", where attackers gain enough access inside a network to set up their own DC to use in further infiltration.

2.6 OAuth 2.0

In legacy client-server authentication schemes, the client is granted access to a protected resource after authentication with the server providing the resource owner's credentials. For provision of access to third-party applications to protected resources, the resource owner has to share its credentials with the third party, compromising security. OAuth was designed to address issues related to authorization layer to separate the role of the client and resource owner. In OAuth, the client requests access to protected resources of the resource owner which are hosted by the resource server, accesses the request using different set of credentials apart from resource owners. The client is issued an access token with attributes indicating type of access, its scope, duration and others. The Access tokens are issued by an authorization server upon approval of the resource owner. The client than acquires access using the access token resources owned by respective resource owner and hosted by the resource server [6].

2.7 Existing Proposed Protocols in IoT

Communication protocols have been given more focus in IoTs than the security protocols. Thus, these devices are lagging behind in security standards and are vulnerable to wide

variety of threats. As a consequence, there is no framework defines for IoT and Identity management, Authentication and Authorization are at risk. Ownership and identity relationships are closely related to the authentication and authorization management of IoT [7]. The owner of a device might change over its lifecycle and may be asked for authentication. Moreover, the data collected by a device need proper authorization mechanism in order to ensure privacy and traceability. The classical authentication mechanisms like passwords are no more effective and most of the devices are compromised due to folk model implementation of security in these devices by manufacturers. As discussed earlier no standard security protocol exists for IoTs, hence, a number of proposed authentication and authorization protocols exist. These protocols have been generalized on the basis of centralized vs decentralized architecture in following paragraphs.

Many IoT based authentication protocols have been designed but a few exist which are specifically designed for e-healthcare based IoTs [8] [9]. Jiang *et al.* [10] improved password-based authentication work of [11] both protocols rely on password based authentication which is vulnerable to guessing attacks and weak password vulnerability. Ferag *et al.* [12] has carried out a comprehensive survey of around 40 authentication protocols designed for IoT. These protocols mostly cater for a specific attack in IoT domain and does not provide a comprehensive solution. Due to Ubiquitous, heterogeneous nature of IoT, access control and identity management are a major concern. Riveria *et al.* [13] used OAuth 2.0 to define an access control model for IoT, the drawback of this model is it relies on third-party services and centralized architecture. Significant work exists on Authentication mechanism and access control but few of the approaches utilize both in their approach [14] [15] [16]. Identity based access control models have a central identity server or a trust server to manage the access control [17] [18] [19]. These servers if compromised are single point of failure. DTLS is also used to achieve security in IoTs [20] [21] [22] [23] but these lack dynamic access control and adaptive authentication and are resource heavy.

Blockchain has some security features by design which can be utilized for achieving overall security for different systems [24] [25]. Zyskind *et al.* [26] used Blockchain to ensure privacy of user but only utilized blockchain for storing access control information

thus wasted the true potential of blockchain. Similarly, Dorri *et al.* [27] utilized Blockchain to store access control policies to achieve immutability and distributed property but does not apply identity management and authentication mechanisms. Furthermore, their approach underutilizes blockchains computational capacity. Ouddah *et al.* [28] utilized true computational potential of blockchain to achieve decentralized access control. They used access tokens for giving access rights from one peer to other through transactions. The access control policy is part of locking script which has to be unlocked by possessor to prove he has the token. The computational capability of locking script is limited than the Smart contract thus this model is less efficient. Zhang *et al.* [29] utilized smart contract which is feature of Ethreum Blockchain for access control in IoTs. Their architecture is designed around gateways and thus gateways are assumed as a trusted entity and not truly verified. Ramchandaran *et al.* [29] also utilized smart contract for access control but they only stored access control policies, time of day, signature of last change and logs etc. Qu *et al.* [31] used Blockchain to verify credibility of an IoT device. The model uses gateway as a trusted entity for connected IoTs. Azaria *et al.* [32] utilized Blockchain to access, store and modify health records. There model only ensures security of health-related data instead of system. These approaches use Proof of work consensus model is which has inherited 51% problem which makes system vulnerable [29] [30] [32]. Lee *et al.* [33] implemented Zero-knowledge proof on authentication server to protect data of smart meter stored on Blockchain. Uses primitive method of username password-based authentication which necessitates the use of authenticating server which introduces a single point of failure. Banerjee *et al.* [34] has suggested a blockchain based solution for compromised firmware detection and self-healing. They stored the Reference Integrity Metrics (RIM) on the blockchain to ensure its integrity. Huh *et al.* [35] proposed a blockchain based IoT management system which manages the electricity usage of a smart meter by implementing Ethereum smart contract. Different Blockchain solutions [36] [37] [38] were analyzed based on security features, scalability and compatibility and Hyperledger was found best suitable for healthcare domain being consortium blockchain ensuring privacy as well as scalability and compatibility with other systems.

For IoT and efficient mechanism is required for authentication and authorization based on trust as many devices work mutually and if a single device acts maliciously it can com-

promise the whole network. Fuzzy logic-based systems can quantify trust uncertainty in a better way and can be utilized for malicious behavior detection [39]. Mahalle *et al.* [40] has utilized fuzzy logic for access control in IoT but their approach is centralized in nature and introduces a single point of failure in the system. Furthermore, their approach has scalability issues as all trust logic is centrally located. [41] has generalized the idea of risk-based authentication and emphasized upon its application in IoT domain. Thus, risk-based authentication forms basis of our concept for adaptive security implying the same concept for trust driving and access control.

2.8 Conclusion

This chapter highlighted AAA based standards currently in use for healthcare environment. These protocols are centralized in nature and involve single point of failure. These protocols were not designed for IoT environment thus involve various vulnerabilities discussed. Then Blockchain based proposed solutions by various researchers are discussed and analyzed. Most of the work involve public blockchains which are resource heavy and have privacy issues and only few utilize the true potential of blockchain technology.

Chapter 3

BLOCKCHAIN

3.1 Introduction

This chapter gives broad view of blockchain technology. Five core elements of blockchain are discussed followed by the various types of blockchain networks. A brief overview of smart contracts designed by Ethereum for blockchain technology is given. In the end challenges related to blockchain technology are highlighted.

3.2 Background

The Internet of 90s was never thought of being so gigantic to transform the world into a global village. Whereas the internet of now is ever changing the human history with emergency of new technologies leveraging human device interaction and transforming this world into a digital one. This paradigm shift has incorporated multiple technologies and Blockchain is one such technology recently joining the list. The internet has facilitated the business transactions but the most crucial deterrent has been "Trust". The internet was not found reliable for carrying out financial transactions securely and this is what stopped the invention of digital currencies. The digital transactions constitute a vast landscape with a potential for all sorts of communications raging from IoT devices interacting in smart cities, financial transaction with palm devices without physical interaction to home centric based healthcare services. This all has been made possible due to advent of AI, IoT and Cloud computing. So far, the internet has been relying over the third-party trust anchors such as Certificate Authorities. However, these central authorities are also prone to cyber-attacks and can be easily compromised and once an attack is successful the effects are devastating [42].

Satoshi Nakamoto presented a solution for maintain trust in a decentralized way [43]. He presented a financial model in which trust is derived from transactional cost and thus rendering any kind of fraud useless for the participants. Thus, cost and requisite trust of

central organization can be minimized or abolished by using blockchain which is a truly decentralized trust mechanism. The efforts for decentralized currency date back to 1980s [44]. Some digital currency models were proposed which included B money [45], Bitgold [46] and Karma [45] but all of these utilized concepts of crypto puzzle for generation of digital currency but were still dependent on a central entity for upkeep of possession records. Thus, Bitcoin concept constituting the first truly decentralized crypto currency was introduced by Satoshi Nakamoto [43] in 2008.

The Bitcoin is built over a core concept given by Satoshi himself and this concept is Blockchain Technology. Blockchain technology helps in validation, maintenance and tracking of all digital transactions in a distributed manner. We can say that Bitcoin is the first use case application of Blockchain technology. Therefore, in order to understand the mechanism of blockchain, understanding bitcoin is crucial.

3.3 Core Elements of Blockchain

The first question which comes in mind is "What is Bitcoin??". The creator of Bitcoin Satoshi has defined Bitcoin in his white paper [43] as
"A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based Proof-of-Work, forming a record that cannot be changed without redoing the Proof-of-Work."

In order to understand the blockchain and bitcoin we need to dissect this statement of Satoshi and this will help us understand the core elements of Blockchain and Bitcoin. Satoshi placed great emphasis on "Peer to peer networks". So first of all, we define what is a peer to peer network.

3.3.1 Peer to Peer Network

A group of computers where each computer acts as node which behaves as a client and as a server at the same time. Instead of a central server to act as a network shared drive or resource every participant node is a resource for network. When an internet based P2P network is established there is either a central server which maintains file index or a distributed network can be established where the file sharing is responsibility of each peer node [47]. The Bitcoin is also based on the later definition where each node is a resource and every peer in the network is responsible for keeping the record of transaction which in case of Bitcoin is related to Crypto Currency called **Bitcoin**. The biggest challenge for P2P network based currency is double spending problem which means over the internet it is a problem to share a digital asset with someone without involvement of a trusted third party which is mostly a central entity like a bank or government which keeps the record of your transaction so that you do not send the same asset to someone else. This is known as Double spending problem. The Bitcoin has utilized Blockchain in order to cater for this double spending problem. The P2P model is compared with client- server model in table 3.1 [48].

Table 3.1: P2P model vs Client-Server Model

	P2P Network	**Client-Server Model**
Main	No Client or Server each node has same role and all share resources	Server is a specific node provides resources and clients don't share resources, use server's resources
Service	Each node can request as well as provide services	Server is responsible for providing services
Data	Each device has its own data	Data is located on server
Bottleneck	There is no bottleneck as services are provided by every node	Servers are single point of failure and are a bottleneck for network
Cost	P2P model is less expensive to implement	Client Server architecture is expensive to implement
Stability	P2P networks are less stable because if a node is offline the specific resource on that node is not available to others	Server - Client systems are more stable as the servers are always online to provide a specific resource to its clients
Scalability	As the number of peer nodes increase the performance of network reduces therefore it is less scalable.	The client server architecture is easily scalable.

There are two types of nodes in blockchain P2P network. The type is based on the processing and storage capability of the node and both are defined below:

1) **Main Node**

The main node are fully capable nodes which use bottom-up approach for verification of transactions. These nodes can track a coin since its creation till the last transaction taken place related to that coin. They start from the genesis block and goes up to the last block in which latest transaction took place. This requires certainly more processing power to calculate and verify hashes and to maintain a large database a big storage capacity is needed. Currently the Bitcoin data is approx. 160 Gigabytes [49].

2) **SPV Nodes**

These are lightweight nodes; they only store a part of blockchain and use top down approach for verification. They can keep only block headers to verify a transaction thus requiring less space as compared to main nodes. In order to verify a transaction, they locate it in block and then ensure that at least 6 blocks have been built on top of that transaction in order to deem it as verified.

3.3.2 Time Stamps

Blockchain utilize time stamps to avoid double spending. All the transactions of Bitcoin are recorded on shared ledger (Blockchain Database) maintained by each node creating a log. When a person spends Bitcoin for example Bob send 2 Bitcoins to Alice, the timestamp is recorded on Blockchain and this timestamp is recorded on shared ledger which is maintained by everyone on the network. Thus, afterwards if Bob wants to send same 2 Bitcoins to Charlie he cannot because all peers will detect double spending by bob through the log everyone maintain. The mechanism through which all nodes agree upon a certain transaction is discussed later on.

3.3.3 Consensus Mechanism

Above solution seems easy but there is still a problem. What if all the network has not reached a consensus at certain point of time when Bob initiates the second transaction with same 2 Bitcoins. In order to detect this Satoshi cleverly designed a game theory-based Consensus algorithm. In order to reach consensus, the Bitcoin and all other blockchain based solutions are based on Byzantine General's Problem.

a. **Byzantine General's Problem**

In Byzantine empire during war a problem emerged, the army was deployed across different cities and the war was going in each city at the same time. The king came to know that there was a traitor among his generals which all were fighting in different cities. The king didn't know which general is the traitor and he wanted to change the tactical plan. The only way to communicate with generals was through messages. Thus, traitor general can send a misleading message to others and they cannot reach consensus on a plan. They need to come up with a solution to solve this issue and to reach consensus for the tactical plan. This is Byzantine General's Problem. In computer systems, **A system must be able to function properly in the presence of faulty components that may send conflicting information to different parts of the system.** This matter is even more intense in decentralized computer networks.

b. **Byzantine Fault Tolerance Algorithm**

The solution to the Byzantine General's Problem is known as Byzantine Fault Tolerance Algorithm. We will discuss Byzantine fault tolerance with reference to Bitcoin here and later on the technical details of the algorithm will be discussed in succeeding sections. Nodes are the network entities which can create, store, send and receive data and these are known as "Miners" in a blockchain network. From understanding perspective these nodes are the Byzantine army's Generals. The nodes start with an assumption that everyone on the network is not a trusted entity. Processing nodes treat the longest history of Blocks as the trusted history. This chain of blocks is called Main chain. Multiple computers can be simultaneously working to find out a block, the decision that which block is the right one to be part of the main chain is reached through consensus. If the block under consideration is n+1 than the decision that which version of n+1 block will be part of main chain is made by winning the n+2 block [50]. However, there might be conflicting versions of n+2 as well. But according to Satoshi it is highly unlikely that such instance exists for more than 2-3 blocks. Therefore, the bitcoin users wait for some number of blocks to be committed to chain so that are ensured that their transaction is successful. In case of Bitcoin usually the number of blocks is 6 but it may vary depending on network conditions and deployment scenarios.

c. **Consensus Algorithms**

Mainly there are three consensus algorithms although hundreds of variations exist. We will discuss the key differences between these three as shown in table 3.2 without going in details.

Table 3.2:

	PoW	PoS	PBFT
Winning Block	Miners solve cryptographic puzzle to win Block mining	Relies on the stake of person on the network in terms of assets PoS Algo randomly selects a validator for block creation like lottery	Selected Endorsers vote for state changes
Energy Consumption	High	High	Low
Blockchain	Public	Public	Private
51% Attack	Possible	Possible	Not Possible
Transaction rate	Low	Low	High
Scalability	High	High	Low

3.3.4 Cryptography

It is the application of techniques for secure communication in presence of a third party known as "Adversary". The cryptographic techniques used for securing Bitcoin are as follows:

a. **Asymmetric encryption**

It is type of Cryptography in which Key used for encryption is different from the key used for decryption. The key which is usually used for encryption is known as Public key because anyone can encrypt data intended for the user say Alice and send it to her. Alice uses a private key to decrypt data and the key is only known to Alice as the name suggests. Now in case of Bitcoin the Public key is used for two things the sender encrypts the transaction with receiver's public key and the receivers public address which is just like a bank account number is also derived from the public key. Thus, public key can be shared with anyone from whom transaction is expected. But the private key must be kept secret at all costs because if compromised the user will lose all his funds. For the purpose

hardware and software wallets are used and more secure way is to save the key in some external device or write it down on a piece of paper and keep it safe.

b. **Hash Functions**

Hashing is the technique where an input string of arbitrary length is taken and converted into an output string of a fixed length. In Bitcoin, the transaction is taken as input of hashing algorithm which is SHA-256 in case of Bitcoin and is converted in to a fixed length string of 256-bits. But in addition to this a cryptographic hash function must satisfy following properties [51]:

- **Property 1 - Deterministic**

  Regardless of number of times a string is parsed as input to a hashing function, the output is always the same. This is important because it would be useless if the result was different each time thus rendering tracking to be impossible.

- **Property 2 - Quick Computation**

  The processing time of a hash function must be as minimum as possible in order to be efficient.

- **Property 3 - Pre-Image Resistance**

  Pre-Image Resistance is stated as "If H(A) is given, it is infeasible to determine A where A is the input to Hash function and H(A) is the Output of the Hash function."

- **Property 4 - Avalanche Effect**

  If make a small change in the input string the output will change dramatically this is known as Avalanche Effect.

- **Property 5 - Collision Resistant**

  If two hashes H(A) and H(B) are given for two inputs A and B, it is infeasible for H(A) to be equal to H(B) meaning by the hash for both inputs is same. This property makes the hash of each input unique from others and this is very useful property especially in light of its use in Blockchain technology.

c. **Digital Signatures**

These are the core cryptographic components of Blockchain just like Hash functions.

These provide the non-repudiation as well as integrity for the transaction/ message generated by a party. The digital signatures are generated from the private key of the person initiating the transaction. The senders private key is taken as input along with message and passed through digital signature algorithm, thus producing a unique signature for every transaction. The receiving party can decrypt the message and generate a hash if the hash of both is same the message as well as the sender are authentic.

d. **Nonce and Difficulty**

Nonce is a random variable which provides randomness to a process to make it unpredictable. In Bitcoin protocol, all parameters are known except nonce, it is the missing piece of puzzle and in proof of work concept a miner needs to find the nonce and calculate its hash which is the block header hash. Here comes the Difficulty part, Difficult is a number set be the Bitcoin protocol and the nonce hash found by the miner must be less than the difficulty. The miner who finds this block header hash first wins the block.

3.3.5 Software Code Base

The fifth and the most important element of Blockchain is its software code which translates all the protocols and mechanisms defined in Bitcoin into an application. Satoshi Nakamoto developed the initial code himself and handed it over to software development community later on. It is an open-source project thus anyone can view the code and contribute to it. Bitcoin core is the main open source software that powers the bitcoin system. It is the reference point for implementation of other applications related to Bitcoin and thus defines the core elements of Bitcoin. Other applications developed over time to run with Bitcoin are as following:

a. **Wallets**

Wallets are just like real life wallets but instead of paper money they contain the cryptographic keys in order to make bitcoin transactions. These are used for storing funds and making transactions thus we can say these are the gateway to Bitcoin just like web browsers are gateways to internet. There are two types of wallets basically:

i. **Software Wallets**

The software wallets are applications which store the key information securely on

either your desktop or mobile and they can be browser-based wallets in which case the information is stored online in a third-party server and can be accessed through browsers just like using email account services managed by google on their servers.

ii. **Hardware Wallets**

These are specially designed hardware which can store cryptographic keys securely and they are similar in appearance to USB drives. Another primitive and effective solution is to write down the keys on a piece of paper and then keep the paper secure, because if the paper goes into wrong hands your key is compromised.

b. **Blockchain Explorer**

The blockchain explorer is just like a web search engine which is specifically developed for blockchain. These search applications allow users to search for a particular transaction, block and an address with its balance amount.

c. **Bitcoin Scripting Language**

Bitcoin being digital currency is programmable and the language at the back-end is a high-level programming language known as "Script". It is a scripting language just like python and JavaScript but domain specific meaning by it has been deliberately kept limited in functionality in order to be executable on lightweight devices as well. It is not a Turing-complete language meaning by it cannot be used for any kind of problem solving and is limited to only bitcoin related functionalities which include following steps:

i. Sender digitally signs the transaction using his private key, the coins associated with transaction are locked by script and these are called "Transaction Inputs".

ii. These coins can be unlocked only by the private key of addressed recipient.

iii. The transaction output which is coins in case of bitcoin are transferred to recipient account on once script is successfully unlocked. Thus, locking and unlocking script are two most important parts of a bitcoin transaction.

3.4 Blockchain Classification

Blockchain technology is under incubation and has not yet been standardized. Therefore, multitude of categorizations exist. However, ISO formed a technical committee in 2016

at Sydney which has been given task of standardizing Blockchain and Distributed Ledger technology (DLT) [52]. There are interoperability issues between different blockchain technologies such as Hyperledger, R3's Corda and Ethereum specifically related to the way they define permissions in their architectures. ISO plans to release the standard no later than 2021 [53]. Before, Ethereum blockchain was pretty much one of a kind of technology but after Ethereum and Hyperledger jumped into blockchain race new types of blockchain emerged. Based on the participants and Consensus mechanism we can classify blockchain into following types [54]:

3.4.1 Public Blockchain

These are the most common blockchain protocols based on Proof of Work consensus algorithm and anyone can participate without permission in these just like accessing internet. Anyone can generate a transaction, can validate one, can verify and see the whole chain through Block explorer. The transactions on these blockchains are transparent but anonymous or pseudonymous. Examples are Bitcoin, Ethereum, Monero, Dash, Litecoin etc.

3.4.2 Consortium or Federated Blockchain

These are unlike public blockchain and not everyone can participate in these blockchains. These Blockchains are maintained by certain organizations collaborating together for business or other functional matters like Banking sector interoperate and they have formed a private blockchain which is interoperable and is known as R3. Preselected nodes participate in consensus. Some people have argued over Consortium blockchain as being not fully distributed hence not to be categorized as blockchain but since no standard exists, they have been termed as a separate category. These Blockchains are faster, highly scalable and ensure privacy to an extent. Other examples are Hyperledger, EWF, Corda and B3i.

3.4.3 Private Blockchain

One organization is in complete control of private blockchain and write permissions are kept centralized within the organization. Whereas, public or a group of people outside organization may be given the read permissions. This kind of blockchain is mostly used for database management, auditing, etc. which are mostly internal matters of the organization. Transactions are verified internally hence the internal threat exists as compared to game theory-based incentive mechanisms. But the private chains are also introducing incentives based on the

conduct of participants in order to cater for internal threats. Table 3.2 shows a broad outline of difference between all three types of blockchain.

Table 3.3: Public vs Consortium vs Private Blockchain

	Public Decentralized management	**Consortium** Numerous Organizations	**Private** Single Organization
Consensus Mechanisms	**Proof of Work, Proof of Stake, etc..** • Heavy • Slow • Large energy consumption • No finality • 51% attack	**Voting or multi-party consensus algorithm** • Lightweight • Fast • Low energy consumption • Enable finality	**Voting or multi-party consensus algorithm** • Lightweight • Fast • Low energy consumption • Enable finality
Participants	**Permissionless** • Anonymous • Could be malicious	**Permissioned** • Identified • Trusted	**Permissioned** • Identified • Trusted
Transaction Approval Freq.	**Long** Bitcoin: 10 min or more	**Short** 100x ms	**Short** 100x ms

3.5 Ethereum

Bitcoin sets the baseline for the next generation of blockchain which are discussed in succeeding paragraphs. Ethereum is one such blockchain and it took the cradle from bitcoin in this race. It is the first Turing-Complete blockchain platform. Bitcoin brought a snowball effect in technology and various developers started experimenting with the bitcoin code and in process created other crypto based currencies known as "Alt-Coins".

Ethereum represents the second generation of Blockchain technology with augmented functionalities going beyond digital currency, deep diving in the world of decentralized blockchain applications knows as **DAPPS** in shorts form. This has been made possible due to decentralized computing innovation by Ethereum implemented through **Ethereum Virtual Machine**. This has been made possible due to Turing-complete language of Ethereum which can be used to program or run any kind of function or task. This is the reason behind most of the DAPPS available in market using Ethereum at backend. Ethereum proffers the tools

enabling the creation of distinct digital assets (such as commodities, gold, real estate), financial instruments (bonds, Currency), and decentralized applications (Health records, deeds, Documents). All these can be represented by a special token, and stored or transacted on a distributed ledger maintained by blockchain. The Blockchain elements as discussed in Bitcoin are same except the software code. Here we discuss the innovative Smart Contract mechanism enabled by Ethereum.

3.5.1 Smart Contracts

Smart Contracts are programs set up and executed on a blockchain. These programs can be used to connote triggers, conditions, and business logic enabling complex programmable transactions to be executed on blockchain. Solidity is high level programming language used by smart contract developers in Ethereum blockchain, it is compiled into a low-level stack-based bytecode language being run one Ethereum Virtual Machine (EVM) which is part of every node in the Ethereum blockchain network. For consistency across the blockchain, EVM code is designed to be executed deterministically. Smart contracts are deployed with the help of a contract creation transaction. The input data of the transaction contains the object code of the smart contract. The signature of the transaction initiator authorizes the transaction to create the smart contract on the blockchain. Once the smart contract is created and stored on blockchain, it is identified by a contract address. All smart contracts have blockchain account which can contain following:

- A piece of executable code
- An internal storage to store its internal state
- An amount of Ether, i.e. the contract balance

Smart contracts are invoked externally and there is a cost involved for each execution. This cost is known as Gas and it is in terms of Ethers. As the Smart contract execution involve computational and storage resources so these are compensated by Ethereum eco system through payments by gas. The amount of gas to be deducted is mentioned in Ethereum yellow paper [55].

3.6 Hyperledger Fabric

Hyperledger fabric was an IBM initiative which joined the consortium of Linux foundation and became one of the base line projects laying foundations for other Hyperledger frameworks. Fabric is the ultimate choice for enterprise and industrial use because of its modular approach and flexibility in its design to accommodate heterogeneous environment. It has plug and play option for consensus as well as offers support to chaincode development in various programming languages such as Java, JavaScript, Go language and Python. This makes the Hyperledger fabric the ultimate choice for our framework as well. Moreover, the Hyperledger fabric provides scalability as well as privacy in a permissioned environment without compromising on performance. The basic building blocks of Hyperledger are as following:

3.6.1 Identity Management

Identity is the first part of any IT system; it helps identify various actors in an organization and these actors then verify their identity through authentication and are authorized to perform certain actions allowed by the system. Without a centralized approach to identity management it is a challenge for IT professionals to maintain authentication and authorization across large number of IT devices. X.509 certificates provide detailed identity which is verifiable by the system administrators. In Hyperledger Fabric two entities play vital role in identity management and are as follows:

a. Certificate Authority (CA)

   A Certificate Authority is an entity responsible to dispense Certificates to the various actors in a network. These certificates bind the public key of the principal (The device to whom identity belongs) and various other attributes associated and are digitally signed by the CA. Consequently, if CA is a trusted entity and its public key is known, then one can trust that the specific principal is has a valid certificate, and owns the included attributes and public key, by validating the CA's signature on the principal's certificate. CA can be of various types e.g., Root CA, Department CA and local CA. If an entity for example a patient is issued an identity by Root CA his identity will be available in every department. The identities are issued physically to each device or entity. CA provide following features:

i. Registration of Identities

ii. Issuance of Certificates

iii. Certificate renewal and revocation

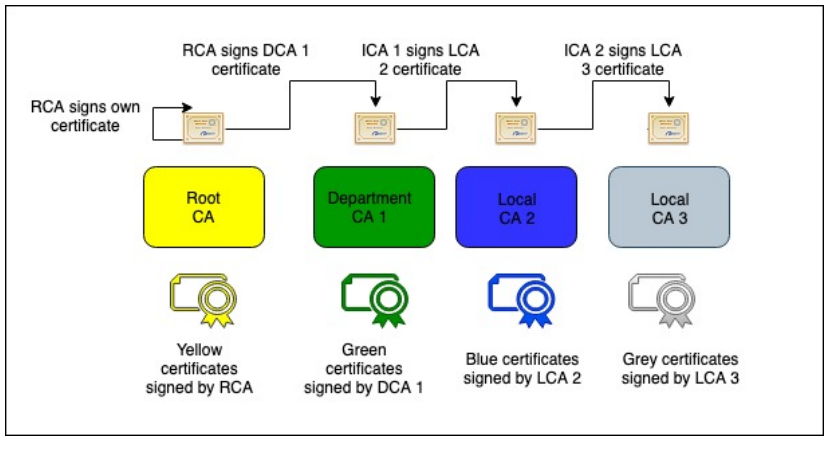


Figure 3.1: CA Hierarchy of Adaptive Security Framework

b. **Membership Service Provider**

Once an Identity is issued it must be verifiable for this purpose, we require another entity known as MSP (membership Service Provider). The CA has not been delegated with verification in order to distribute trust. MSP is responsible for managing identities once they have been created by the CA. The MSP can also be deployed at any level and depends on the network size and security requirements. A device itself can have an isolated MSP running within which can verify signatures belonging to other actors of the network. If network is large enough, several MSPs can be setup. For example, a channel MSP is responsible for verification of all transactions occurring on that channel.

3.6.2 Client

Clients are the end users which are not directly involved in blockchain process but the main entities involved in transactions. The client is also registered to blockchain, therefore has a particular identity and certificate issued by the CA. The clients submit their transactions to blockchain through anchor peer and once a transaction is successful are responded back by the same.

3.6.3 Peers

Peers are the nodes which are active part of blockchain and they perform one or many roles in the blockchain. These are the nodes which are responsible for maintaining the ledger. Following are the types of peers in our blockchain network:

i. **Endorser**

Endorser or endorsing peer is the one which simulates the transaction by running the chaincodes related to a particular transaction before it is committed to a block. Every chaincode specifies an endorsement policy which defines all the necessary conditions for a transaction to be termed as valid. Furthermore, the endorsers compare the generated RW sets with existing ones in the ledger and validate individually. Every endorser verifies the all the signatures and identities associated with transaction and each endorser forwards back the signed transaction now called "Endorsed Transaction" to the anchor peer. In our system we limit the number of endorsers to 3 with weights according to roles to gain performance benefits.

ii. **Committing Peer**

It is the peer specified or selected by the Blockchain to commit the transaction to the Blockchain. The Leading peer as discussed above is usually the committing peer. This peer commits the transaction to the block as specified by the ordering service and initiates the gossip protocol for ledger update by other peers of channel. This peer can be elected through consensus or may be assigned a specific role.

iii. **Ordering Service**

Ordering service provides the communication channel to all the participants of blockchain and guarantees deliveries. Ordering service can be implemented in variety of ways using different node fault models. It provides connectivity between clients and peers through channel. Clients broadcast their transaction requests which are broadcasted to all peers. The channel supports atomic delivery of all messages.

3.6.4 Channel

A channel is primary communication mechanism for managing communication between entities participating in the Blockchain. Channel behaves like a LAN logically and all data

and transactions are private within channel meaning by the ledgers are maintained and kept within channel and no data is shared with outside peers. In healthcare environment data privacy is of utmost importance, therefore, each department has a separate channel and a device or entity can be part of more than one channel for example, if a doctor has OPD in medical department but also performs duties at the Emergency department, in that case he will have two separate data sets for each channel however his same identity will work across both channels.

When a new channel is created, a genesis block is created which stores the configuration information about the channel policies, members and anchor peers. When a new member is added to an existing channel either the genesis block or a more recent reconfiguration block, is shared with the new member. A leading peer is also elected which is the one which has the responsibility to determine which peer communicates with the ordering service on behalf of the member. If no leader has been designated than a leader is chosen through consensus. The ordering service orders transactions and delivers them to each leading peer in form of a block, which then distributes the block to its member peers, and across theÂăchannel, using theÂăgossipÂăprotocol. The propagation of data, which includes transactional information, ledger state and channel membership, is restricted to only those peers which has verifiable membership for the channel.

3.6.5 Ledger

Ledger provides verifiable history of all successful and unsuccessful transactions occurring over the blockchain. Ordering service is responsible for construction of ledger by maintaining ordered hashchain of blocks of transactions. Hashchain imposes the total order of blocks in a ledger, where each block is an array of totally ordered transactions which formulates an entirely ordered blockchain. All peers have ledgers and optionally orderer can also have a ledger which is called "Order Ledger". All other peers have peer ledgers and they can replay the history of transactions to update or reconstruct the ledger state.

3.7 Hyperledger vs Ethereum

Ethereum was designed basically for public consumption and the designer didnâĂŹt intent to keep it permissioned. This comes at the cost of performance and privacy. Whereas, Hyper-

ledger was designed explicitly for business use hence keeping the performance, scalability and privacy in sight. Both blockchains have their own smart contracts. The Ethereum currently relies on PoW consensus algorithm which requires high computational resources and energy consumption. Whereas, Hyperledger uses PBFT which is very fast and efficient and not at all computationally intensive. Moreover, Ethereum relies on cryptocurrency for its functioning whereas there is no such bounds on Hyperledger fabric. In case of healthcare, privacy and performance are top priority, therefore, Hyperledger fabric has been chosen for this thesis research.

3.8 Blockchain Challenges

As Blockchain is technology enabler for many distributed network-based applications. Thus, it is imperative to review the challenges faced by this technology in order for secure and beneficial utilization of this technology. We have segregated the challenges under technical and non-technical challenges for easy consumption of the readers. Authors in [56] have identified some technical challenges in blockchain but majority of the work is based on Public blockchains such as Bitcoin and Ethereum because 80% of world market is still using these two blockchains based solutions. These challenges are based on earlier work of Swan [57] and are as following:

a. **Throughput**

   In the blockchain world everything is taken as a transaction thus in order to check the speed we go by the parameter of "Transactions per second ". But some argue that transaction per second should not be a concern for blockchain networks as the main feature of blockchain is to provide security not speed [58]. But when we talk of scalability issue or network handling issue as in case of using blockchain for a security solution than the speed is primitive because it also effects security in terms of DOS attacks. A Bitcoin network can handle 7 transaction per second (TPS) [56] [57] [59] whereas currently Ethereum is having 20 TPS which is poor but is likely to improve once Ethereum moves from Proof of Work(PoW) to Proof of Stake(PoS) algorithm. The VISA is currently having average transaction rate as 2000TPS so these public blockchain networks are expected to at least be around this benchmark. Whereas Private blockchains overcome this challenge.

b. **Latency**

Public blockchains like Bitcoin and Ethereum rely on PoW consensus mechanism which establishes the authenticity of a block after finding a mathematical solution to the problem at hand. This mechanism is vulnerable to double spending attack in which a user can transfer same coin or asset more than one time [60]. This is dealt with by public blockchains through verification of each transaction before committing to blockchain and this comes at the price of latency. Latency in private blockchains like Hyperledger is much less because of PBFT as the transaction is fully committed after a voting-based consensus is reached [61].

c. **Bandwidth and Size**

Bitcoin blockchain data size is 238 GB at the time of writing this thesis [62]. This is estimated to increase to 214 PB every year if the transaction rate of bitcoin increases from 7 tps to that of twitter and VISA. This issue is a challenge for permissionless blockchains more than permissioned ones and thus scalability is an issue in blockchain technology.

d. **Security**

Blockchain technology like others also face few security challenges. 51% attack is the one in which an adversary can get hold of 51% of computational resources of blockchain meaning by it can dominate the mining capability and can introduce hard forks to validate invalid blocks and vis a vis. Similarly, a miner can create a longer blockchain than the actual one to dominate racing conditions resulting into double spending problem as discussed earlier. However, these challenges have been addressed by private blockchains like Hyperledger.

3.9 Conclusion

The blockchain itself is a technology enabler which is designed based on core concepts of cryptography for secure transactions in untrusted environment. This technology is the backbone of web 3.0 through utilization of smart contracts invented by Ethereum. However, this technology still faces various challenges in terms of performance and security which have been discussed in this chapter.

Chapter 4

Adaptive Security Framework

4.1 Introduction

Our Architecture is based on Hospital centric healthcare services but can be enhanced for home centric environment as well. The Healthcare services are crucial requiring high privacy, data integrity and availability. Therefore, Permissioned Blockchain was found more suitable for the purpose as justified in Chapter 3. Our framework is based on Hyperledger hence some of the entities are named accordingly and the assigned roles are also defined according to the specific function they perform in blockchain. In next section the generalized scenario of our architecture is explained as a layman view followed by the technical details of the framework.

4.2 Threat Scenario

Let us consider a hospital with various specialized departments. The hospital has recently adopted the latest technology to provide state of the art medical facilities to its Patients. With adoption of IoT however, serious threats emerge regarding the security of such devices and adherence to HIPAA and GDPR. The current Cloud based solutions for data management provide a solution for Data protection but are treated as "honest but curious " and may act as adversary. Similarly, a doctor or a technician may abuse their privileges and misuse the patient information without his consent violating HIPPA and GDPR. This necessitates a solution which can provide secure access control mechanism without a central point of failure and system must be self-reliant so that no external actor can interfere in the security process. The system must also be able to satisfy transparency, immutability and provenance as basic requirements. The new system should be easily integrated with existing mechanisms and must still ensure viable communication across various organizations for interoperability and ensuring smart health integrated services integrating home centric health solutions. Keeping the above tasks in mind we first design a threat model which will be required for evaluation of our Adaptive Security Framework.

4.2.1 Attackers

Attackers whether insider or outsider mostly interact with system as a user. In healthcare monitoring systems the attacker can be an insider compromising EHR and selling them on black market or it can be an outsider with ill intentions to malign hospital reputation by disturbing the working mechanisms of medicals devices. As recently, a vulnerability was found in authentication of Anesthesia devices of GE Aestiva and Aespire [63]. This vulnerability allows a remote attacker to modify device parameters like changing gas density, silencing alarms and warnings and even changing the time settings of machine. Thus, in our threat model both insider and outsider attackers are considered.

4.2.2 Assets

Hospitals provide healthcare services which are life critical and thus any system or device dealing with any sort of healthcare data is treated as an asset. The data is collected from sensors, synthesized by special servers into intelligent information which can be translated to the Patients health records for analysis and treatment buy Caregivers. This data is then stored may be by own hospital database or may be integrated with cloud services for inter-operability between various medical organizations, government and services like insurance. Thus, the assets in healthcare environment under consideration for our solution are following which directly interact with the hospital monitoring system.

1) Medical IoTs

2) Caregivers

3) Patients Health records

4) Gateways, Database servers involved in computations

4.2.3 Threats

Healthcare faces more imminent threats because of high value of patient information in black market and large volume of sensitive data easily available as least importance is given to cyber security in healthcare. Protection against cyber threats in compliance with HIPAA can be challenging and any oversights could easily cost a breach or regulatory fine. Following are the threats identified in healthcare environment which required to be mitigated by our suggested solution:

1. Unauthorized access to medical sensors and devices.
2. Tempering of recorded patient data.
3. Corruption of data by collusions of peers.
4. Leakage of information between various tiers (hospital, cloud services and other organizations).
5. Accidental or deliberate loss of data by caregivers.
6. Unauthorized access to medical data by users in contrast to assigned roles and responsibilities.
7. Manipulation of activities and manipulation of auditing history.

4.3 System Design

Hospital is taken as a one big organization under which the departments work in isolation as far as disease specific treatment is concerned. However, this may happen that a cancer patient is also being treated by a dentist thus the data is required to be shared between the two departments. But in these cases, even the patient is referred from one department to other for a certain case and each case is exclusive from each other. As our architecture has been designed specific to IoT devices, these devices are mostly deployed for a specific service at a departmental level and it is highly unlikely that someone from some other department will seek access request to device data directly. Likewise, it is highly unlikely that a device is moved temporarily from one department to other and if such is the case the device will be re-registered in the new department. Fig 4.1 shows the basic layout of medical department. IoTs associated with a patient are connected to gateway which is part of Blockchain and acts as Anchor peer for IoT devices. Caregivers form integral part of blockchain network and are randomly assigned roles of blockchain peers according to their privileges defined in certificate.

We leverage the existing identity management system of Hyperledger which provides functionality for deployment of CA (Certifying Authority) locally. In our framework we have a Root CA at hospital level who is responsible for registering the departmental level

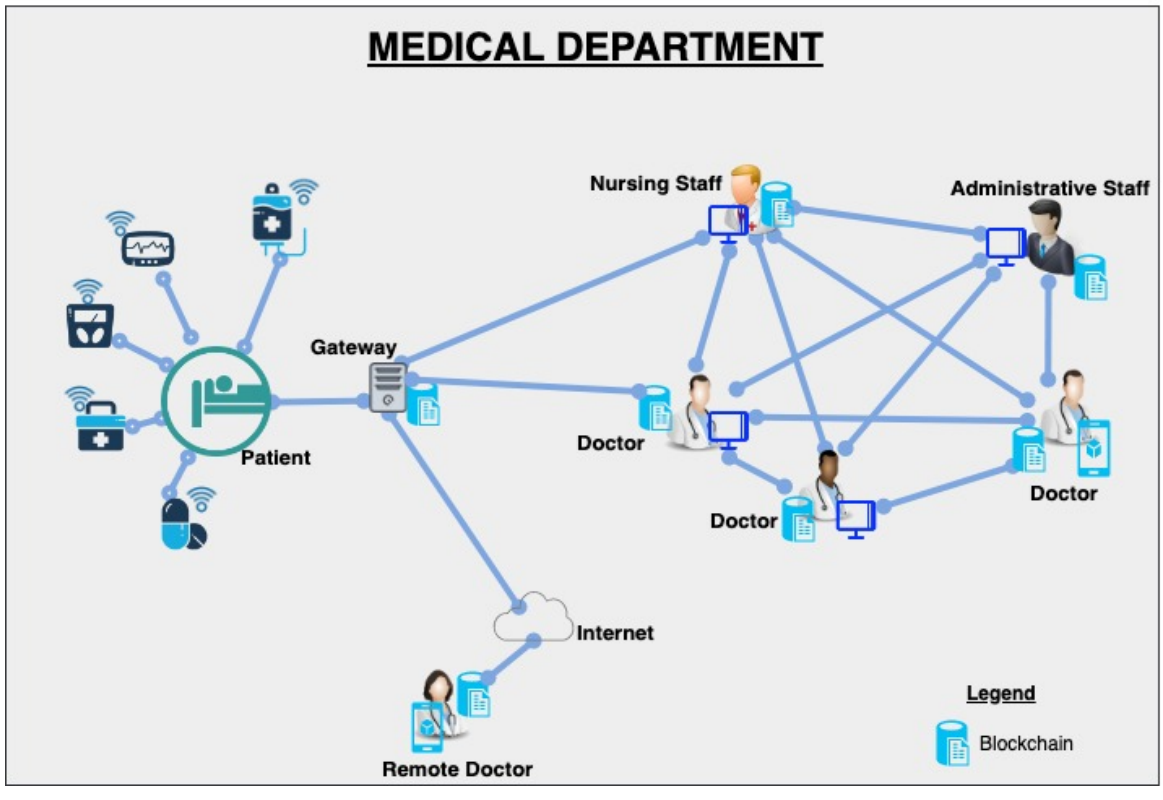


Figure 4.1: Birdseye view of System Layout

CA who are responsible for issuing certificates further. The design is flexible and can be decentralized further or may be centralized as per the requirement and capacity of an organization. In our case we have kept it decentralized to achieve performance benefits. All transactions have Patients as context as all healthcare data transactions have to be authorized by patients. The IoT devices thus provide data in context of patients and will be discussed subsequently. SMD (Smart Medical Devices) and RE(Requesting Entities) are clients in our case and they interact with blockchain through Anchor peers which in case of SMD is gateway and in case of RE it's the device itself if also designated as a peer (Doctor, Nursing Staff, Administrator). The client is also registered to blockchain, therefore has a particular identity and certificate issued by the CA. The clients submit their transactions to blockchain through anchor peer and once a transaction is successful are responded back by the same.

4.3.1 Transaction Flow in Blockchain

To understand the transaction flow in semantic way and for easy understanding, let us consider a scenario where a doctor wants to get ECG readings of a patient from an ECG machine which we call SP_{ECG} and the doctor is RE_D (Requesting Entity) in this case. The Doctor

can be serving in multiple departments in a hospital for example a heart specialist will have emergency duty in Medical Department Emergency, thus in order to carry out the transaction in focus which is in medical department he has interact with blockchain using the id associated with this department. Further, the context is very important especially healthcare environment so transaction must be associated by a single context and in our case every transaction has a common context and that's patient. Following is stepwise scenario and each phase of transaction is discussed sequentially:

- **Transaction Flow**

  The transaction flow in Hyperledger of our framework is shown in Fig and as follows:

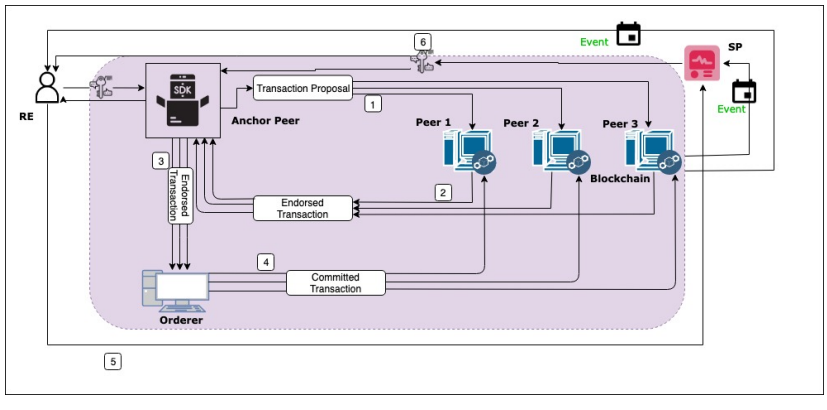


Figure 4.2: Hyperledger transaction flow

1) The RE_D initiates a request access transaction by sending transaction parameters using blockchain protocol of Hyperledger. The clients are connected through anchor peers as already discussed, in this case the RE_D itself is an anchor peer and can initiate transaction. The transaction packet contains following parameters T_A = $\{\text{ID}_{RE}||\text{ID}_P||\text{ID}_{SP}||\text{Access Type}||\text{Nonce}_{SP}\}$.

2) The transaction parameters are verified by the Endorsing peers, in each case depending on the group of devices interacting, a set of Endorsing peers are nominated and these can be assigned weights or can use any pluggable consensus algorithm supported by Hyperledger fabric. The Endorsing peers simulate the Read, write set of transaction meaning by, they simulate the chaincodes and verify the inputs and outputs.

3) Once the chain code is successful, endorsing peers forward back endorsed transaction to Anchor peer meaning by, the transaction now includes the signatures of each Endorsing peer.

4) The Anchor peer forwards the endorsed transaction to Orderer who verifies the Endorsed transaction. All the validations of transaction involve a local MSP running within each peer as separate module and it is responsible for verifying all the signatures of every transaction.

5) The Orderer after verifications assigns a block number to the transaction TR and initiates gossip protocol. Once gossip protocol is initiated all the Peers of concerned channel update their ledgers.

6) An Event is generated on completion of this transaction and the RE_D is granted access according to the current access right set of doctor. After successful transaction SP generates a simple transaction to send a Nonce to RE which is also recorded on ledger.

4.4 Transaction logic

The main driving force of our adaptive security mechanism is the chaincode part of transaction. Here we try to utilize the computational power and rich features of chaincode for maximum benefit of driving security in a distributed fashion. In order to work efficiently the framework requires at least 50 transactions data stored on blockchain. Thus, biometric based verification will be done in initial transactions and predefined access rights will be used. After 50th transaction the framework will be initialized. The transaction logic is based on three functions for understanding purpose however constitute part of same chain code. Fig 4.4 gives the overview of chaincode logic described below:

4.4.1 Authentication Function

The Authentication mechanism is designed to achieve adaptivity through risk assessment based on parameters usually available in the network packets such as HTTP header. The RE will always initiate a transaction request in a particular context in case of healthcare this context is Patient. Thus, all transactions must contain patient's ID along with RE and SP ID. Furthermore, chaincode will get additional parameters from HTTP header and these param-

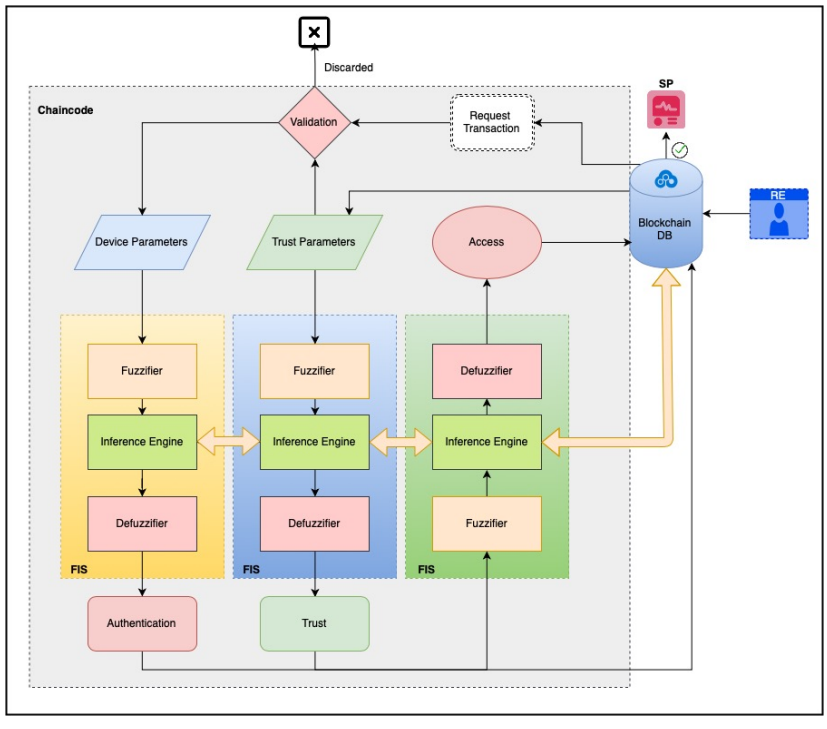


Figure 4.3: Chaincode Logic

eters will be analyzed in conjunction with history of transactions maintained by Blockchain. We define a Mamdani FIS for our authentication system as shown in fig 4.5. The parame-

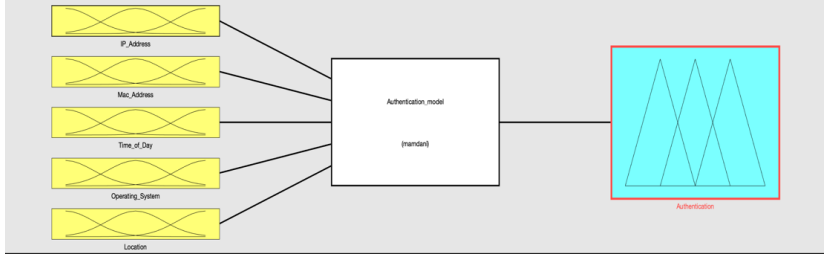
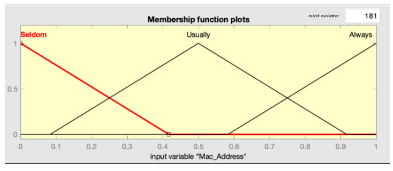


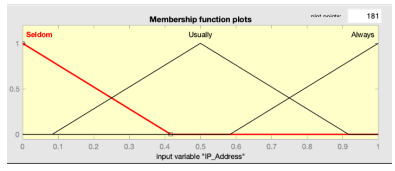
Figure 4.4: Authentication FIS

ters for our framework are IP address, MAC address, Time of day, Operating System and Location. Each parameter will be analyzed separately and frequency distribution for that particular parameter will be calculated. This frequency distribution is normalized to get the membership functions for each Fuzzy set associated with parameter. For example, we define three fuzzy sets for each parameter which are seldom, usually and always. The membership functions and sets are shown in fig 4.6 (a-e).

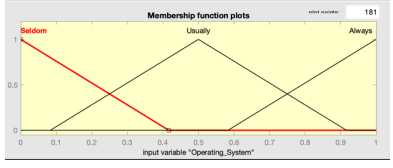
Mamdani FIS is used to calculate fuzzy output which is type of authentication mechanism. Based on 5 parameters and each having 3 fuzzy sets, 243 rules can be defined for fuzzy system, fig 4.7 shows 29 rules due to space constraint and in this example the frequency dis-



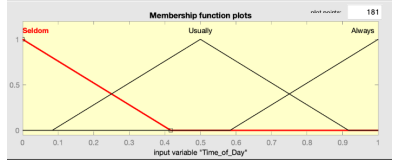
(a) IP Address



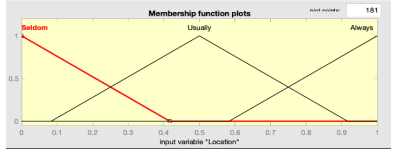
(b) Mac Address



(c) Time of day



(d) Operating System



(e) Location

Figure 4.5: Membership Functions of Device Parameters

tribution for parameters is 0.206 IP, 0.55 mac address, 0.179 for Time of day, 0.133 for Operating System and 0.095 for location thus subsequent output of fuzzy system is 0.391 which is Biometric authentication. These rules are generated only once and stored on blockchain.

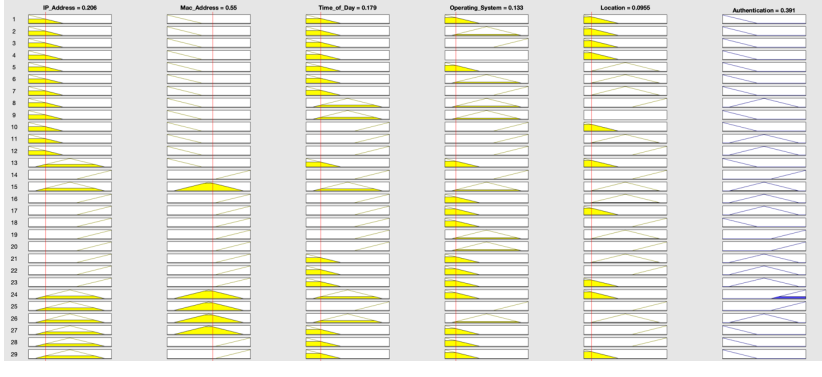


Figure 4.6: FIS Rules Viewer

The output contains 3 fuzzy sets of biometric, OTP and CA and their membership functions are shown in fig 4.8. According to the given parameters, the MFA is applied and RE is required to authenticate through particular method given by fuzzy output. For example, the Membership function of device is max for Biometric than the device will be authenticated through Biometrics. Furthermore, Biometrics and OTP based authentication also involve an OTP being sent to patient device for endorsement. On successful authentication a nonce generated by IoT during previous transaction is hashed with Hash of last valid transaction

and new hash is treated as direct knowledge K_d for RE.

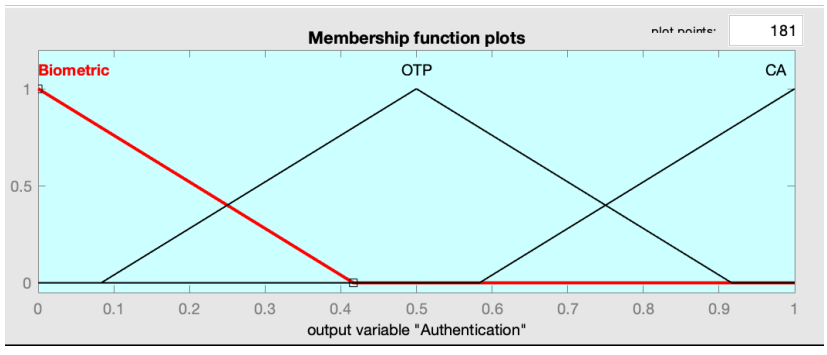


Figure 4.7: Membership Function of Authentication Output

4.4.2 Trust Evaluation function

The purpose of this function is to provide trust feedback based on previous transactions as input to the Fuzzy logic of Authorization transaction. The Trust feedback along with Authentication provides sufficient proof for fuzzy logic to apply rules to assign which type of access privileges the RE can have. The RE request is mapped to particular access right permission set according the trust feedback score. Trust of a device constitute of three main elements [39] discussed as following:

a. **Experience:** The transactions experience which is dependent on the previous transactions between RE and SP. The experience ${}_{RE}E_{SP}$ is calculated by eq (1)

$$ {}_{RE}E_{SP} = \begin{cases} 0 \text{ if } n = 0 \\ \frac{\sum_{t=1}^{n} E_t}{\sum_{t=1}^{n} |E_t|} \text{ if } n \neq 0 \end{cases} $$

Here, range of ${}_{RE}E_{SP}$ is $\in$[-1,1]. E_t is +1 for successful transaction and -1 for unsuccessful transaction. The membership functions and fuzzy sets of ${}_{RE}E_{SP}$ is shown in Fig 4.9.

b. **Knowledge:**As discussed above K_d is calculated in each transaction and if K_d is different then -1 or else 1 is given as value of Kd and aggregate value of ${}_{RE}K_{SP}$ is given by eq (2)

$$ {}_{RE}K_{SP} = \begin{cases} 0 \text{ if } n = 0 \\ \frac{\sum_{t=1}^{n} K_t}{\sum_{t=1}^{n} |K_t|} \text{ if } n \neq 0 \end{cases} $$

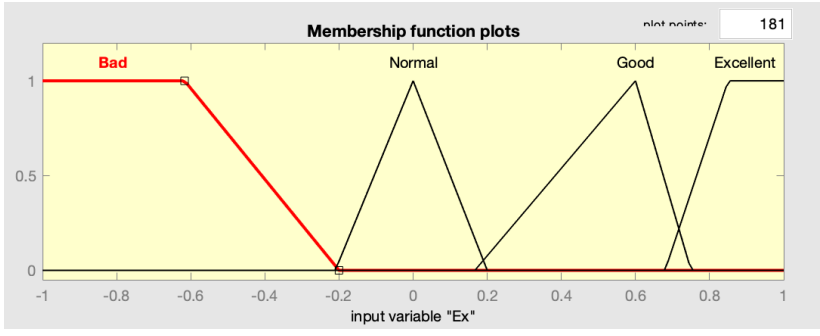


Figure 4.8: Membership functions of ${}_{RE}E_{SP}$

In eq ${}_{RE}K_{SP}$ ∈[-1,1] and denotes the knowledge of RE with respect to SP. The membership functions and fuzzy sets for ${}_{RE}K_{SP}$ are shown in fig 4.10.

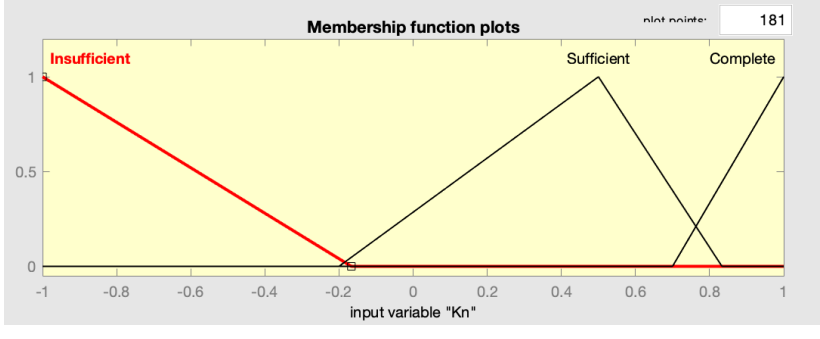


Figure 4.9: Membership functions of ${}_{RE}K_{SP}$

c. **Reputation:**

The last is the Reputation calculated by blockchain based on the experiences of all devices with pretext to RE. In this case the context is RE, thus Reputation is given by eq (3)

$$R_{RE} = \frac{\sum_{t=1}^{n}\{E_{sp}\}_t}{\sum_{t=1}^{n}|E_{sp}|_t}$$

In eq R_{RE} ∈[-1,1] and denotes the experience of BAN SP devices with RE. The membership function and fuzzy sets associated with reputation are shown in fig 4.11. The fuzzy output in terms of trust is calculated based on 27 rules. The fuzzy output in terms of trust is shown in fig 4.11.

4.4.3 Access Control Function

The last function is Access control function. In this function the Trust and Authentication linguistic values of previous functions is taken as input and Access Control is given as an output Fig 4.13. The Access Rights are linguistically defined as {ϕ,Read, Read/Write,

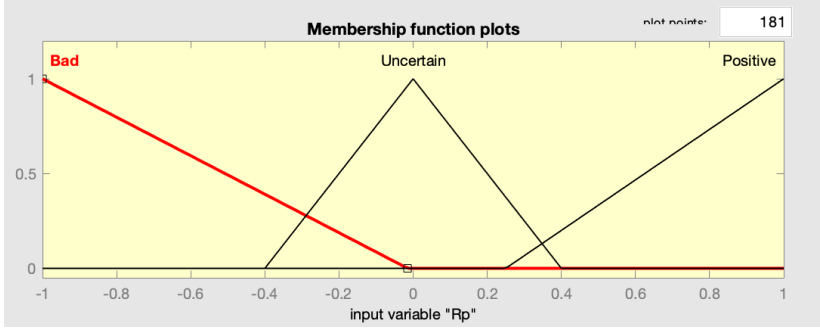


Figure 4.10: Membership functions of R_{RE}

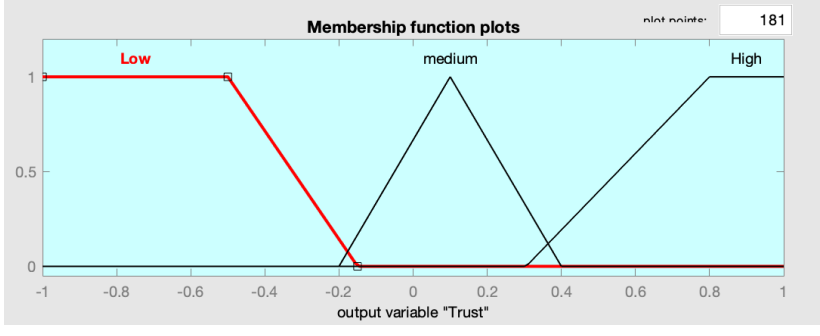


Figure 4.11: Membership functions of Output 'Trust'

Read/Write/execute} and their membership functions are shown in fig 4.14. The authentication input provides a fresh behavior input of RE whereas the Trust function provides a feedback-based input and this way the access control is adjusted according to device behavior. For example, if trust is low and the device had authenticated through biometrics the output is No Access as shown in fig 4.15. The device access is revoked and it is asked to revalidate its certificate through admin and admin is notified. If trust is high and authentication is OTP based than access assigned is different. If a device is assigned NO Access, the RE is deemed as malicious, its access is revoked and it has to re-validate its certificate through CA and the transaction parameter of REESP is given -1 value accordingly for this transaction. Otherwise the access is granted on basis of least privilege. For example, if output access right is Read/Write whereas the permissions defined for device only contain read access the device will be granted only read access.

4.5 Framework Simulation

The framework was designed in MATLAB and tested for different use cases. The parameters were chosen at random to validate concept and analyze outputs of each function. Fig

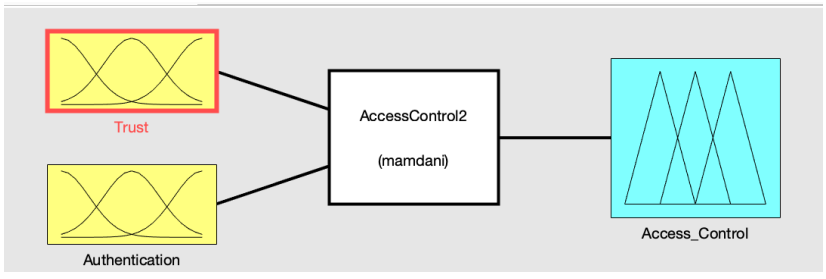


Figure 4.12: Access Control FIS

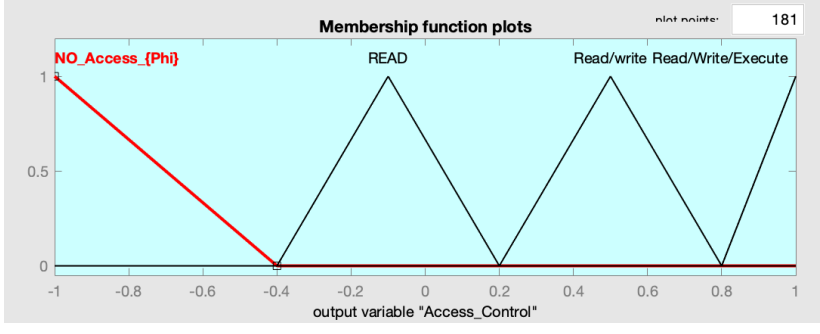


Figure 4.13: Membership functions of Output 'Access Right'

4.16 shows random values for each parameter in authentication function and the crisp output of authentication function is calculated using Centroid defuzzification method. The surface view in fig 4.16 shows the input/output domain of Ip Address and Mac Address. The frequency distribution of both inputs is directly proportional to Authentication mechanism in use.

The MATLAB tested logic was than applied to Hyperledger fabric for functioning validity. The architecture is validated and as the number of transactions increases the Fuzzy Output gets more precise.

4.6 Conclusion

This chapter discusses in detail the framework of our adaptive security mechanism. The core authentication and authorization mechanism of system design are based on behavioral changes based on device input parameters. Blockchain provides trust parameters based on transaction data provenance and immutability. The Access control is context and role-based hybrid ensuring foolproof security based on device behavior. The system is modelled using MATLAB.

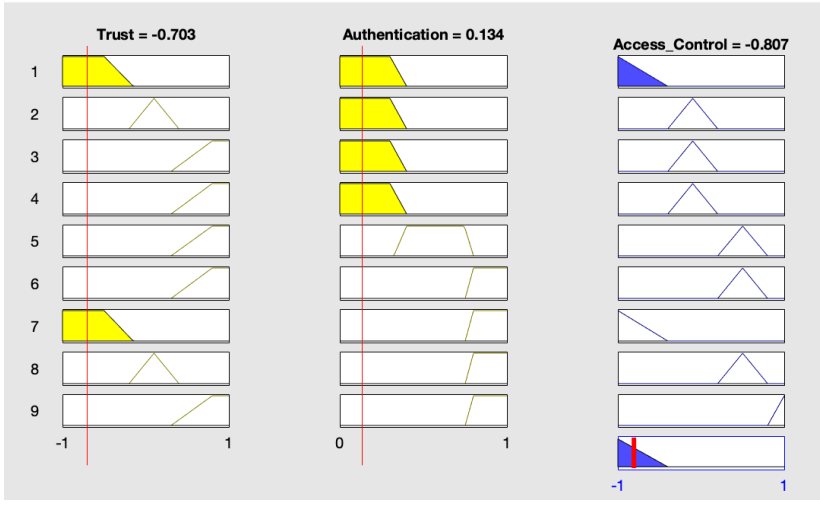


Figure 4.14: Rule Viewer - Access Rights

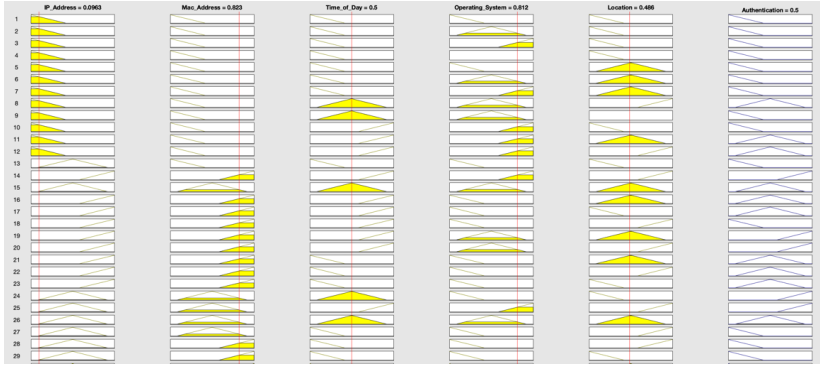


Figure 4.15: Context Behavior Based Crisp Output of FIS

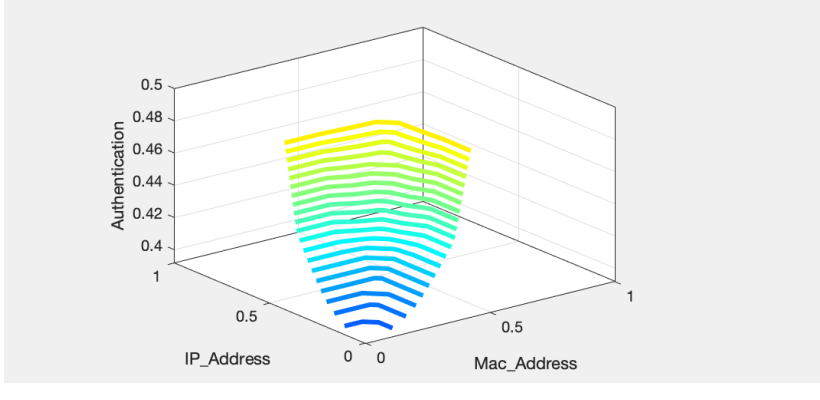


Figure 4.16: Surface View of Authentication with two input variables

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Introduction

The functionality validation of framework was tested on MATLAB R2019a by implementing the Mamdani based FIS as already discussed in chapter 4. After successfully achieving the adaptive mechanism. The framework was deployed over Hyperledger Fabric. In this chapter the realization of framework as a practical solution is achieved and results are analyzed. Following tools were utilized for the implementation phase:

5.1.1 Hyperledger Fabric

Hyperledger fabric is a permissioned distributed ledger framework hosted by Linux foundation in collaboration with IBM [60]. It provides decentralized applications-based enterprise solutions. It has modular architecture and supports different consensus algorithms and databases enabling performance and scalability in various domains. Hyperledger fabric version 1.4 was used for the project.

5.1.2 Docker

Docker is a tool designed for secure deployment and functioning of applications using containers. Developers can package all the libraries and dependencies of an application in the container and transfer it as one package. This enables the smooth execution of application regardless of difference in specifications in machines. Unlike virtual machines docker run on same kernel thus boosting the performance and reduction in size of application. Docker desktop version 2.0.0.3 was used for implementation.

5.1.3 Node.js

Node.js is an open-source cross platform tool to provide runtime environment for development of server-side and network-based applications. It uses JavaScript programming language for developing web applications. Version 10.16.3 of node.js was used.

5.1.4 System Specifications

Table 5.1 shows the system specifications used for design and testing of our proposed framework.

Table 5.1: System Specifications

Sno	Item	Version/ Specification	Remarks
1.	MacBook pro	2.2 GHz core i7, 16 GB RAM and 256 GB SSD	
2.	OS	MacOS Mojave 10.14.6	
3.	Hyperledger Fabric	1.4	
4.	Docker Desktop	2.0.0.3	
5.	Node.js	10.16.3	
6.	VS Code	1.37.1	

5.2 Comparative Analysis

The main objective of our framework is to achieve adaptive security based on user behavior without depending upon traditional security mechanisms like passwords and tokens. Moreover, centralized architecture presents single point of failure and thus vulnerable to many attack vectors like DOS attacks, ransomware attacks etc. Most of the research work in this domain relies on central architecture and very few have utilized the true potentials of blockchain technology. Furthermore, most of the work relies on a single authentication mechanism which may be subverted by the adversaries thus our system adapts by applying second factor authentication based on users' attributes and behavior. Table 5.2 shows comparative analysis of our framework with existing solutions discussed in chapter 2.

Table 5.2: Comparison of Proposed Framework with Existing Solutions

Paper	Decentralized	Authentication	Access Control	Trust Management	MFA	Adaptivity
[14]	✗	✓	✗	✗	✗	✗
[15]	✗	✓	✗	✗	✗	✗
[16]	✗	✓	✓	✗	✗	✗
[18]	✗	✓	✓	✗	✗	✗
[19]	✗	✓	✓	✗	✗	✗
[20]	✗	✓	✗	✗	✗	✗
[21]	✗	✓	✗	✗	✗	✗
[22]	✗	✓	✗	✗	✗	✗
[26]	✓	✗	✓	✓	✗	✗
[27]	✓	✗	✓	✗	✗	✗
[28]	✓	✗	✓	✗	✗	✗
[29]	✓	✗	✓	✓	✗	✗
[30]	✓	✗	✓	✗	✗	✗
[31]	✓	✗	✓	✓	✗	✗
[32]	✓	✗	✓	✗	✗	✗
[33]	✓	✓	✗	✗	✗	✗
[35]	✓	✗	✗	✗	✗	✗
Adaptive Security Framework	✓	✓	✓	✓	✓	✓

5.3 Practical Usability and Comparison with other Blockchains

As discussed in section 3.5 the permissionless or public blockchains faces several challenges regarding performance parameters. The public blockchains like Bitcoin and Ethereum rely on mostly PoW consensus which is heavy and thus involves high latency in order to achieve security. Some of the public blockchains like Litecoin have less block formation time of 2.5 minutes as compared to 10 minutes of Bitcoin. Consequently, Litecoin uses a smaller number of hashes to verify the block as compared to Bitcoin. This problem is absent in Hyperledger because the consensus is achieved through PBFT depending on predefined endorsers and trust is anchored by the governing body as in our case it's the hospital's administrative authority. Thus, virtually there is no deliberate latency for achieving security and the block is formed as soon being verified by the endorsers. The security and performance can be achieved in a similar manner as in traditional networks by limiting the channel users to the concerned parties as the concept of VLANs in traditional networks. This enables

privacy and scalability at the same time by segregating different parts of networks from each other. Therefore, in this thesis our proposed framework was implemented on Hyperledger blockchain to ascertain the practical feasibility of solution as compared to existing state of the art solutions as well as other blockchain based solutions. Following parameters are briefly discussed to give performance overview as compared to other blockchains and technologies:

5.3.1 Latency

Transaction latency is the time transaction takes starting from the point it is submitted to the network to the point it is committed by all peers to the ledger. Hence the performance and throughput somehow rely on this parameter. The take here is the pivot point for latency is the number of endorsers in Hyperledger Fabric. As the number of endorsers increase the latency also increase because it takes more time to collect the endorsed transaction and send to Orderer. That's the reason our system design suggests 3 endorsers with weights as per their roles for smooth functioning of System.

5.3.2 Throughput

Transaction throughput is the amount of time a valid transaction takes to get commit to the blockchain. Many researchers and blockchain benchmarking sites use throughput as the main performance parameter for Blockchain which is not true. As already discussed in section 3.5 the through put of public blockchain networks is very low which is the main reason behind lack of adaption of technology in modern banking systems where this throughput is somewhat around 2000 tps. Whereas in an authentication and authorization setup like ours required throughput is greater. Here, Hyperledger performs better due to its consensus algorithms and segregation of power between different nodes based upon organizational parameters. The throughput of Hyperledger is dependent on following two parameters and can be tweaked accordingly depending on network architecture:

- **Tps vs Block size.** The tps can increase if the block size is reduced. The block size in Hyperledger is fully configurable and can be adjusted as per the network requirements. We used a block size of 10 transactions and single transaction with our chaincode logic took approximately 5ms. This value is not fix and can change depending on multiple factors like changing the chaincode programming language, frequency,

orderers,consensus type and channels.

- **Tps vs Endorsing peers.** As we increase the number of endorsing peers it takes more time for a transaction to get committed to the blockchain after due endorsement of each endorser. The performance can further degrade if we use endorsers from multiple medical departments as per our scenario. This is the main reason for keeping a separate channel on department basis and using weighted endorsement approach limiting the number of endorsers to three to achieve max performance for proposed architecture. However, as already discussed these parameters are configurable in Hyperledger and can be tweaked accordingly as per requirements of the deploying network.

5.3.3 Mitigation Strategies

Table 5.3 enumerates the mitigations strategies against most common threats achieved through our framework to achieve security objectives for IoTs in healthcare domain.

5.4 Conclusion

The system logic was designed and tested on Hyperledger fabric and was found practically suitable for employment in healthcare environment. The parameters affecting performance of system are discussed and relevant suggestions to improve are solicited. In the end mitigation strategies against various threats of information security achieved through our framework are tabulated.

Table 5.3: Mitigation Strategies

Threat	Strategy	Description
Spoofing	X.509 Certificates (Provided by Hyperledger Fabric)	All entities can interact with blockchain only through certificates issued by CA and reliance on local CA hierarchy eliminates third party breaches
Tampering	Blockchain's cryptographic means SHA256, ECDSA)	Blockchain provides immutability through use of hashing and signatures
Repudiation	Digital Signatures	All transactions include signatures thus no entity can deny its actions
Replay Attacks	Read/ write sets, version number	Endorsers use read write sets to validate transactions, invalid key value pairs and version numbers simply deem a transaction invalid
Remote Access	MFA	The Adaptive MFA ensures the device behavior is consistent with usage and only granted access after behavior analysis
Privilege Escalation	Identity Management and Access Control	The X.509 based issued identities define roles and Adaptive access control mechanism works on least privilege mechanism.
Ransomware/ Malware	Adaptive Security	Behavioral analysis helps in better authentication and access control mechanism to deny access to malicious entities.

Chapter 6

FUTURE WORK AND CONCLUSION

6.1 Conclusion

The healthcare domain is mission critical and inclusion of IoT has greatly improved the health services and use of such technologies is being rapidly adapted across the globe. However, these devices have a bigger threat vector and lack of IT knowledge in caregivers makes it a bigger problem. Access control and authentication breaches have been found to be root cause of security breaches in healthcare and IoT. This research work leverages the blockchain technology to provide decentralized trust and computation to achieve behavior driven adaptive security for IoTs in healthcare domain in order to cater for aforementioned threats.

This research work provides an alternative to the common password-based authentication which is prone to dictionary attacks, guessing etc. The solution successfully utilized blockchain technology to achieve decentralized authentication eradicating single point of failure as found in state-of-the-art security solutions. A novel approach towards adaptive authentication and access control is achieved which can detect and deny access to a malicious entity. The comparative analysis shows significant performance and security benefits as compared to other blockchain based security solutions for IoTs.

We validate the practical feasibility of proposed architecture and comparative analysis proves choice of consortium blockchain like Hyperledger has several advantages over public blockchains. The adaptive security approach can open new avenues by exploration of AI and blockchain amalgamation using utilized approach to achieve further security objectives for IoT in healthcare domain.

6.2 Future Work

The proposed framework presents a novel approach to explore the capabilities of blockchain technology for enhancing security specially in IoT domain. Following are the possible future research related objectives:

- Use of Adaptive Neuro Fuzzy Inference Systems utilizing decentralized processing of blockchain to enhance the behavioral analysis and extending the idea to intrusion detection and intrusion prevention mechanisms.
- Benchmarking the proposed solution for deployment in corporate sector by performance analysis.
- Integrating a data privacy management solution to make a complete blockchain based healthcare IoT security management package for deployment.

BIBLIOGRAPHY

[1] J. Villasenor, "How connected healthcare today will keep the doctor away, Freescale semiconductor," [Online]. Available: http://www.freescale.com/healthcare.

[2] Goodin and Dan, "9 baby monitors wide open to hacks that expose users' most private moments," 2015. [Online]. Available: https://arstechnica.com/information-technology/2015/09/9-baby-monitors-wide-open-to-hacks-that-expose-users-most-private-moments/. [Accessed 10 1 2019].

[3] Barth and Bradley, "Philips cardiovascular software found to contain privilege escalation, code execution bugs," 18 August 2018. [Online]. Available: https://www.scmagazine.com/home/security-news/vulnerabilities/philips-cardiovascular-software-found-to-contain-privilege-escalation-code-execution-bugs/. [Accessed 16 Jan 2019].

[4] A. A. Vladimirov, K. V. Gavrilenko and A. A. Mikhailovsky, Wi-Foo, Addison Wesley, 2004.

[5] J. Petters, Varonis, 25 May 2018. [Online]. Available: https://www.varonis.com/blog/kerberos-authentication-explained/. [Accessed 16 January 2019].

[6] Errata Exist, "The OAuth 2.0 Authorization Framework," October 2012. [Online]. Available: https://tools.ietf.org/html/rfc6749. [Accessed 15 january 2019].

[7] F. Ingo, "Challenges from the Identities of Things," in 2014 IEEE World Forum on Internet of Things (WF-IoT), 2014.

[8] H. Jia-Li and Y. Kuo-Hui, "Novel Authentication Schemes for IoT Based Healthcare Systems," International Journal of Distributed Sensor Networks, vol. 2015, pp. 1-9, 2015.

[9] Thota, M. Gunasekaran, R. Varatharajan, L. Daphne, M. K. Priyan, S. Revathi and Chandu, "A New Architecture of Internet of Things and Big Data Ecosystem for Secured Smart Healthcare Monitoring and Alerting," Future Generation Computer Systems, 2017.

[10] J. Qi, M. Jianfeng, Y. Chao, M. Xindi, S. Jian and A. C. Shehzad, "Efficient end-to-end authentication protocol for wearable health monitoring systems," Computers and Electrical Engineering, vol. 63, pp. 182-195, 2017.

[11] A. Ruhul, H. I. SK, G. Biswas, K. K. Muhammad and K. Neeraj, "A robust and anonymous patient monitoring system using wireless medical sensor networks," Future Generation Computer Systems, vol. 23, 2016.

[12] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang and L. Shu, "Authentication Protocols for Internet of Things: A Comprehensive Survey," Security and Communication Networks, vol. 2017, pp. 1-41, 2017.

[13] R. Diego, C.-P. Luis, L.-C. German, E. d. l. Hoz and M.-M. Ivan, "Applying an unified access control for IoT-based Intelligent Agent Systems," in IEEE 8th International Conference on Service-Oriented Computing and Applications, 2015.

[14] H. Ning, H. Liu and L. T. Yang, "Aggregated-Proof Based Hierarchical Authentication Scheme for the Internet of Things," IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 3, pp. 657-667, 2015.

[15] Chatterjee and A. Singh, "A secure multi-tier authentication scheme in cloud computing environment," in International Conference on Circuits, Power and Computing Technologies, 2015.

[16] C. H. Wen and J. Zhang, "An identity-based personal location system with protected privacy in IOT," in IEEE International Conference on Broadband Network and Multimedia Technology, 2011.

[17] Lin and J. H. Yang, "An ID-Based User Authentication Scheme for Cloud Computing," in Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2014.

[18] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang and X. Shen, "CPAL: A Conditional Privacy-Preserving Authentication With Access Linkability for Roaming Service," IEEE Internet of Things Journal, vol. 1, no. 1, pp. 46-57, 2014.

[19] M. Ali, M. ElTabakh and R. C. Nita, FT-RC4: A Robust Security Mechanism for Data Stream Systems, Tech. Rep, Purdue University, 2005.

[20] T. Kothmayr, C. Schmitt, W. Hu, M. BrÃijnig and G. Carle, "DTLS based security and two-way authentication for the Internet of Things," Ad Hoc Networks, vol. 11, no. 8, pp. 2710-2723, 2013.

[21] S. Moosavi, T. N. Gia, E. Nigussie, A. M. Rahmani, S. Virtanen, H. Tenhunen and J. Isoaho, "End-to-end security scheme for mobility enabled healthcare Internet of Things," Future Generation Computer Systems, vol. 64, pp. 108-124, 2016.

[22] S. R. Moosavi, E. Nigussie, S. Virtanen and J. Isoaho, "Cryptographic key generation using ECG signal," in 14th IEEE Annual Consumer Communications & Networking Conference, 2017.

[23] Gandhi and P. M. Kumar, "Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application," in The Journal of Supercomputing, 2017.

[24] Salah and M. A. Khan, "IoT security: Review, blockchain solutions, and open challenges," Future Generation Computer Systems, vol. 82, pp. 395-411, 2018.

[25] Z. Kaspars and S. RenÄĄte, "Blockchain Use Cases and Their Feasibility," Applied Computer Systems, vol. 23, no. 1, pp. 12-20, 2018.

[26] G. Zyskind, O. Nathan and P. A. sandy, "Decentralizing Privacy: Using Blockchain to Protect Personal Data," in IEEE Security and Privacy Workshops, 2015.

[27] Gauravaram, A. Dorri, S. S. Kanhere and R. Jurdak, "Blockchain for IoT security and privacy: The case study of a smart home," in EEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2017.

[28] A. Ouaddah, E. A. Abou and O. A. Ait, "Fairaccess: a new blockchain-based access control framework for the internet of things," Security and Communication Networks, vol. 9, no. 18, p. 5943âĂŞ5964, 2016.

[29] 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, p. 1, 2018.

[30] A. Ramachandran and D. Kantarcioglu, "Using blockchain and smart contracts for secure data provenance management," arXiv, 2017.

[31] C. Qu, M. Tao, J. Zhang, X. Hong and R. Yuan, "Blockchain Based Credibility Verification Method for IoT Entities," Security and Communication Networks, 2018.

[32] A. Azaria, A. Ekblaw, T. Vieira and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in Open and Big Data (OBD), International Conference on IEEE, 2016.

[33] Kim and C. H. Lee, "Implementation of IoT system using block chain with authentication and data protection," in 2018 International Conference on Information Networking (ICOIN), 2018.

[34] M. Banerjee, J. Lee and R. Choo, "A blockchain future for internet of things security: a position paper," Digital Communications and Networks, vol. 4, no. 3, pp. 149-160, 2018.

[35] S. Huh, S. Cho and S. Kim, "Managing IoT devices using blockchain platform," in 2017 19th International Conference on Advanced Communication Technology (ICACT), 2017.

[36] V. Shermin and K. Valentin, Blockchain A Beginners Guide, https://blockchainhub.net/, 2017.

[37] Hintzman and Zane, "Comparing Blockchain Implementations," SCTE-ISBE and NCTA, 2017.

[38] P. Suporn, S. Chaiyaphum and T. Suttipong, "Performance Analysis of Private Blockchain Platforms in Varying Workloads," 2017.

[39] J. LEI, G. CUI and G. XING, "Trust Calculation and Delivery Control in Trust-Based Access Control," Wuhan University Journal of Natural Sciences, vol. 13, no. 6, pp. 765-768, 2008.

[40] P. N. Mahalle, P. A. Thakre, N. R. Prasad and R. Prasad, "A fuzzy approach to trust based access control in internet of things," in 3rd International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace and Electronic Systems, 2013.

[41] A. Walker, "Risk-Based Authentication: The Future of Workplace Security," G2, 18 January 2018. [Online]. Available: https://learn.g2.com/trends/risk-based-authentication. [Accessed 26 October 2018].

[42] E. Vanderburg, "TCDI," 2018. [Online]. Available: A Certified Lack of Confidence: The Threat of Rogue Certificate Authorities.

[43] S. Nakamoto, "Bitcoin," 2009. [Online]. Available: https://bitcoin.org/bitcoin.pdf. [Accessed 20 November 2018].

[44] F. T. BjÃűrn Scheuermann, "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies," IEEE Communications Surveys & Tutorials, vol. 18, 2016.

[45] V. V. ,. E. G. S. S. Chandrakumar, "KARMA: A secure economic framework for peer-to-peer resource sharing," in Proc.1st Workshop Econ. Peer 2 Peer SYst, 2003.

[46] M. Phillip, "What Is Bit Gold? The Brainchild of Blockchain Pioneer Nick Szabo," COIN CENTRAL, 22 May 2018. [Online]. Available: https://coincentral.com/what-is-bit-gold-the-brainchild-of-blockchain-pioneer-nick-szabo/. [Accessed 10 January 2019].

[47] J. Cope, "Computer World," 8 April 2002. [Online]. Available: https://www.computerworld.com/article/2588287/networking-peer-to-peer-network.html.

[48] "Tech differences," 9 January 2017. [Online]. Available: https://techdifferences.com/difference-between-client-server-and-peer-to-peer-network.html. [Accessed 22 October 2018].

[49] Lipovyanov and Peter, Blockchain for Business 2019, Birmingham: Packt, 2019.

[50] X. Xiwei, W. Ingo and S. Mark, Architecture for Blockchain Applications, Springer, 2018.

[51] Blockgeeks, "Blockgeek," 2018. [Online]. Available: https://blockgeeks.com/guides/cryptographic-hash-functions/. [Accessed 12 October 2018].

[52] ISO, "International Organization for Standardization," 2016. [Online]. Available: https://www.iso.org/committee/6266604.html.

[53] Morris and Nicky, "Ledger Insights," 2018. [Online]. Available: https://www.ledgerinsights.com/iso-blockchain-standards/. [Accessed 8 November 2018].

[54] V. Hermin and K. Valentin, Blockchain Handbook: A Beginners Guide, Blockchain Hub, 2017.

[55] D. G. Wood, "ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER," 9 june 2019. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf. [Accessed 13 June 2019].

[56] Y.-H. Jesse, K. Deokyoon, C. Sujin and P. Sooyong, "Where Is Current Research on Blockchain Technology?âĂŤA Systematic Review," PLOS ONE, vol. 11, no. 10, 2016.

[57] M. Swan, Blockchain: Blueprint for a New Economy, O'Reilly, 2015.

[58] J. Halladay, "Hacker Noon," 28 November 2018. [Online]. Available: https://hackernoon.com/the-biggest-myth-in-blockchain-transactions-per-second-c300ca16d802. [Accessed 10 Decemeber 2018].

[59] O'Neal and Stephen, 22 January 2019. [Online]. Available: https://cointelegraph.com/news/who-scales-it-best-inside-blockchains-ongoing-transactions-per-second-race. [Accessed 27 Feb 2019].

[60] M. Rosenfeld, "Analysis of hashrate-based double-spending," 13 Decemeber 2012. [Online]. Available: https://bitcoil.co.il/Doublespend.pdf. [Accessed 23 January 2019].

[61] A. Mahesh and B. Salman, "Hyperledger Blockchain Performance Metrics : White Paper," 2018. [Online]. Available: https://www.hyperledger.org/resources/publications/blockchain-performance-metrics#transaction-latency. [Accessed 16 June 2019].

[62] Blockchain, "BLOCKCHAIN," 2019. [Online]. Available: https://www.blockchain.com/charts/blocks-size?scale=1&timespan=all. [Accessed 12 June 2019].

[63] HIPAA, "HIPAA JOURNAL," 10 July 2019. [Online]. Available: https://www.hipaajournal.com/vulnerability-identified-in-ge-aestiva-and-aespire-anesthesia-machines/. [Accessed 23 July 2019].

[64] Linux Foundation, "Hyperledger," 2018. [Online]. Available: https://www.hyperledger.org/projects/fabric. [Accessed December 2018].

[65] F. Paul, "Network World," 14 January 2019. [Online]. Available: https://www.networkworld.com/article/3332032/top-10-iot-vulnerabilities.html. [Accessed 19 February 2019].

[66] HIPAA, "HIPAA Guide," 2018. [Online]. Available: https://www.hipaaguide.net/gdpr-for-dummies/. [Accessed 25 October 2018].

[67] HIPAA, "HIPAA Guide," 2018. [Online]. Available: https://www.hipaaguide.net/hipaa-for-dummies/. [Accessed 20 October 2018].

[68] Cope and James, "Computer World," 8 April 2002. [Online]. Available: https://www.computerworld.com/article/2588287/networking-peer-to-peer-network.html.

[69] D. Fangfang, S. Yue, M. Nan, W. Liang and Y. Zhiguo, "From Bitcoin to Cybersecurity: a Comparative Study of Blockchain Application and Security Issues," in International Conference on Systems and Informatics, 2017.

[70] M. S. J, T. Gia, E. Nigussie, A. Rahmani, S. Virtanen, H. Tenhunen and Isoaho, "End-to-end security scheme for mobility enabled healthcare Internet of Things," Future Generation Computer Systems, vol. 64, 2016.

[71] MSP SolarWinds, "SolarWinds MSP," 30 november 2018. [Online]. Available: https://www.solarwindsmsp.com/blog/centralized-vs-decentralized-network. [Accessed 12 Decemeber 2018].

[72] F. Paul, "Network World," 28 March 2018. [Online]. Available: https://www.networkworld.com/article/3267065/people-are-really-worried-about-iot-data-privacy-and-securityand-they-should-be.html#nww-fsb. [Accessed 13 Feb 2019].

[73] M. K. Priyan and D. G. Usha, "Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application," The Journal of Supercomputing, 2017.

[74] S. R. Moosavi, E. Nigussie, S. Virtanen and J. Isoaho, "Cryptographic key generation using ECG signal," in 14th IEEE Annual Consumer Communications & Networking Conference, 2017.

[75] T. Kothmayr, C. Schmitt, W.Hu, M.BrÃijnig and G. Carle, "DTLS based security and two-way authentication for the Internet of Things," Ad Hoc Networks, vol. 11, no. 11, pp. 2710-2723, 2013.

[76] Tech differences, "Tech differences," 9 January 2017. [Online]. Available: https://techdifferences.com/difference-between-client-server-and-peer-to-peer-network.html. [Accessed 22 October 2018].

[77] K. Thomas, S. Corinna, H. Wen, B. Michael and C. Georg, "DTLS based Security and Two-Way Authentication for the Internet of Things," Elsevier Journal of AdHoc Networks, vol. 11, pp. 2710-2723, 2013.