Defense Against Integrity and Privacy Attacks

in

The Internet of Things

A thesis submitted in fulfillment
of the requirements for the degree of

Doctor of Philosophy

by

Imran Makhdoom
School of Electrical and Data Engineering
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW 2007, Australia

August 2020

ABSTRACT

The world is resorting to the Internet of Things for ease of control and monitoring of smart devices. The ubiquitous use of the Internet of Things ranges from Industrial Control Systems to e-Health, e-Commerce, smart cities, supply chain management, smart cars, and a lot more. Such reliance on the Internet of Things is resulting in a significant amount of data to be generated, collected, processed, and analyzed. The big data analytics is no doubt beneficial for business development. However, at the same time, numerous threats such as attacks on message and device integrity, the vulnerability of end-devices to malware attacks, physical compromise of devices, and threats to user data security and privacy pose a great danger to the sustenance of Internet of Things. Therefore, it is the need of the hour to develop a security mechanism for the Internet of Things systems to ensure the integrity and privacy of data being processed by these systems.

This study thus endeavors to highlight most of the known threats at various layers of the Internet of Things architecture with a focus on the anatomy of some of the significant attacks. The research also construes a detailed attack methodology adopted by some of the most successful malware attacks on the Internet of Things, including Industrial Control Systems and Cyber Physical Systems. The study further infers an attack strategy of a Distributed Denial of Service attack through the Internet of Things botnet followed by requisite security measures. The illustration of attack methodologies is followed by a composite guideline for the development of an Internet of Things security framework based on industry best practices.

Sequel to the Internet of Things threat modeling, this research investigates the use of blockchain technology to protect the Internet of Things against data integrity and privacy attacks. Hence, to arrive at intelligible conclusions, a systematic study of the peculiarities of the Internet of Things environment, including its security and performance requirements and progression in blockchain technologies, is carried out. Moreover, this thesis also identifies unique challenges to blockchain's adoption in the Internet of Things and recommends a possible way forward.

Based on a systematic and analytical review of blockchain technology, this study proposes a privacy-preserving and secure data sharing framework for smart cities. The proposed scheme preserves user data privacy by dividing the blockchain network into various channels, where every channel comprises a finite number of authorized organizations and processes a specific type of data such as health, smart car, smart energy, or financial details. Moreover, access to users' data within a channel is controlled by embedding access control rules in the smart contracts. The

devised solution also conforms to some of the essential requirements outlined in the European Union General Data Protection Regulation.

Another important contribution of this work is the conception and design of a novel Internet of Things centric consensus protocol with the Internet of Things focused transaction validation rules. The proposed Proof-of-Honesty consensus protocol not only reduces the possibility of Byzantine behavior by block proposers (validator/mining nodes) during the consensus process but is also scalable with low communication complexity. It is believed that the proposed consensus protocol will prove to be a governing factor for the Internet of Things systems considering to adopt blockchain technology.

Correspondingly, the main conclusion of this research and evaluation is that a sensibly selected and carefully designed blockchain-based IoT application can provide some assurance to the users concerning the security and privacy of their data. In this context, the focus should be on developing an IoT-centric consensus protocol with an intelligent misbehavior detection mechanism to detect and identify malicious miner/validator nodes. Moreover, validation of IoT devices' integrity is also an open challenge that requires due attention.

STATEMENT OF ORIGINAL AUTHORSHIP

I, Imran Makhdoom, declare that this thesis is submitted in fulfillment of the requirements for the award of the degree of Doctor of Philosophy in the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, at the University of Technology Sydney.
This thesis is wholly my work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.
This document has not been submitted for qualifications at any other academic institution.
This research is supported by the Australian Government Research Training Program.
This research was also [partially] supported by funding from Food Agility CRC Ltd, funded under the Commonwealth Government CRC Program. The CRC Program supports industry-led collaboration between industry, researchers, and the community.

Signed:

Production Note:
Signature removed prior to publication.

Imran Makhdoom
3 August 2020

STATEMENT OF THE TYPE OF THESIS

The thesis is structured as a single manuscript comprising a combination of chapters. Whereas, the chapters other than the Introduction Chapter, are the compilation of published/publishable work.

ACKNOWLEDGEMENTS

I express my profound gratitude to my supervisors, A/Prof Mehran Abolhasan (Principal Supervisor), A/Prof Justin Lipman (Co-Supervisor), and Dr. Wei Ni (External Supervisor) for their fruitful guidance and support throughout my Ph.D. candidature. Their encouragement and regular interactive meetings made my experience very productive. I am especially thankful to Dr. Mehran for his kind support that enabled me to focus on my research. He was also very supportive in terms of providing assistance to me to attend international conferences.

I am also thankful to the University of Technology Sydney for giving me the opportunity to pursue my Ph.D. studies by granting me scholarships to cover my tuition fee and living expenses. I also acknowledge the handwork put in by the School of Electrical and Data Engineering staff to provide efficient admin and academic support to the research students.

In the end, I would like to extend my gratitude to my parents for their kind prayers, my lovely wife for taking care of me and the kids, and also for bearing with me during hard times.

LIST OF PUBLICATIONS

Journal Papers

J-1. I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu and W. Ni, "Anatomy of Threats to the Internet of Things," in IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1636-1675, Secondquarter 2019. **Research Question 1**

J-2. I. Makhdoom, M. Abolhasan, H. Abbas and W. Ni, "Blockchain's Adoption in IoT: The Challenges, and a Way Forward," Journal of Network and Computer Applications, vol. 125, pp. 251–279, 2018. **Research Question 2**

J-3. I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman and W. Ni, "PrivySharing: A Blockchain-Based Framework for Privacy-Preserving and Secure Data Sharing in Smart Cities," Computers & Security, vol. 88, pp. 101653, 2020. **Research Question 3**

Conference Papers

C-1. I. Makhdoom, M. Abolhasan, H. Abbas and W. Ni, "Blockchain for IoT: The Challenges and a Way Forward," in 15^{th} International Conference on Security and Cryptography (SECRYPT), Porto, Portugal, Jul. 2018, pp. 428-439. **Research Question 2**

C-2. I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman and W. Ni, "PrivySharing: A Blockchain-Based Framework for Privacy-Preserving and Secure Data Sharing in Smart Cities," in 16^{th} International Conference on Security and Cryptography (SECRYPT), Prague, Czech Republic, Jul. 2019, pp. 363-371. **Research Question 3**

C-3. I. Zhou, I. Makhdoom, M. Abolhasan, J. Lipman and S. Negin, "A Blockchain-based File-sharing System for Academic Paper Review," in 13^{th} International Conference on Signal Processing and Communication Systems (ICSPCS), Gold Coast, Queensland, Australia, Dec. 2019, pp. 1-10.

C-4. I. Makhdoom, F. Tofigh, I. Zhou, M. Abolhasan and J. Lipman, "PLEDGE: A Proof-of-Honesty based Consensus Protocol for Blockchain-based IoT Systems," **In press** in the IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, Canada, May. 2020. **Research Question 4**

Patents

P-1. I. Makhdoom, F. Tofigh, “A Method of Electronic Device Integrity Check Based on Device Digital Genome (D2iGen),” Australia Patent 2018204784, Jun. 29, 2018.

CONTENTS

List of Figures

List of Tables

List of Acronyms

ADEPT	Autonomous Decentralized Peer-to-Peer Telemetry
ADS	Alternate Data Streams
AI	Artificial Intelligence
AP	Attribute Provider
API	Application Program Interface
App	Application
Approx	Approximately
ARP	Address Resolution Protocol
ASIC	Application Specific Integrated Circuit
AT&T	American Telephone & Telegraph
Avg	Average
BFT	Byzantine Fault Tolerance
BLE	Bluetooth Low Energy
BNC	Business Network Card
BPDIMS	Blockchain-based Personal Data and Identity Management System
BSN	Biomedical Sensor Network
BSoD	Blue Screen of Death
CA	Certificate Authority
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CASB	Cloud Access Security Broker
CC	Channel Configuration
CCS	Command and Control Server
CERT	Computer Emergency Response Team
CH	Cluster Head
Ch	Channel
CIoT	Cognitive IoT
CMSP	Channel Membership Service Provider
CN	Core Network
CoAP	Constrained Application Protocol
CPS	Cyber Physical System

CSMA	Carrier Sense Multiple Access
CTO	Chief Technology Officer
DApps	Decentralized Applications
DAC	Decentralized Autonomous Corporation
DAG	Directed Acyclic Graph
DAO	Decentralized Autonomous Organization
DARPA	Defense Advanced Research Projects Agency
DBFT	Delegated Byzantine Fault Tolerance
DDoS	Distributed Denial of Service
DKOM	Direct Kernel Object Manipulation
DH	Diffie Hellman
DHT	Distributed Hash Table
DLT	Distributed Ledger Technology
DNS	Domain Name System
DoS	Denial of Service
DPoS	Delegated Proof of Stake
DSS	Digital Signature Standard
DTLS	Datagram Transport Layer Security
DVR	Digital Video Recorder
ECDSA	Elliptic Curve Digital Signature Algorithm
EEPROM	Electrically Erasable Programmable Read-only Memory
eNB	evolve NodeB
EU	European Union
EU GDPR	European Union General Data Protection Regulation
EVM	Extended Verification Module
FCM	Fuzzy Cognitive Maps
Fintech	Financial Technology
FMC	Follow Me Cloud
FTP	File Transfer Protocol
FTTH	Fiber-To-The-Home
GDPR	General Data Protection Regulation
GSM	Global System for Mobile Communications
HDFS	Hadoop Distributed File System
HIPPA	Health Insurance Portability Accountability Act
IAT	Import Address Table
ICS	Industrial Control System
ID	Identity
IDM	Identity Management
IDP	Identity Provider
IDS	Intrusion Detection System
IMA	Integrity Measurement Architecture

IoT	Internet of Things
IoV	Internet of Vehicles
IPFS	Interplanetary File System
IPS	Intrusion Protection System
ISMS	Information Security Management System
ITS	Intelligent Transportation System
JTAG	Joint Test Action Group
KSI	Keyless Signature Infrastructure
LDAP	Lightweight Directory Access Protocol
LMSP	Local Membership Service Provider
LTE	Long Term Evolution
MAC	Medium Access Control
MB	Megabytes
MBR	Master Boot Record
MCMC	Markov Chain Monte Carlo
MEC	Mobile Edge Computing
MITM	Man-In-The-Middle Attack
MMC	Mobile Micro Cloud
ML	Machine Learning
MSISDN	Mobile Station International Subscriber Directory Number
MSP	Membership Service Provider
M2M	Machine-to-Machine
ms	Milliseconds
NC	Network Configuration
NMS	Network Management System
NMSP	Network Membership Service Provider
NIST	National Institute of Standards and Technology
O	Organization
OAuth	Open Authorization
ODS	Ordering Service
OFC	Optical Fiber Cable
OOK	On-Off-Keying
OPC	Open Platform Communications
OS	Operating System
OSI	Open Systems Interconnection
OSN	Online Social Networks
OTA	Over-The-Air
OTAA	Over-The-Air Activation
OWAC	One Way Accountable Channel
OWASP	Open Web Application Security Project
P2P	Peer-to-Peer

P2PKH	Pay to Public Key Hash
PBFT	Practical Byzantine Fault Tolerance
PII	Personally Identifiable Information
PIPEDA	Personal Information Protection and Electronic Documents Act
PKG	Public Key Generator
PKI	Public Key Infrastructure
PLC	Programmable Logic Controller
PoA	Proof-of-Activity
PoC	Proof-of-Concept
PoET	Proof-of-Elapsed-Time
PoS	Proof-of-Stake
PoT	Proof-of-Trust
PoW	Proof-of-Work
PoH	Proof-of-Honesty
PubKeyHash	Public Key Hash
PubKeyScript	Public Key Script
QoS	Quality of Service
RAT	Remote Access Trojan
RCA	Root Certificate Authority
RFID	Radio Frequency Identification
RPL	Routing Protocol for Lossy Networks
RT-IoT	Real-Time Internet of Things
SCADA	Supervisory Control and Data Acquisition
SCC	Small Cell Cloud
SCeNB	Small Cell evolve NodeB
SCM	Supply Chain Management
SDN	Software Defined Network
SED	Self Encrypting Drives
SFTP	Secure File Transfer Protocol
SIEM	Security Information and Event Management
SigScript	Signature Script
SMB	Server Message Block
SMP	Security Management Provider
SoC	System on Chip
SSH	Secure Shell
TCG	Trusted Computing Group
TCS	Trusted Candidate Set
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TOR	The Onion Router
TPM	Trusted Platform Module

TPS	Transactions Per Second
TX	Transaction
UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol
UE	User Equipment
UI	User Interface
UMTS	Universal Mobile Telecommunications Service
USD	United States Dollar
VANET	Vehicular Ad-Hoc Networks
Ver	Version
VPN	Virtual Private Network
V2V	Vehicle-to-Vehicle
VRF	Verifiable Random Function
WAP	Wireless Application Protocol
WebRTC	Web Real-Time Communications
WSN	Wireless Sensor Network
WWW	World Wide Web
XPolM	Cross-Polarization Modulation
XSS	Cross-Site Scripting
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks

”A person who never made a mistake never tried anything new”

- Albert Einstein

1 Introduction

Millions of embedded devices are being used today in safety and security-critical applications such as Industrial Control Systems (ICS), Vehicular Ad-Hoc Networks (VANET), disaster management and critical infrastructure [1]. A massive number of these devices have been interconnected to each other and further connected to the internet to form an Internet of Things (IoT). IoT-based services have seen exponential economic growth in the last five years especially in telehealth and manufacturing applications and are expected to create about USD 1.1-2.5 trillion contributions in the global economy by 2020 [2]. It is also estimated that by 2020, the number of IoT connected devices will exceed to 30 billion from 9.9 million in 2013 [3] and Machine-to-Machine (M2M) traffic will constitute up to 45% of the whole internet traffic [4]. However, due to interconnection with the internet, IoT devices are vulnerable to various attacks, including malware, remote access, Man-in-the-Middle (MITM), storage, and Distributed Denial of Service (DDoS) attacks [1, 5–10]. Moreover, it is believed that IoT devices are being manufactured rapidly without giving much attention to security challenges and the requisite threats [11].

According to [12], more than 85% of enterprises around the world will be turning to IoT devices in one form or the other, and 90% of these organizations are not sure about the security of their IoT devices. Similarly, Joseph Steinberg in [13] has listed many appliances that can spy on people in their homes. A recent study carried out by HP [14] also revealed that 70% of the devices connected to the internet are vulnerable to numerous attacks. Moreover, the development of smart cars is also on the rise in the world, in which vehicle on-board computer systems are connected to the internet, thus making them vulnerable to cyber-attacks [7]. Also, the legacy industrial systems such as manufacturing, energy, transportation, chemical, water and sewage control systems (connected by the IoT to achieve better monitoring, control, and conditional maintenance) have

greater security risks [15]. Attacks on industrial systems are not just a threat; instead, it is a reality, as two Russian security researchers found vulnerabilities in more than 60,000 internet-connected control systems that could be exploited to take full control of the compromised systems running energy, chemical, and transportation applications [16]. Furthermore, it is expected that by the end of 2020, more than 25% of corporate attacks would be because of compromised IoT devices [17]. Similarly, the successful launch of sophisticated cyber-attacks like Mirai [18], Ransomware [19], Shamoon-2 [20] and DuQu-2 [21] on ICS and other critical infrastructure in recent past have rendered existing IoT protocols ineffective.

Moreover, despite centralization and controlled access to data, even the cloud supported IoT is vulnerable to security and privacy issues [22]. Security flaws in IoT are thus leading to attacks on; device integrity, data integrity and privacy, availability of network and attacks on the availability and integrity of services e.g., Denial of Service (DoS) and DDoS attacks [10]. The current security issues in IoT can be attributed to centralized network architecture, lack of application layer security, inadequate standardization on IoT products concerning security i.e., hardware and software, and the wide gap between manufacturers and security analysts. According to IBM Institute for Business value [23], it is critical for the future of IoT that its operational model is revived from costly, trusted and over-arched centralized architecture to a self-regulating and self-managed decentralized model. Such a transformation will provide scalability, reduced cost of infrastructure, autonomy, secure operations in a trustless environment, user-driven privacy, access control and redundancy against network attacks. In this regard, blockchain [24] is being considered as one of the possible mechanisms to realize desired decentralization, data security, and privacy and trustless operational environment [25]. However, blockchain technology must be assessed thoroughly before it can be used securely and efficiently in an IoT environment.

1.1 Motivation

People around the world use smart devices to improve the quality of life by monitoring and analyzing their private data such as health information, smart home environment, smart car operation and management, and daily routine. This data analytics is no doubt beneficial. However, at the same time, IoT devices are vulnerable to a vast number of security and privacy attacks [26]. Additionally, the user data collected by numerous sensors is stored and processed by various Online Social Networks (OSN), smart city control center or various other smart city components such as Intelligent Transportation Systems (ITS), health emergency response, fire and rescue, etc., These components (with mostly centralized control) process user data for the provision of various services to the users and third parties. Although such a centralized control may look effective from the outside, yet it has some significant security concerns.

Centralized control is subject to a single point of failure in case of a cyber-attack or other technical malfunctions [22]. Moreover, it also has trust issues, as the users have to put their trust in the entity that is handling their data. Hence, users have no control over their data assets. Further concerns for user data include: Users do not know where their data is stored, what is happening to it and is there any unauthorized disclosure to the third parties. The above-mentioned users con-

cerns are very much real as the disclosure of personal data leakage concerning millions of users by Facebook Inc. [27, 28] and a bug in Google Plus [29] that resulted in the exposure of personal information of approximately (approx) 500,000 users is a candid example of one of the cloud/OSN vulnerabilities. Hence, we aim to unfold the IoT threat environment and propose a potent defense mechanism that can protect users' data and give users the liberty of controlling access to their private information. Also, the data sharing process should be transparent that provides a clear picture to the data owners about the collection and use of their data assets.

Besides, it is essential to make a clear distinction of data security, data privacy, and the absolute goals of this research. Accordingly, data security generally encompasses three elements, i.e., data confidentiality, integrity, and availability. Whereas, data privacy is another explicit requirement that is mostly mixed up with data confidentiality. But in reality, data privacy is more than just ensuring data confidentiality. Correspondingly, privacy may differ as per individual preferences [30]. E.g., public disclosure of health data or financial information may be fine with some people, but it may be a privacy issue for the most. Hence, data privacy measures such as differential privacy, or user anonymity, go beyond data confidentiality. Therefore, it is stated that this thesis endeavors to devise a solution concerning data security focused on data integrity and controlled access to data. Though user-defined fine-grained access control provides some privacy guarantees, data privacy as a specific element is out of the scope of this thesis.

1.2 Challenges

One of the critical requirements concerning IoT security is to understand and characterize the IoT security threats. Correspondingly, preservation of IoT users' privacy while making the system transparent to ensure accountability of the policy violators is a daunting task. Moreover, the shift from centralized control to a secure and efficient decentralized and distributed system to enable immutable and trustless operation is another challenge that requires considerable research. Similarly, the correct interpretation of IoT users' security and performance requirements and their systematic mapping to blockchain technology is also an essential requirement that needs to be dealt with a meticulous analytical approach.

Besides, since the inception of Bitcoin in 2009, thousands of new cryptocurrencies [31] and numerous blockchain platforms have been introduced. Most of these platforms were developed to resolve some specific limitations of Bitcoin Blockchain, such as energy and computation-intensive Proof-of-Work (PoW) consensus algorithm [24], low TX throughput, utility other than financial value transfer, lack of data privacy, non-availability of identity (ID) and key management and latency in consensus finality. However, from the IoT perspective, still, there are many unresolved issues. The primary challenge is the implementation of user-defined access control to regulate access to the users' personal data and also preserving user privacy. Another critical problem is the non-availability of IoT centric consensus protocol. It also has some embedded issues such as TX/block validation rules, consensus finality, resistance to DoS attacks, low fault tolerance, and scalability concerning high TX volume, protection against Sybil Attack, and communication complexity.

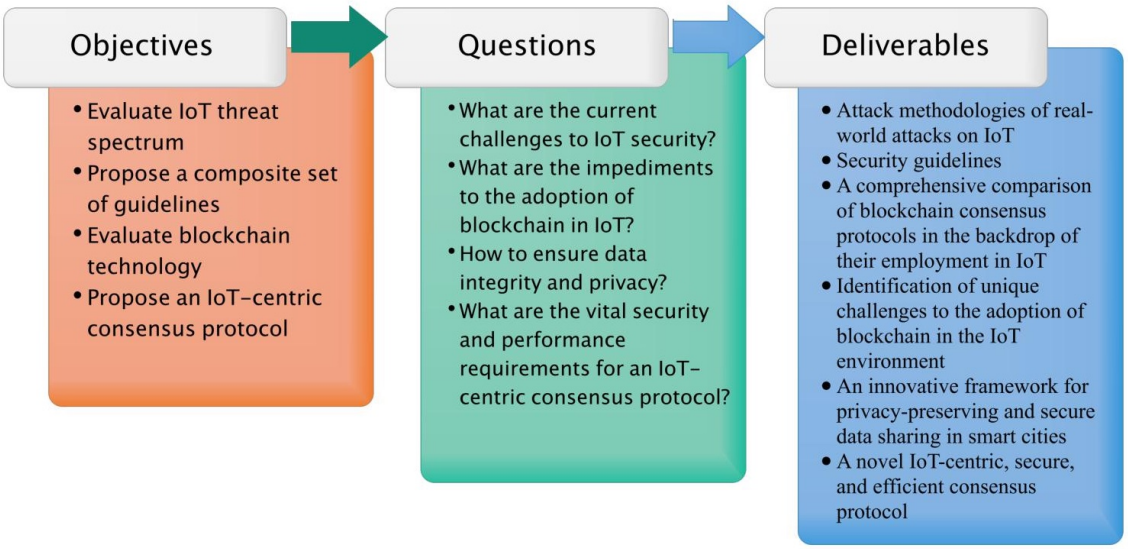

Figure 1.1: Objectives, research questions, and deliverables

1.3 Objectives and Research Questions

As depicted in Figure-1.1, this research aims to analyze the IoT threat spectrum in detail and recommend an appropriate security framework to ensure the integrity and privacy of data in IoT, with the following objectives to achieve:

a. Evaluate the IoT threat spectrum and highlight all the possible threats to IoT.

b. Propose a composite set of guidelines to develop a comprehensive IoT security framework.

c. Evaluate blockchain technology for its capabilities to protect against data integrity and privacy threats in an IoT environment.

d. Propose an IoT-centric consensus protocol for blockchain-based IoT systems with a focus on:

- IoT-oriented TX validation rules.
- Resistance to DoS attacks exploiting weak timing assumptions.
- Fault tolerance against more than $1/3$ faulty nodes.
- Avoidance of Sybil Attack.
- Low communication complexity.

Based on above-mentioned objectives, following research questions have been derived:

a. What are the current challenges to IoT security, and how can they be mitigated?
b. What are the impediments to the adoption of blockchain in IoT?
c. How to ensure data integrity and privacy in the IoT environment using blockchain?
d. What are the vital security and performance requirements that should be considered while designing an ideal consensus protocol for IoT systems?

1.4 Deliverables

The scope of the research established in line with the objectives of this thesis has made the following contributions.

a. Attack methodologies of most of the known real-world attacks on IoT.
b. Security guidelines to help IoT standardization bodies in the design of minimum security standards for IoT systems as per the type of application.
c. A comprehensive comparison of blockchain consensus protocols in the backdrop of their employment in IoT. Additionally, the identification of unique challenges to the adoption of blockchain in the IoT environment.
d. An innovative framework for privacy-preserving and secure data sharing in smart cities.
e. A novel IoT-centric, secure, and efficient consensus protocol.
f. IoT-oriented TX/block validation rules.

1.5 Stakeholders

The thesis will assist the IoT device manufacturers, IoT/blockchain solution architects, blockchain service providers, and application developers in understanding the nature of existing voids in IoT security and taking appropriate security measures. Besides, the IoT security researchers in academia will also be introduced with some open challenges to explore in future research. Moreover, the users of smart devices will be enlightened on the overall cyber threat environment and probable security options.

1.6 Research Methodology

The primary objective of this research is to enhance the security and privacy of user data collected and processed in an IoT setting. Therefore, this research progresses very systematically. As shown in Figure-1.2, the first stage of the research focuses on the review of all possible threats to the IoT. It is followed by the formulation of attack methodology of malware and DDoS attacks, security guidelines based on industry best practices, and finally identification of open research challenges to IoT security. Based on the outcome of Stage-1, blockchain technology is identified as the savior of IoT that can mitigate a majority of integrity threats in IoT. Hence, in Stage-2, blockchain is comprehensively reviewed to ascertain its applicability to the IoT environment. Relative to this requirement, it was deemed essential to determine the right blockchain platform and challenges associated with the adoption of blockchain in IoT systems. Based on the study and experimentation in Stage-2, the requirement for the design and development of a secure and efficient IoT data-sharing framework was determined. Therefore, Stage-3 focuses on the development of privacy and integrity-preserving data sharing and also the design and development of an IoT-centric consensus protocol.

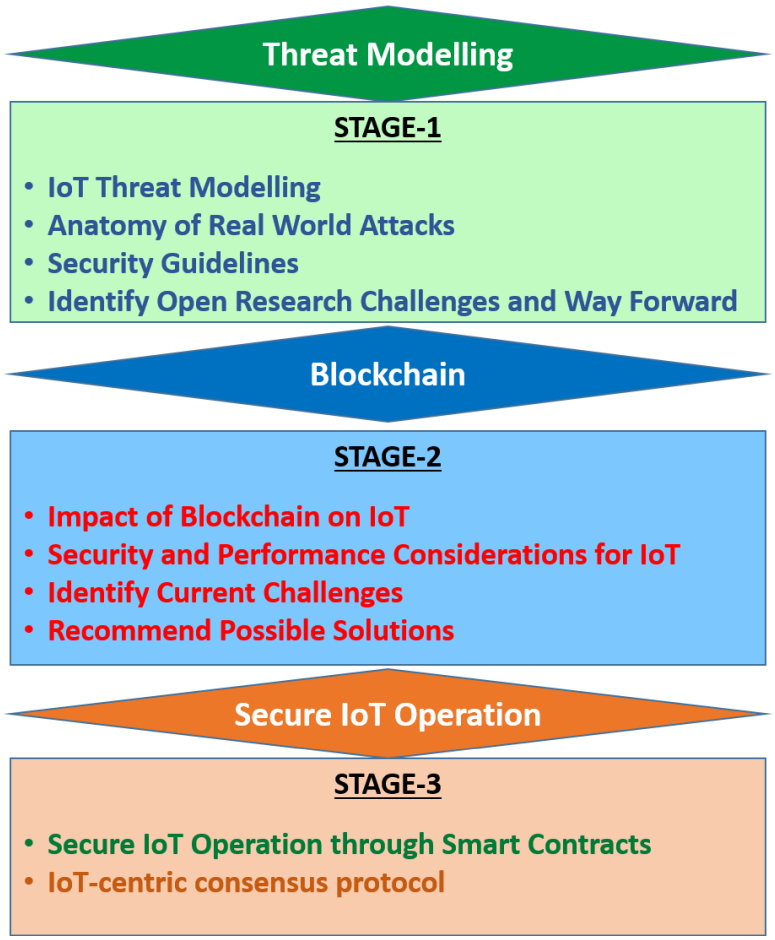

Figure 1.2: Research methodology

1.7 Organization of the Thesis

Based on the proposed research questions, the thesis is divided into seven chapters:

- Chapter-2 presents a comprehensive study on the anatomy of threats to the IoT, including attack methodology of some of the most successful attacks.
- Chapter-3 proposes a defense-in-depth approach for the development of a comprehensive IoT security framework.
- Chapter-4 investigates blockchain technology with a focus on consensus protocols, the impact of blockchain on IoT, and particular security and performance requirements for future IoT systems. It also presents a comparison of various blockchain platforms, current challenges, comprehensive gap analysis, and a way forward to resolve some of the significant challenges.
- Chapter-5 introduces an innovative framework for privacy-preserving and secure data sharing in an IoT setting (smart cities). The objective of this research is to devise a mechanism such that users can control access to their private data in a transparent way. Moreover, the study also proves the efficacy of a multi-channel blockchain network over a single-Channel

blockchain system.

- Chapter-6 presents a unique IoT-centric blockchain consensus protocol that aims to reduce the possibility of malicious behavior by a block proposer (validator/miner node).
- Finally, Chapter-7 summarizes the thesis by highlighting some important conclusions for each chapter and a gist of future research challenges.

”What the Internet of Things is really about is information technology that can gather its own information. Often what it does with that information is not tell a human being something, it [just] does something.”

- Kevin Ashton

2

Anatomy of Threats to the IoT

This chapter comprises eight sections. Section-2.1 presents a gist of existing surveys on IoT security. Similarly, Section-2.2 introduces a generalized IoT architecture and Section-2.3 and 2.4 provide a detailed description of some generalized and various specific threats to different layers of IoT architecture. Correspondingly, attack methodologies of numerous real-world threats to IoT systems are also discussed with a focus on vulnerabilities that can be exploited by a malicious person to realize these threats into successful attacks. Whereas, Section-2.5 highlights security challenges to cloud-supported IoT, and Section-2.6 presents a comprehensive study on the attack sequence and methodology of malware and a botnet attack targeting critical IoT infrastructure. Finally, Section-2.7 focuses on the gap analysis and also proposes a defense mechanism against botnet-based DDoS attacks, and Section-2.8 summarizes the chapter. Most of this chapter has been published as a tutorial paper titled “*Anatomy of Threats to the Internet of Things*,” in *IEEE Communications Surveys & Tutorials* [26].

2.1 Related Work

To date, many reviews and surveys [8, 10, 32–36] have been conducted to highlight the security issues of IoT. However, they do not cover the full spectrum of IoT security (as illustrated in subsequent sections). A detailed comparison of existing work is shown in Table-2.1. Most of the current work focuses on a few aspects and leaves the rest. For instance, [8] refers to limited security issues at different IoT layers and discusses all theoretical/non-industrial security methods without defining an overall security model. Similarly, [10] mostly enumerates the DoS attacks on various layers of Wireless Sensor Network (WSN) and some security vulnerabilities in RFID technology.

It does not give examples of such attacks illustrating the vulnerabilities exploited and also lacks recommendations for protecting against the mentioned attacks. Whereas, [32] highlights some generalized IoT security gaps concerning lack of standardization and regulations by discussing the pros and cons of some existing security frameworks such as COBIT, ISO/IEC 27002:2005. The authors propose an integrated security framework with generalized recommendations on hardware and protocol security with an urge to develop IoT specific security standards.

Authors in [33] also briefly discuss the security and privacy issues in IoT with a focus on some open problems. The researchers broadly cover some of the generalized security and privacy threats including internal and external attacks, DoS attacks, physical attacks and attacks on privacy. Authors also highlight some of the security and privacy challenges to IoT such as user privacy, data protection/authentication, ID/trust management, authorization and access control. Whereas, [34] only covers the security and open research issues related to IoT communication protocols. Similarly, [35] briefly highlights some security and privacy issues of five smart-home devices and proposes an SDN-based network-level security mechanism that monitors and controls network operations of each IoT device.

In another notable work [36], authors present an IoT security architecture comprising three layers, i.e., perception, transport, and application layer. This research comprehensively covers security issues of IoT with a focus on RFID and WSN. The authors also discuss access network technologies including WiFi and 3G. Although authors have amply covered some security issues related to IoT, yet there is room for improvement by including examples of practical attacks/vulnerabilities in IoT such as smart-home and wearable IoT devices. There is a further requirement of adding a comprehensive security framework for IoT. Resultantly, there is a need for a comprehensive illustration of practical threats to IoT and formulation of a set of security guidelines that should cater to varying standards of IoT devices and recommend a common framework for end-to-end IoT security [17].

Contributions of this chapter. To cover the gaps in current literature (as shown in Table-2.1), the major contribution of this chapter is to present an "All in one package" that comprehensively covers most of the aspects of IoT security. The chapter progresses methodologically by first introducing a generalized IoT architecture and a detailed IoT protocol stack showing technologies, protocols and functionalities at various layers of IoT. It amply covers a range of generalized as well as specific threats at different layers of IoT with some related examples. We also present a consolidated list of threats to IoT along with the vulnerabilities that can be exploited to convert these threats into successful attacks. Another aspect that makes this work differs from its predecessors is its due diligence on malware attacks and their attack methodology. We also deduce an attack strategy of an IoT-based DDoS attack followed by necessary security measures.

2.2 IoT Architecture

Currently, there is a lack of consistency and standardization in IoT solutions across the globe due to which there are issues related to interoperability, compatibility, and manageability [37].

Table 2.1: Comparison of existing surveys

Existing Survey	Consolidated Introduction to IoT	Illustration of generalized and threats at IoT layers	Threats to IoT Communication Protocols	Examples of real-world attacks	Security issues of Cloud-based IoT and Fog computing	Malware Threat	IoT Botnets	Defense-in-Depth security measures	Summary of threats to IoT and associated vulnerabilities	Open research issues
[8]	X	Limited security issues at IoT layers	X	X	X	X	X	Theoretical security solutions	X	X
[10]	X	DoS attacks in WSN and some security issues in RFID	X	X	X	X	X	X	X	X
[32]	X	Generalized security gaps concerning IoT standardization	X	X	X	X	X	•Pros and Cons of existing security frameworks, e.g., COBIT, ISO/IEC 27002:2005 •Generalized recommendations for hardware and protocol security	X	X
[33]	X	•Broadly covers generalized security and privacy threats •Internal and external attacks •Physical attacks and attacks on user privacy	X	X	X	X	Simple DoS attacks	X	X	X
[34]	X	X	√	X	X	X	X	X	X	IoT communication protocols only
[35]	X	Security and privacy issues in some smart home devices	X	X	X	X	X	Proposes an SDN-based network level security mechanism	X	X
[36]	X	Security issues in WSN and RFID	X	X	X	X	X	Proposes an IoT security architecture comprising perception, transport and application layer	X	X
This work	√	√	√	√	√	√	√	Proposes a security framework against DDoS Attack and defense-in-depth approach in Chapter-3	Yes in Chapter-3	Yes, in Chapter-3

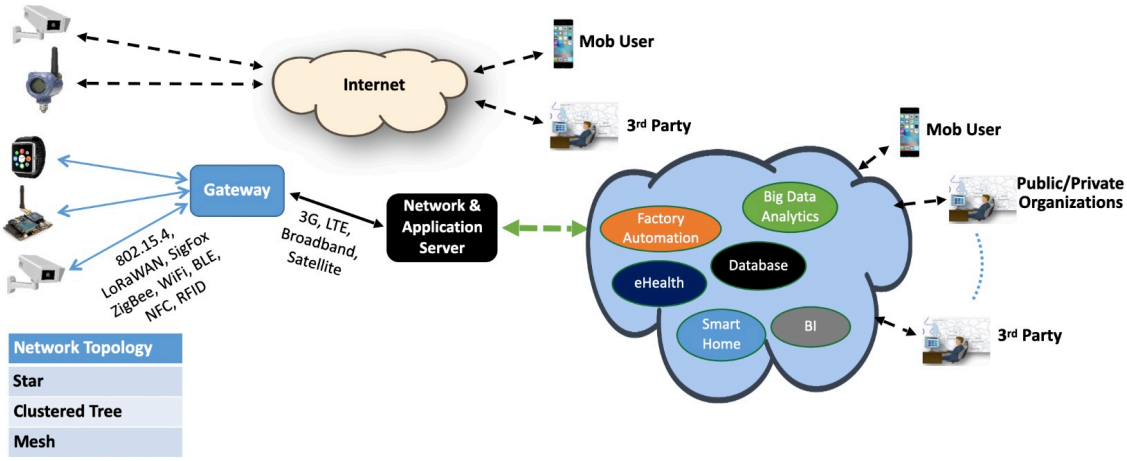

Figure 2.1: Generalized IoT architecture

Likewise, non-uniformity in the presentation of IoT Architecture and layered protocol stack was observed in the literature review [8, 34, 38–46]. Such as, [8] presents IoT layers showing the meagre detail of functionality and the protocols. Similarly, [34] focuses on communication protocols at various IoT layers. Whereas, [38] displays a table of elements and technologies that together form an IoT. Therefore, it is believed that due to this non-standardization, the world has not yet been able to agree on a single IoT reference model [38]. To reduce this non-uniformity, we present a consolidated generalized IoT architecture and a layered IoT protocol stack shown in Figure-2.1 and Figure-2.2 respectively.

An IoT ecosystem may comprise different types of devices, which can be deployed in any of the following topologies, i.e., star, clustered tree, and mesh. "Things" are usually connected to a gateway device using various IoT communication protocols such as 802.15.4, LoRaWAN, SigFox, ZigBee, WiFi, Bluetooth Low Energy (BLE), Near Field Communication (NFC) and Radio Frequency Identification (RFID). The gateway device, which is mostly a full-function device (FFD) is connected to an application or a network server via 3G/4G, LTE (Long-Term Evolution), Optical Fiber Cable (OFC), satellite link, etc. The network/application servers (may be located in the cloud) provide different data analytic services to its users and third parties, including government and private organizations. The processed data is turned into useful information in the form of health statistics, smart home autonomous services, Business Intelligence (BI), industrial automation, environmental monitoring, livable urban communities or smart city sharing services.

As far as IoT protocol stack is concerned, as shown in Figure-2.2, the first layer is the physical/perception layer that consists of sensors, actuators, computational hardware, identification and addressing of the things. As the name suggests, its purpose is to perceive the data from the environment. All the data collection and data sensing are done at this layer [47]. Some other functions of the physical layer include frequency selection, modulation-demodulation, encryption-decryption, transmission and reception of data. The challenges faced by this layer are energy

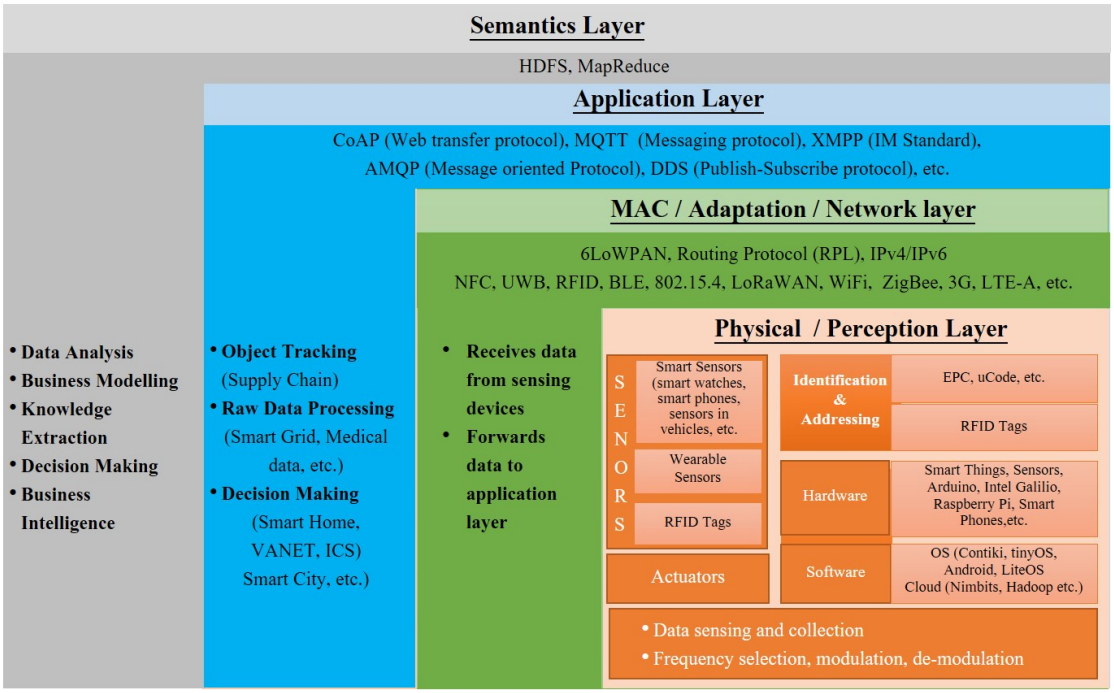

Figure 2.2: IoT protocol stack

consumption, security, and interoperability [37]. The second layer is the MAC (Medium Access Control)/Adaptation/Network layer, which is responsible for receiving data from sensing devices and then forwarding it to the application layer for processing, analytics, and smart services. The network layer also faces specific issues concerning scalability, network availability, power consumption and security [37]. The third layer is the application/services layer which provides smart services to the customers and also feeds processed/aggregated data to the semantics layer. The challenges being faced at this layer are related to handling, storage, and processing of data received from the sensors, security/privacy of user information and conformity to industrial/government regulations. E.g., Health Insurance Portability Accountability Act (HIPAA) in the United States and Personal Information Protection and Electronic Documents Act (PIPEDA) in Canada protect the users' rights concerning their health and personal information. The fourth and the last layer is semantics which can also be termed as a business management layer as it manages all the activities of an IoT system. It implies the use of cognitive technologies to provide certain high-end services such as data analysis, business modeling, strategic decision-making and BI.

Although, by now, we are clear about what IoT is. However, there are many areas in which IoT is different than traditional networks (including LANs and internet), which are being discussed in succeeding paras.

2.2.1 IoT vs Traditional Networks

Before discussing IoT threats, it is important to understand the differences between IoT and traditional networks, as these differences influence the development of requisite security and privacy

solutions for IoT systems. The significant difference between conventional networks and IoT is the level of the resourcefulness of end devices [36]. IoT usually comprises resource constraint embedded devices such as RFID and sensor nodes. These devices have low memory, low computing power, small disk space, and require low power consumption. Whereas, the traditional internet is composed of computers, servers and smartphones that have plentiful resources. Hence, traditional networks can be supported by complex and multi-factor security protocols without little resource consideration. In contrast to this, IoT systems require lightweight security algorithms that maintain a balance between security and resource consumption such as battery life.

IoT devices mostly connect to the internet or gateway devices through slower and less secure wireless communications media such as 802.15.4, 802.11a/b/g/n/p, LoRa, ZigBee, NB-IoT and SigFox. Resultantly, IoT systems are prone to data leakage and other privacy issues. Whereas, on the traditional internet, end devices communicate through more secure and faster wired/wireless media such as fiber optics, DSL/ADSL, WiFi, 4G and LTE. Another difference is that traditional network devices have almost the same operating system (OS) and data format, but in the case of IoT because of application-specific functionality and lack of OS, there are different data contents and formats. Hence, because of this diversity, it is difficult to develop a standard security protocol that fits all types of IoT devices and systems. As a result, a wide range of IoT threats are still at loose and threaten the security and privacy of the users.

If we look at the security design, traditional networks are secured by a blend of static network perimeter defense based on firewalls, IDS/IPS, and the end devices are secured by host-based approaches such as anti-virus and security/software patches. Whereas, the host-based security approach cannot be applied to the resource constraint IoT devices [48]. Nonetheless, there are numerous vulnerabilities in IoT devices such as lack of physical security that includes unguarded distributed deployment, no tamper-proofing, no environmental protection, and plausibility of side-channel attacks. Some other weaknesses include the absence of host-based defense mechanisms (e.g., anti-virus), lack of software updates and security patches, lack of access control measures, cross-device dependencies (e.g., a light bulb is triggered by a light sensor), and lack of IoT-focused attack signatures. Due to said vulnerabilities, the conventional perimeter defense mechanisms cannot protect the IoT devices against insider attacks and physical compromise by unauthorized employees/personnel.

2.3 Generalized Threats

It is estimated that with the rise in the number of things connected to IoT systems to swarming billions of devices by 2020, the potential vulnerabilities will also increase [32]. Hence, the increase in vulnerabilities due to non-standardization of IoT technologies may give rise to security incidents in IoT systems. Some of the most common security issues in IoT are highlighted in succeeding sections.

2.3.1 Security and Privacy Issues

During a security audit conducted by [49], numerous smart devices were checked for security breaches. As per the findings of the security audit, almost 90% of these devices gather personal information about the users in some form or the other. This unauthorized storage of information is vulnerable to data security, privacy, and integrity attacks. Researchers in [9] and [32] have also rendered security issues in IoT a threat to data confidentiality and user privacy. Moreover, the lack of reliable authentication mechanism in IoT devices is also a contributing factor in weak IoT security [10]. Additionally, the lack of data encryption and network access control measures enable an attacker to pose a real threat to user privacy through eavesdropping and traffic analysis [50].

2.3.2 Threats to eHealth IoT Devices

Biomedical Sensor Network (BSN) is a particular case of WSN in which sensors monitor the patients' health and also facilitate chronic disease self-care [51]. BSN has dynamic network topology due to mobile nodes, power constraints, and low bandwidth IoT communication protocols. Therefore, BSN is vulnerable to numerous attacks including DoS, eavesdropping, masquerading, and unauthorized disclosure of personal health information. A successful attack can be life-threatening and can also cause loss of data, misuse of access, loss of personal information, manipulation of data and even in some cases non-availability of critical health services.

2.3.3 Device Integrity Issues

The deployment and successful operation of IoT in critical infrastructures like smart grids, healthcare, ITS, smart vehicles and smart homes are highly dependent on the reliability of devices and the data transmitted between these devices [8]. However, IoT end devices mostly operate in a trustless environment without any physical security. Hence, these devices are subject to physical attacks including invasive hardware attacks, side-channel attacks, and reverse-engineering attacks [52]. In addition, cyber-attacks incorporating compromised IoT devices as bots such as Mirai DDoS attack, are a significant threat to corporate IoT [53].

2.3.4 Software/Code Integrity Issues

Software integrity, including the integrity of the OS, applications (apps), and device configuration, is a key element to guarantee security and privacy of the “Things”. In the recent past a practical manifestation of such an attack was experienced by the world in the form of “Mirai” [54]. This attack created a botnet by hacking into thousands of IoT devices including CCTV cameras and DVRs, by exploiting a firmware weakness. It then directed these devices to launch a DDoS attack on a DNS (Domain Name System) service provider named Dyn.

It is believed that the lack of anti-virus/malware detection mechanisms in IoT leads to attacks on the integrity of the code/software of an end device [8, 9]. The mobile apps are another

source of malware in smart devices that further corrupt the computer networks through infected emails, documents, and direct connection. In 2016, approximately 1 million Google accounts were hacked through an Android malware called "Gooligan." The malware propagated through eighty six seemingly legitimate apps [17]. Therefore, IoT devices need to be protected against malware attacks such as trojans, viruses, and other runtime attacks [9].

2.3.5 Issues Concerning Communication Protocols

Further challenges in security design of IoT arise from the fact that most of the current wireless communication protocols adhere to the Open Systems Interconnection (OSI) layered protocol architecture, and the physical layer encryption is not complemented with additional security mechanisms in the upper layers of the communication [55]. A Man-in-the-Middle (MITM) attack launched by spoofing the Address Resolution Protocol (ARP) at the MAC layer is an example of such a security breach. Moreover, researchers in [56] have identified that cross-layer and hybrid security issues are open research challenges in wireless communications. These issues can be easily extended towards IoT and Cyber Physical Systems (CPS). The same has been demonstrated through various security breaches such as maliciously gaining unauthorized access to a Mitsubishi vehicle through a brute-force hack of the pre-shared WiFi key, exfiltration of private/sensitive data from a computer through a covert FM Channel [57], and hacking of wirelessly controlled implantable medical device [58].

Similarly, cellular technologies such as Universal Mobile Telecommunications Service (UMTS), Global System for Mobile Communications (GSM), and Long Term Evolution (LTE) also suffer from specific security issues [59]. Due to the open implementation of radio baseband stacks, the mobile networks have an added threat of hacking and cyber-attacks. Moreover, GSM and UMTS networks are vulnerable to "International Mobile Subscriber Identity (IMSI) Catching" by an active attacker. Also, there is a time delay in setting security contexts while a User Equipment (UE) is connected to the base station. Such a delay may prove fatal for delay-sensitive applications, e.g., autonomous cars, smart medical instruments, etc. Mobile networks are also vulnerable to DoS attacks launched by mobile bots [59]. The mobile bots may attack the Mobility Management Entity (MME) and Home Subscriber Server (HSS). Correspondingly, radio interface jamming is the DoS attack specific to wireless communications. A smart jamming attack can be launched against 3rd Generation Partnership Project (3GPP) specified mobile networks by using mobile botnets, in which control channels essential for the overall operation of the radio interface can be selectively blocked. DoS attacks are even a threat to 5G networks.

Furthermore, the short-range wireless technologies like Bluetooth and Zigbee are not suitable for applications that require long communication range with low bandwidth. Although cellular technology does provide long coverage for M2M communication, but it requires more power [60]. Therefore, since 2015, Low Power Wide Area Network (LPWAN) is considered to be a suitable technology for the applications that require wide-area coverage, low energy consumption, Quality of Service (QoS), low data transmission rate, low latency and low

costs [60, 61].

Koushanfar et al. also illustrate that communication protocols are subject to protocol attacks, including MITM and DoS attacks [62]. A manifestation of one of the DoS attacks on the wireless communication protocol 802.11b is presented in [63]. The author highlights the vulnerability in the exchange of a disassociation message between the client and the station. It is identified that the message is sent without any authentication. Hence, it enables an attacker to initiate a disassociation message on behalf of other users to stop them from connecting to the network. Correspondingly, this DoS can result in a severe availability issue in the case of a CPS/IoT system [64]. It can further be deduced that almost all communication protocols such as 802.15.4, Zigbee and LoRaWAN provide conventional cryptographic security assurances such as confidentiality, data integrity, data authenticity, replay protection and non-repudiation [34, 40]. However, the cryptographic security embedded in communication protocols is not meant to protect against node compromise and malware attacks.

There is another upcoming communication technology developed by IEEE 802.1 Time Sensitive Networks (TSN) Task Group (TG) for applications requiring Ultra-Low Latency (ULL). TSN promises a secure end-to-end network connection between a sender and receiver node through a time-sensitive capable network [65]. Similarly, Internet Engineering Task Force (IETF) is also working on Deterministic Networks (DetNet) to interconnect the isolated Operational Technology (OT), i.e., CPS with IT networks [66]. However, such interconnection will expose the CPS to various internal and external attacks. Moreover, being a work in progress, security aspects require due consideration to mitigating the internal and external threats ranging from detNet flow modification to path manipulation and attacks on time-synchronized mechanisms.

Coming over to the core network communications media, mostly OFC interconnects multiple corporate data centers or an ISP with the internet gateway. An optical fiber channel may directly impact an IoT system, e.g., a smart home gateway device is connected to an ISP through a Fiber-To-The-Home (FTTH) connection to provide internet-based remote access to various services being used by the owner of the house. Also, the same connection is used by the vendor for maintenance/remote monitoring of the system. Optical channels are vulnerable to eavesdropping, jamming, and attacks to the availability [67]. An attacker can also eavesdrop on classified/private data by tapping into an optical fiber for unencrypted channels [68] or by cracking the encryption keys that are isolated from the payload and are transferred over the Network Management System (NMS) [69]. Whereas, jamming attacks can be launched by introducing in-band and out-of-band cross-talk [70], and by exploiting vulnerabilities of the alien wavelengths [71]. Some other factors that may degrade an optical channel by launching signal insertion attacks include Mixed Line Rate (MLR) networks, On-Off-Keying (OOK) amplitude modulation and Cross-Polarization Modulation (XPolM) [72].

2.3.6 Hardware Vulnerabilities

IoT devices are being commercially developed with more emphasis on device functionality rather than security. Hence, security features are often added in an ad-hoc manner [73]. Therefore, commercial IoT devices have residual hardware vulnerabilities such as open physical interfaces and boot process susceptibilities, which can be remotely exploited [74]. Whereas, the reliable and safe operation of IoT systems depends on the integrity of the underlying devices in general, and the integrity of their code and data in particular [75].

2.3.7 DoS Attacks

Due to constraint resources such as low memory, low computation power, and low battery consumption, IoT devices are vulnerable to resource exhaustion attacks [33]. These attacks include jamming of communication channels, extensive unauthorized or malicious utilization of critical IoT resources such as bandwidth, memory, CPU time, disk space and change of node configuration. All of these attacks will most likely affect the operational functionality of IoT devices and the non-availability of their services to the respective users.

2.3.8 DDoS Attacks

The analysis of past cyber incidents infer that the vulnerabilities of IoT devices make them an ideal platform to launch DDoS attacks. It has also been disclosed by [76] that 96% of the devices involved in DDoS attacks were IoT devices. Whereas 3% were home routers and 1% were compromised Linux Servers. Correspondingly, it is imperative to highlight the difference between DoS and DDoS attacks. In a DoS attack, an attacker targets the victim system, e.g., an application or a network server, mostly using a single source of the attack. Whereas DDoS attacks are launched using more than one attacking/compromised machines.

2.3.9 Security Challenges Specific to WSN

Chen et al. in [77] have classified threats unique to WSN in the following categories: interruption, interception, modification, and fabrication attacks. Moreover, unauthorized insertion of malicious messages in the network has also been highlighted by [39]. Authors in [36] point out that due to wireless communications media, the process of information collection/sharing can be subjected to eavesdropping, malicious routing and message tampering.

2.3.10 Security Issues of RFID and Bluetooth Devices

Due to lack of physical protection and wireless nature of RFID communications, RFID tag data is vulnerable to confidentiality and integrity attacks [39]. Some other security issues include lack of uniform coding, conflict collision, privacy protection, and trust management between the RFID tag and the reader and between the reader and the base station [36]. Similarly, the use of unpatched

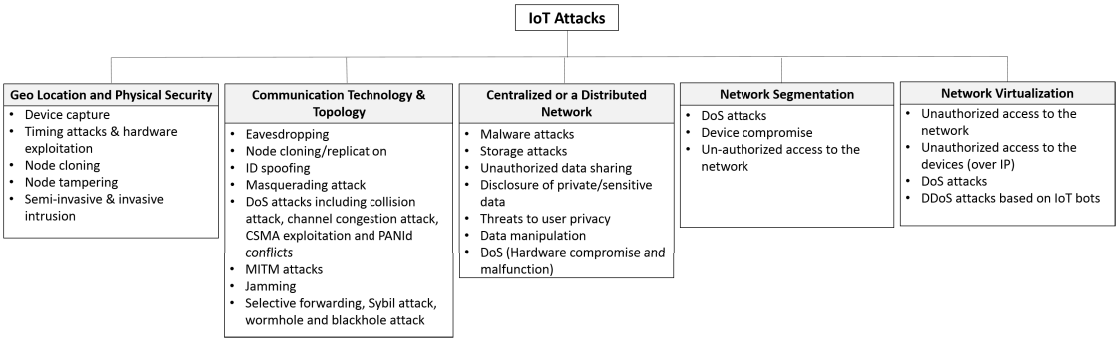

Figure 2.3: Classification of IoT attacks based on their impact on deployment

or old version Bluetooth devices can cause connectivity to unauthorized/malicious devices thus exposing private or security-critical data [39].

2.3.11 User Unawareness

Users are considered to be one of the most common attack vectors. Due to the lack of security training and awareness, employees are vulnerable to social engineering, phishing, spear-phishing, and accidental security breaches. Hence, they unwittingly download malicious codes by clicking infected links in the emails. Also, the sharing of sensitive data over public networks through mobile devices is another cause of security breaches. Therefore, it is estimated that with the increase in smartphone users, almost one-third of the mobile devices are at high risk of exposing official data [17].

2.4 Threats at Different Layers of IoT Architecture

Table-2.2 shows a list of numerous threats at various layers of IoT architecture and the vulnerabilities that can be exploited to convert such risks into successful attacks. Moreover, As shown in Figure-2.3, these attacks have also been classified based upon their impact on IoT node deployment and network architecture. Correspondingly, the IoT threats impinge on the geographical (geo) placement/location and level of physical security of IoT devices as per the sensitivity of data and the critical infrastructure. Besides, the selection of IoT communication protocol and network topology is also derived by the threat environment and the requirement of requisite security measures. E.g., if there is a threat of jamming of wireless channels by the attacker, the use of frequency hopping or a spread spectrum technology would be an appropriate response. Similarly, the decision on the network control by a single entity or a distributed control, and other network security paradigms such as the need of network segmentation and network virtualization for better neutralization and mitigation of IoT attacks are also derived by the extent and types of IoT attacks. The detailed description of these threats at different layers of IoT architecture is presented in the succeeding sections.

Table 2.2: Threats to the IoT

Ser	Threat	Vulnerabilities Exploited	References
Generalized Threats			
1.	Eavesdropping and traffic analysis	Lack of encryption and network access control	[50]
2.	Masquerading and unauthorized disclosure of personal information	Weak data security, authentication and authorization mechanism	[51]
3.	Device integrity	Lack of physical security, no tamper-proofing, trustless environment, open physical interfaces, boot process vulnerabilities	[52, 74]
4.	Remote code execution	Lack of host-based or strong network level security	[53]
5.	Software/Code integrity	No malware detection mechanism, weak network and application layer security	[8, 9]
6.	Threats to communication protocols (MITM, unauthorized access, DoS)	Spoofing the ARP, brute-forcing pre-shared WiFi keys, vulnerability in the exchange of disassociation message	[55, 57, 63]
7.	DoS (Resource exhaustion) attacks	Weak network and application layer security	[33]
Physical/Perception Layer			
1.	Eavesdropping	Unprotected communication channel, no encryption	
2.	Battery drainage attacks	Unchecked volume of legal requests, lack of spam control	
3.	Hardware failure/exploitation	Negligence by the manufacturers, Faults (hardware and software) of the developers, Unprotected interfaces (e.g., UART, JTAG)	[8, 78]
4.	Malicious data injection	Weak access control	[22]
5.	Sybil attack	Lack of ID and device management	[79]
6.	Disclosure of critical information	Lack of physical protection for the devices	
7.	Device compromise	Vulnerable physical interfaces, Boot process vulnerabilities	[80, 81]
8.	Timing attacks and hardware exploitation	Open debugging ports	[1, 82]
9.	Node cloning	Lack of standardization on hardware security and tamper-proofing	[83]
10.	Semi-invasive and invasive intrusions	Lack of physical security and tamper-proofing	[84]
11.	Change of configuration/Firmware-version	Weak implementation of cryptographic algorithms	[84]
12.	Unauthorized access to the devices	Use of default or hardcoded username and passwords	[78, 85]
MAC/Adaptation/Network Layer			
1.	Unfairness, impersonation and interrogation attacks	Weaknesses in communication protocols (Channel access scheme), MAC spoofing, weak network access control	[86, 87]

Continued on next page

Table 2.2 – *Continued from the previous page*

Ser	Threat	Vulnerabilities Exploited	References
2.	DoS attacks to include collision attack, Channel congestion attack, battery exhaustion attack, exploitation of CSMA, PANId conflicts	Flaws in medium-access control and communication protocols	[10, 40, 51, 87–89]
3.	Fragmentation attack	Lack of security mechanism in 6LoWPAN	[34, 90]
4.	MITM, eavesdropping	Weak authentication and data security	[22]
5.	Spoofing, hello flood and homing attacks	Weak authentication and anti-replay protection	[10, 91]
6.	Message fabrication/modification/replay attacks	Weak data authentication and anti-replay protection	[22, 92]
7.	Network intrusion and device compromise (remotely using malware)	Weak network intrusion detection/prevention system, weak device access control once the device is operational, inefficient ID management	[8, 93]
8.	Node replication attack and insertion of rogue devices	Weak network and device access control mechanism	[86, 94]
9.	Selective forwarding attack, Sybil attack, wormhole attack, blackhole attack	Weaknesses in network routing protocols	[10, 95]
10.	Storage attacks	Centralized data storage, non-replication of data storage, no protection against malware such as cryptlocker and ransomware	[8]
11.	DoS attacks launched by sending fake/false messages to a node, server or a gateway device	Weak link layer authentication and lack of anti-replay protection	[51, 91, 96]
Application Layer			
1.	Malicious codes	Lack of application/web security, authentication and authorization mechanism	[8]
2.	Software modification	Lack of application/web security	[9]
3.	Brute force and dictionary attacks, escalation of privileges and data tampering	Weak authentication and authorization mechanism	[97]
4.	SQL injection attacks	Injection flaws in SQL/noSQL databases, OS and Lightweight Directory Access Protocol (LDAP)	[98]
5.	ID theft and password/key/session-token compromise	Incorrect implementation of authentication in applications vis-a-vis session management	[97]
6.	Disclosure of sensitive/private data	Insecure web applications and APIs	[97]
7.	Cross-Site Scripting (XSS)	Vulnerabilities in web applications and user unawareness	[99]
Semantics Layer			
ID theft, compromise of user privacy		Lack of data/application security	[100]

2.4.1 Physical/Perception Layer

Some of the significant threats at physical/perception layer include:

a. **Eavesdropping on Wireless Communication.** Attackers can install devices similar to end-nodes in an IoT system to sniff wireless traffic and extract some valuable information about users.

b. **Loss of Power.** A battery drainage attack in which a node is bombarded with a large no of legal requests, thus preventing it from going into sleep or energy-saving mode.

c. **Hardware Failure.** IoT devices installed in ehealth, ITS, smart cities, and smart grids can be termed as the lifeline to the users. Hardware failure due to a manufacturing fault or as a result of a cyber-attack may lead to substantial damage to the system and physical impairment to the users [8]. In such an endeavor, researchers from security consultancy Rapid-7 [78] discovered that seven commercially available smart devices are vulnerable to cyber-attacks. These devices include the Philips In.Sight wireless baby monitor, iBaby Monitor M3S/M6, Summer Infant Baby Zoom, TrendNet WiFi Baby Cam, Lens Peek-a-View and a Gynoii device.

   In some cases, attacks were as simple as guessing or switching out sections of web addresses/URLs. In the particular case of iBaby M6, it was possible to guess the serial number of the device, camera type, and the user ID. These parameters were then used in the web login URL to execute an authentication bypass access to the device. In a similar attack, the researchers were able to initiate video and audio streams in a Philips camera. In general, there was no blacklisting or whitelisting of IP addresses to control access to these URLs. The researchers were also able to register a new user account for the Summer Baby Zoom Camera by manipulating the URL related to Summer Baby WiFi Monitor and Internet Viewing System without any disclosure/alarm to the legitimate users.

d. **Malicious Data Injection by Forged Devices.** Any determined malicious attacker can introduce a forged device in an IoT system to eavesdrop on the radio traffic, inject fabricated messages, or flood the radio channels with fake messages to render the system unavailable to the legitimate users [22].

e. **Sybil Attack.** In this attack, a malicious node may present multiple IDs by impersonating other nodes or by generating new fake IDs. In the worst-case scenario, multiple IDs may be generated using a single physical device [79]. The attacker may present all the Sybil IDs simultaneously or one by one in different instances. A Sybil Attack may affect the outcome of a voting-based fault tolerance system or a routing protocol.

f. **Disclosure of Critical Information.** A malicious attacker, say a smart thief continually monitors the wireless sensors traffic of a smart house. Even if the wireless data is encrypted, the reduced data traffic may infer critical information to the attacker that the house is empty. Therefore, he can plan a robbery.

g. **Side-Channel Attacks.** These attacks are based on side-channel information about the IoT device. Such information is other than the plaintext or ciphertext messages, i.e., data about processing time or power consumption of the device in encrypting/decrypting various messages and during the computation of different security protocols like Diffie Hellman (DH) key exchange or Digital Signature Standard (DSS) protocols [101].

h. **Device Compromise.** In a practical manifestation of such an attack, researchers in [80] compromised a smart controller of a home automation system (device setup is shown in Figure-2.4) through an open Universal Asynchronous Receiver-Transmitter (UART) interface. The complete attack sequence is also shown in Figure-2.5. Initially, the researchers collected in-

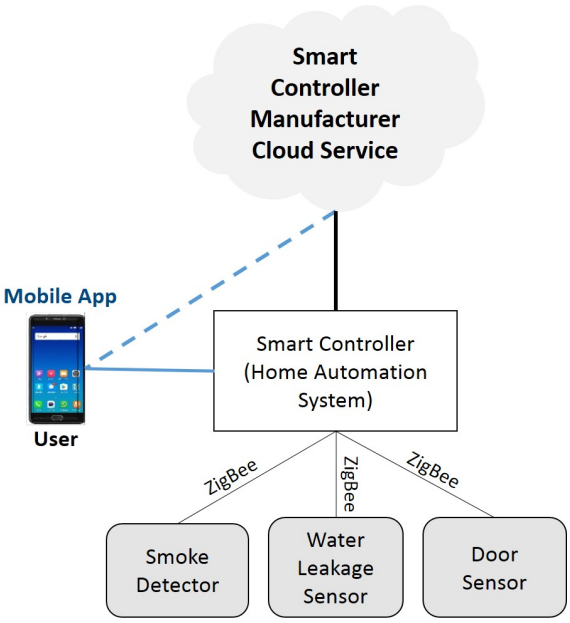

Figure 2.4: Home automation device setup

formation about the smart controller device and identified some weaknesses. Based on the vulnerabilities, the researchers were successful in gaining access to the device. Once inside the device, they were able to view the start-up sequence. They modified the boot parameters and gained low-level access. They also brute-forced the root password and launched network layer attacks such as port scanning and network traffic analysis. In another vulnerability assessment, the researchers were able to modify the ID of a smart meter by compromising the device through a Joint Test Action Group (JTAG) interface. They re-enabled write access to an Electrically Erasable Programmable Read-only Memory (EEPROM) that stored the device ID. As a result of such an attack, the spoofed device ID can be used to feed wrong power consumption data to the smart meter reader. Similarly, owing to the boot process vulnerabilities, the compromise of boot sequence not only facilitates the attackers in attacking other high-level layers but also in taking control of the device. In an experimental setting in [81], a similar attack was successfully executed on Google Nest Learning Thermostat and Nike+ Fuelband SE fitness tracker. The researchers exploited vulnerabilities in the boot process of the Nest Thermostat OS and also some weaknesses in the physical design. The devices were compromised despite the availability of default security features including WPA-2 personal security on WiFi interface, Transport Layer Security (TLS) 1.2 for transmission of any log related data, access to Nest Cloud using OAuth authentication tokens and use of PKCS-7 certificates to ensure authentication and integrity of update images.

i. **Timing Attacks and Hardware Exploitation.** Debugging ports (UART, JTAG, etc.) left open by the manufacturers make the system vulnerable to timing attacks and re-flashing of external memory [1]. E.g., a weakness in Xbox 360 allowed the system to be downgraded to a vulnerable kernel version through a timing attack [82].

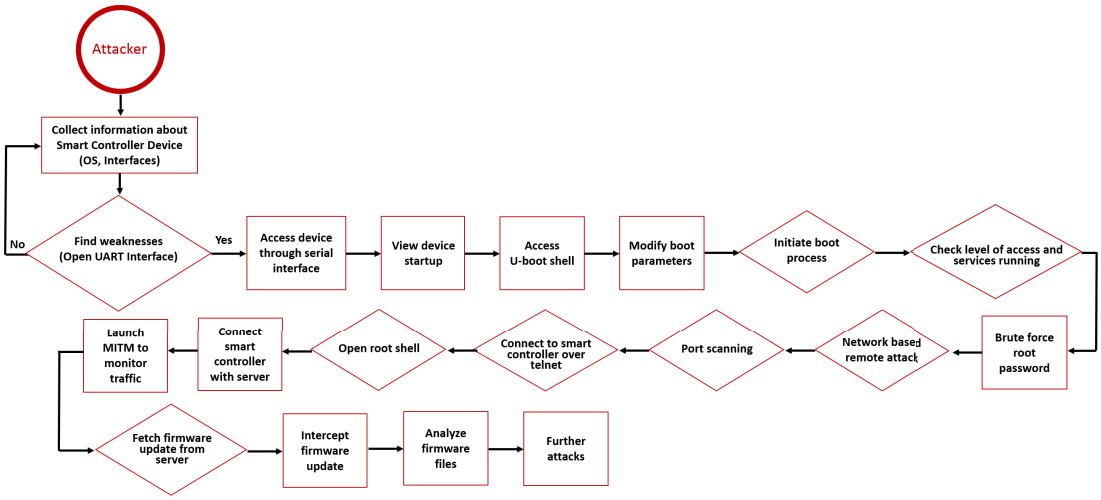

Figure 2.5: Attack sequence of compromising a smart home controller through an open interface

j. **Node Cloning.** Due to the lack of standardization of IoT device design, mostly the IoT devices such as sensor nodes and CCTV cameras are developed without any hardware tamper-proofing. Therefore, these devices can easily be forged and replicated for malicious purposes. This phenomenon is also known as node cloning [83]. It can happen in any of the two phases, i.e., the manufacturing phase, as well as during the operational phase. In the former case, an internal attacker can substitute an original device with a similar, pre-programmed thing for unauthorized purposes. Whereas, during the operational phase, a node can be captured and cloned. Node capture may further lead to the extraction of security parameters and firmware replacement attacks.

k. **Invasive/Semi-invasive Intrusions.** Semi-invasive and invasive intrusions are a serious threat to smart devices, as trusted boot sequence relies on trusted on-chip assets. Since long, encryption/decryption keys and other sensitive information stored on-chip is considered secure. However, today the invasive methods can reveal valuable assets stored on the chip and may compromise any protocol utilizing the secret information. In such an endeavor [84], the researchers were able to extract the stored Advanced Encryption Standard (AES) Key from the internal memory of Actel ProASIC3 FPGA, by launching "Bumping Attacks."

l. **Change of Configuration/Firmware-Version.** Improper implementation of encryption and hash functions threatens the security of the underlying system. E.g., even if a system is secured with robust authentication mechanisms such as X.509 certificate-based TLS, unless the credentials are securely stored, they can be subjected to malicious attacks. Correspondingly, researchers in [102] were able to downgrade the firmware of Sony Play Station-3 by exploiting weak cryptographic implementations.

m. **Unauthorized Access to The Devices.** The use of default passwords by the users and hardcoded usernames and passwords by the manufacturers is a major security vulnerability nowadays. For instance, the iBaby M3S wireless monitor is shipped with a hardcoded username and a password of "admin." Whereas, the hardcoded credentials can only be fixed by a firmware

update from the manufacturer [78]. Moreover, the channels that are left open by the manufacturers for debugging or Over-the-Air (OTA) firmware updates are not always secure. Similarly, the developers may leave some open Application Programming Interfaces (APIs) for accessing the devices at a later time. In such an attack, the Summer Baby Zoom WiFi camera that comes with hardcoded admin access was compromised by the security researchers [85].

2.4.2 MAC/Adaptation/Network Layer

Numerous threats affect security at the MAC layer, such as unfairness, interrogation, impersonation, and Sybil attack [86, 87]. Some of the DoS attacks at this level include collision attack, channel congestion attack [10,88], battery exhaustion attack (by increasing the frame counter value and spoofing of acknowledgment frames) [40, 89], exploitation of Carrier Sense Multiple Access (CSMA) by transmitting on multiple channels [40, 88] and initiation of fake PANId conflicts. At the adaptation layer, there is a likelihood of a fragmentation attack on 6LoWPAN protocol [34,90].

Next comes the network layer, at which most of the attacks are anticipated because it not only connects multiple private LANs but also provides an interface to the internet. Significant threats to security and integrity of the system include MITM, eavesdropping [22], spoofing [10], message fabrication/modification/replay attacks [22], unauthorized access to network, compromise of a device (done remotely using malware) [8], node replication [86] and insertion of rogue devices [94]. Similarly, the threats to the availability of the network/services are; hello flood attack, selective forwarding, Sybil attack, wormhole attack, blackhole attack [10], and storage attacks [8]. DoS attacks can also be launched by sending fake/false messages to a node, server [51], or a gateway device [96].

2.4.3 Application Layer

Security is rarely a preference for the application developers as they focus more on efficiency and service delivery [103]. As a result, the applications can easily be compromised, and their services can be denied to legitimate users. Major threats to the application layer are:

a. **Malicious Code.** Malicious codes spreading over the internet or targeted malware can easily compromise the connected IoT devices by exploiting their unique vulnerabilities, e.g., lack of application security and weaknesses in authentication and authorization mechanism. The infected devices can be used as bots to launch further attacks on other end devices/network applications [8].

b. **Software Modification.** An attacker can compromise an IoT device physically or by remote access and then modify the software or firmware to perform an unauthorized action [9]. The exploitation can be done via binary patching, code substitution or code extension.

c. **Weak Application Security.** Security of application/OS running on an IoT device is of utmost importance. Any weakness in the authentication and authorization mechanism can result in brute force attack, dictionary attack, unwanted disclosure of information, the elevation of privileges, or data tampering. Moreover, the latest application security risks ranked by Open Web

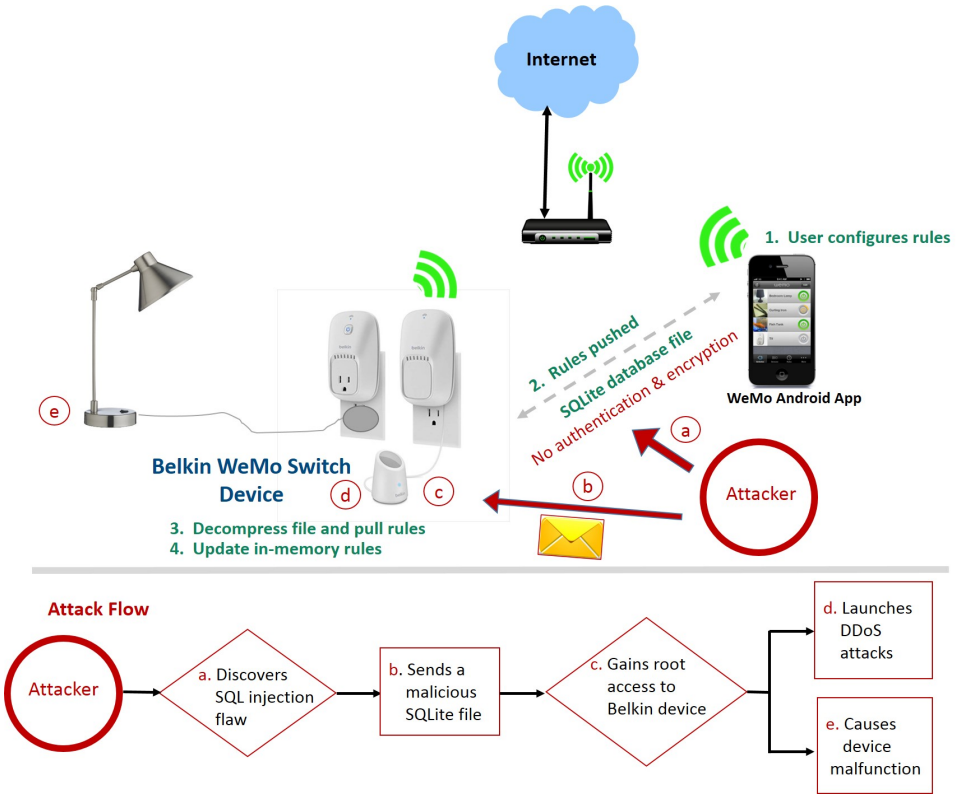

Figure 2.6: Attacking a Belkin WeMo Switch by exploiting an SQL injection vulnerability

Application Security Project (OWASP) [97], pose a valid threat to IoT systems that rely on websites and applications to provide relevant services to their users. Some of these application risks include:

- Injection flaws that threaten SQL/noSQL Databases, OS, and Lightweight Directory Access Protocol (LDAP), pose an equal risk to IoT application and database servers. In such an endeavor security researchers were able to exploit an SQL injection vulnerability in Belkin's smart home products [98]. This vulnerability allows an attacker to inject malicious code into the paired Android WeMo smartphone app and take root control of the connected home automation device. The sequence of attack is illustrated in 5 steps in Figure-2.6, i.e., from a to e. In that, firstly, the attacker discovers an SQL injection vulnerability in the Belkin WeMo Android app. He also discovers that there is no authentication and encryption used for communication with the Belkin device. Hence, anyone can send a malicious SQLite file to the device. He does the same and resultantly gets root-level access to the Belkin device. Once inside, the attacker can launch a DDoS attack or can cause the IoT devices to malfunction. E.g., The lamp is kept on for a long time irrespective of the rules defined by the user. It is imperative to mention here that once an attacker gains root-level access to the device, he can even kill the firmware update process initiated remotely by the vendor. Hence, the device can be kept in the compromised state for as long as desired by the attacker or until the device is updated on the site.

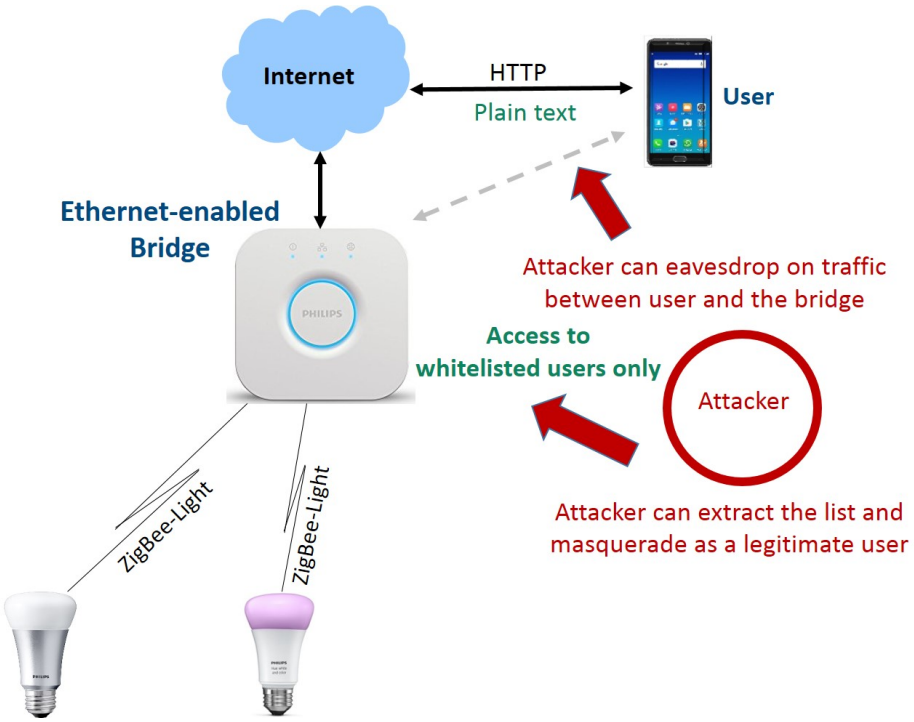

Figure 2.7: Threats to the Philips Hue connected bulb

- Incorrect implementation of authentication in applications vis-a-vis session management allow attackers to steal IDs of other users and compromise passwords, keys, and session tokens. The inability of a user to change the default username and password to access a new device or application is an example of this weakness. This aspect is critical for IoT systems based on smart devices, such as smart cities, smart homes, smart vehicles, and wearable health monitors. An example of such a vulnerable device is the Withings smart baby monitor that allows the users to monitor their babies remotely via a mobile app. However, the video stream sent from the baby monitor to the WiFi Router is in plain-text. Hence researchers in [35] were able to hijack the session using ARP poisoning and gain access to the camera feed.
- Sensitive data exposure due to insecure web applications and APIs pose a significant threat to the confidentiality and privacy of user data collected or processed by IoT devices such as smartphones, wearable health monitors and smartwatches. An example of such a vulnerability is the Philips Hue smart bulb [35]. It enables users to control the lighting system through a mobile app wirelessly. However, the data exchange via HTTP between the app and the Ethernet-enabled bridge that forwards the commands to the smart bulb is in plain text. Hence, any MITM attacker or eavesdropper can sniff the communication between the user and the smart bulb and ascertain the personal habits of the user. Moreover, an attacker can also extract the list of authorized users from the bridge and can masquerade as a legitimate user later. The threat scenario is shown in Figure-2.7.
- Broken access control is due to the lack of restrictions on authenticated users. The same

can be exploited in an IoT system by attackers to access unauthorized functionality or data. Such as change of health monitor's thresholds for generating an alarm/notification.

- Security misconfiguration is one of the most common weaknesses. It implies insecure default configurations, open cloud storage, misconfigured HTTP headers, and overblown error messages that may contain sensitive information. An IoT device is insecure without secure configuration and timely up-gradation of its OS and applications.
- Cross Site Scripting (XSS) is a very prominent threat to web-based applications, and IoT is not an exception. Correspondingly, security researchers were able to exploit a XSS vulnerability in Belkin's smart home products [98]. Such a vulnerability allows an attacker to run an arbitrary JavaScript code in the victim's browser [99]. It can further lead to hacking into the smart device and theft of private data.

2.4.4 Semantics Layer

The creation of the semantics web has transformed the web from human-readable form to machine-processable form. The machine processing has no doubt augmented human reasoning, interpreting and decision-making abilities based on automated Big Data analytics. However, extraction of intelligence or application-specific information from Big Data has its security and privacy issues. E.g., unauthorized disclosure of personal information stored on social media or sensitive health-related data may compromise the privacy of a user. Currently, the tools being used to store and compute big data, such as Hadoop Distributed File System (HDFS) and Google's MapReduce framework, are considered inadequate to protect sensitive data [100].

2.5 Security and Privacy Challenges to the Cloud-Supported IoT

The vision of future IoT is the large-scale integration of various technologies, i.e., sensors, actuators, personal devices such as smartphones, location services, applications, servers, etc. The data originating from a multitude of devices will be available for open sharing across a range of applications, servers, and users. This public sharing is currently achieved with cloud technologies. Over the period cloud computing [104] has evolved to process, analyze and store Big Data. Though, cloud services offer benefits in terms of resource management, scalability [11,105], cost-effectiveness, and shifting of business risks including hardware failures to the infrastructure providers that have better risk management capabilities [106]. However, mostly the IoT systems are developed for a particular application in mind. Therefore, the security aspects are also limited to that specific application with very less or no consideration for security while data is in the cloud and being shared openly across a range of devices. If the legacy IoT systems are connected with the cloud for extended data sharing, i.e., horizontally between things or various applications via the cloud, the IoT sub-systems usually consider and adopt security measures within their sub-networks. However, once the data leaves the sub-group and enters the cloud for wide/open sharing, then numerous issues of security and data privacy emerge. In addition to data confidentiality, there

are other issues in cloud computing concerning the trust mechanism between the service provider and cloud infrastructure provider at various layers of cloud architecture [106].

2.5.1 Security of Data

The cloud usually provides secure communication using TLS/DTLS (Datagram Transport Layer Security). TLS provides communication secrecy (using symmetric key encryption), server authentication (using public key and domain controllers), and message integrity using message authentication code. Now here a question arises that what if the things encrypt the data before it is sent to the cloud. This encryption by things may have the following impacts:

- The cloud provider will not have access to legible data.
- The data cannot be shared publicly.
- The security is to be managed by the things, including complexities of key management, especially, once the old keys are revoked, and new keys have to be generated and issued.
- It will affect scalability and restrict data aggregation and analytics to be performed by the cloud provider.
- Cloud provider is restricted to provide only storage/Infrastructure as a Service (IaaS).

2.5.2 Handling of Heterogeneous Data

IoT applications deal with a large amount of widely distributed data gathered from sub-systems based on a multitude of devices like WSN, RFID, smartphones, GPS, etc. Such diversified data may exist in different formats, hence, demanding appropriate data fusion before the cloud can analyze it. However, integration and fusion of such a heterogeneous data may create privacy-related issues [22].

2.5.3 User Anonymity vis-a-vis ID Management

In a cloud-supported IoT, drawing a balance between user anonymity and ID management for authentication, authorization, and audit is a big challenge. E.g., in eHealth applications, the health-related data of patients are provided to various organizations for data analytics and the development of future policies on health issues. The importance of such use of patient data for improving health care cannot be denied. However, it always raises security and privacy concerns for the patients. Hence, various user anonymity techniques are being practised to disassociate the ID of the patients from the health data. But at the same time, to ensure the security of the cloud-based health services, user authentication is equally essential for restricting network access to legitimate users only.

2.5.4 In-Cloud Data Sharing

The vision of future IoT is extensive sharing of data across a range of devices and applications, which can only be achieved with a policy on protection and sharing. Otherwise, if things' data

is stored on the cloud and isolated from other devices [105], the data processing incorporating multiple streams may not be possible, and it may also affect the efficient data analytic services by the cloud provider. Furthermore, it is estimated that at least one-fifth of the documents uploaded to file-sharing services contain sensitive information and 82% of cloud service providers ensure data security during transmission. However, only 10% encrypt data, once it is stored in the cloud [17].

2.5.5 Large-Scale Log Management

In a cloud-supported IoT, there would be a huge number of heterogeneous devices such as sensors, smartphones, smart controllers, etc. Therefore, logging and audit of the network may be challenging. A few of these challenges may include: What does the cloud provider must record. If the log is decentralized, then there would be variations in what is recorded on different systems, and resultantly there would be different interpretations of the logged data [107]. Moreover, insufficient logging and monitoring coupled with missing or ineffective integration with an incident response may result in implausible auditing and accountability thus allowing attackers to launch further attacks on the system. No doubt, most breach studies show that time to detect a breach is over two hundred days which is typically detected by external parties rather than internal processes or monitoring [97].

2.5.6 Vulnerability to DoS Attacks

Cloud providers usually implement requisite controls to protect against various cyber-attacks. These checks include vulnerability mitigation by updating the OS, secure computing using Trusted Platform Module (TPM) to protect against malware/code modification attacks, etc. Even if an attack is successful, the isolation mechanisms contain the effects. However, an IoT Cloud is vulnerable to a DoS attack launched from compromised things. Moreover, cloud services are usually designed to scale up/down resources in response to varying demands but are still vulnerable to DoS attacks [108].

2.5.7 The Threat of Malicious Things

The cloud being resourceful and the coordinator between things can augment the security of cloud-based IoT systems. It can detect a malicious thing/node during the validation process. The cloud can also offer a protective security measure by triggering software/firmware updates where deemed necessary and resultantly sending control messages to the things to revoke them from the network or turn them off. However, there are some challenges involved in determining/detecting the malicious nodes in a system [109]. These problems may include: What method be used to identify or detect a malicious node, or when to initiate the node attestation procedure. Similarly, if the attestation is based on software/code verification, then will it be a challenge-response protocol or a one-way attestation scheme. Finally, is software-based attestation scheme effective, or there is a need for a hardware-based attestation protocol.

2.5.8 Security and Privacy Issues in Fog Computing for IoT

Cloud security is an essential factor that has adversely affected the development of cloud computing. Cloud's centralized data storage and computing framework present a single point of failure and a concentrated target to the attackers. Hence, to reduce the visibility of end nodes to the external attackers, fog computing enables the data to be transiently maintained and analyzed on local fog nodes thereby, also reducing the processing load, overcoming the bandwidth constraints, and minimizing the latency for time-sensitive applications in IoT [110,111]. Fog computing does compliment the cloud by reducing the latency in data provisioning [112]; however, as it is deployed by different fog service providers that may not be entirely trusted, the devices are vulnerable to be compromised. Fog nodes have distinctive features, such as decentralized infrastructure, mobility support, location awareness, and low latency [113], which make them vulnerable to various security and privacy threats [114, 115]. These threats include ID and data forgery, eavesdropping, MITM attacks, DoS attacks, data and device tampering, Sybil attack and user privacy leakage (ID and location information, social habits, personal details, etc.).

Although a broad spectrum of IoT threats is discussed in preceding sections, however, the most common and ever-evolving of these threats are the malware attacks. Which, if left unattended, will prove detrimental to the security of future autonomous IoT systems. Correspondingly, Bruce Schneier, Chief Technology Officer (CTO) at IBM Resilient states that IoT devices being connected to the internet are vulnerable to ransomware attacks [116]. Recently, in a practical demonstration of such an attack, white hat hackers have developed a first of its kind ransomware that compromises a smart thermostat and then demands a ransom to unlock it [117]. Such a demonstration has shown the possibility of remote code execution on smart devices that can ultimately compromise the complete network, e.g., smart home, smart grid, ICS, smart city. It is, therefore, imperative to understand the malware attack methodology to conceive a robust defense mechanism.

2.6 Malware Threat

The history of computer viruses goes back to 1981 when the first “In the Wild” computer virus named Elk Cloner targeted Apple-II systems [118]. Moreover, since the commercialization of the internet in the early nineties, there has been a considerable rise in cyber-attacks around the world. This number has drastically increased since the start of the twenty-first century. The same can be observed in Table-2.3 that shows the trends in cyber and malware attacks over the past thirty-eight years [119–121]. It can be seen that from 2017 onwards, attackers have mostly preferred non-malware techniques such as side-channel attacks, weaknesses in communications or authentication protocols, weak user credentials, etc., to compromise and exploit target systems. However, still, we cannot rule out the possibility of malware/cyber attacks on the internet-connected IoT devices/systems. Hence, it is essential to analyze the functioning and attack methodology of some of the significant malware.

Table 2.3: Trending in cyber/malware attacks

Malware Type	1981-1990	1991-2000	2001-2010	2011-2016	2017	2018	2019
Virus	10	7	3	-	3	2	1
Worm	1	2	27	1	-	1	-
RAT + Rootkit	-	-	21	12	3	2	-
Botnet	-	-	2	2	-	-	
Ransomware	1	-	-	16 [119]	1	1	-
Others (Side-channel, insecure protocols, insecure user credentials, remote code execution, etc)	-	-	-	-	2	8	3
Total	**12**	**9**	**53**	**31**	**9**	**14**	**4**

2.6.1 Anatomy of Malware

Different types of malware are developed to achieve diverse objectives. Some are research-oriented, and some are released into the wild to attain malicious aims set by the attackers. The malware roaming in the wild can further be categorized as targeted and general threats. Before we go further, it is imperative to clear the difference between a threat and an attack. In the information security domain, a threat can be defined as a constant danger that has the potential to cause harm to an information system, such as malware, application misconfiguration, and humans. Whereas, an attack is the successful execution of a malicious act by exploiting vulnerabilities in an information system. Therefore, in this section, an attack methodology of some of the successful malware attacks is explained. Numerous malware attacks such as NotPetya [122], DuQu-2 [123–126], Cryptlocker [19], Shamoon-1 [127, 128], Shamoon-2 [129, 130], Flame/SKyWIper [131–134], Gauss [133, 135, 136], Icefog [137], Dragonfly-Group/Energetic Bear [138, 139], Red October [140–142], and Night Dragon [131, 143] have been analyzed to derive the attack methodology (discussed in section-2.6.2).

Correspondingly a perceived attack sequence of a cyber-attack based on malware is shown in Figure-2.8. The attacker initially gains information about the target system and then prepares attack vectors as per the identified vulnerabilities. The adversary tries to get access to the target system by sending malware that is disguised as a legitimate application. Once the attacker is successful in injecting the malicious payload, the malware stays in stealth mode until it identifies the target system correctly. The malicious code then executes and downloads additional payload. It can also gain escalated privileges and launch further probes and attacks. Most of the malware, transfer stolen data to a Command and Control Server (CCS) and can also take directions from the adversary for further attacks. The malware is also capable of self-propagation through various means once it identifies some other target systems in the network.

Similarly, some of the significant malware attacks targeting IoT systems, including ICS, CPS, smart devices, and critical infrastructure, are discussed in this section. The evaluation focuses on the attack description, vulnerabilities exploited, attack vectors, propagation mechanism, and effects incurred by respective malware.

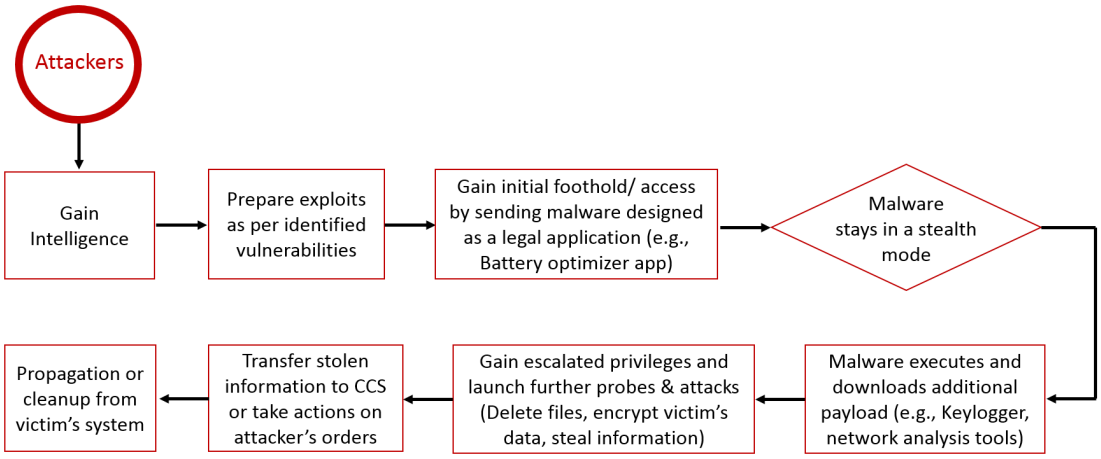

Figure 2.8: Malware attack

2.6.1.1 Xafecopy Trojan

A trojan from the Ubsod family (Blue Screen of Death) was identified in Sep 2017 by Kaspersky Labs as Trojan-Clicker-AndroidOS.Xafekopy [144]. Xafecopy trojan mostly disguised as a battery optimizer app targeted Wireless Application Protocol (WAP) based Android devices. The malicious app subscribes to the victim user's Mobile Station International Subscriber Directory Number (MSISDN) for numerous services on various websites with a WAP billing system that charges directly to the user's mobile bill. This trojan is also capable of bypassing the Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) systems. A modified version of Xafecopy can also send a text message from the user's phone to some premium-rate phone numbers. It can also delete incoming messages from the mobile network provider, and hide notifications about balance deduction by checking for words like “subscription” in the messages. It is also capable of switching a user from WiFi connection to mobile data.

2.6.1.2 WannaCry

It is a typical ransomware, also known as, Wanna Decryptor, WannaCrypt, WanaCrypt0r, and WCry [145]. It was detected in May 2017, and by then it had affected around two hundred and thirty thousand systems including health, telecommunications, transportation, shipping and energy sectors in one hundred and fifty countries. It propagated over the internet and exploited Server Message Block vulnerability (SMB) (MS17-010) in Microsoft Windows 7, 8, 10, and XP systems. It is assumed that it probably spread through phishing emails or malicious websites [146]. Once inside the target system, it would encrypt selected file types before deleting the original files. The malware also changed the Windows wallpaper and displayed a message bearing instructions on how to make the payment in Bitcoins to get the files decrypted. The worm had a kill switch in itself as it looked for a non-existent domain (www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com) to continue exploitation. However, a security expert found out this weakness and created the respective domain thus slowing down the propagation of the malware [147].

Moreover, security researchers in [148], have identified that the ICS is of primary concern in the backdrop of malware, especially ransomware attacks. It is because most of the ICS are always in an operational state; hence, it is challenging to patch them. Additionally, the ICS software and protocols rely on NetBIOS and SMB for operation in a distributed computing environment. Therefore, malware exploiting SMB and NetBIOS vulnerabilities can harm these systems.

2.6.1.3 Cryptlocker

Researchers discovered 4 million samples of this ransomware in 2015. Cryptlocker encrypted files on the target system, thus restricting access of legitimate users to their data. The objective was to get ransom in return for decrypting the data [19]. The attackers used Angler Exploit Kit to find the vulnerabilities that were exploited by the malware. The malicious software is embedded in a pdf document and propagates as an email attachment through Gameover Zeus Botnet using encrypted peer-to-peer (P2P) communication named Kademlia [149]. It is installed in the user profile folder %APPDATA% or %TEMP%. The vulnerable systems and applications include Windows, MAC, Linux, internet explorer and Adobe Flash. Cryptlocker kept its files encrypted which made it difficult for ordinary users to identify the malicious files. Moreover, to avoid forensics, the malware clears itself from the target computer after putting up ransom demand. It is estimated that Cryptlocker inferred a loss of over USD 1 billion in 2016. The gravity of such an attack can be ascertained from an incident in Austria [150], where an electronic lock system installed in a hotel was attacked, and guests were locked out of their rooms. The hotel management had to pay fifteen hundred Euros as a ransom to get the system unlocked by the attackers.

2.6.1.4 Mirai

An internet-based DDoS attack [54] launched against a computer security journalist Brian Krebs's security website through IoT botnet created out of Digital Video Recorders (DVRs) and CCTV cameras. The botnet directed around 620 Gbps traffic towards the website. The attackers exploited the default username and passwords hardwired on the DVRs and CCTV cameras to gain access to these devices by launching a dictionary attack involving sixty-two default usernames and passwords for various account types, such as root, admin, guest, and service. The same malware was also involved in an attack on a French Cloud Computing Company "OVH" [151] and an attack on a DNS provider "DYN" in October 2016. The attack on DYN affected services of some of the significant technology, eCommerce and web giants in the world such as Amazon, Airbnb, PayPal, Visa, Twitter, HBO, CNN, and BBC.

2.6.1.5 Havex

Also known as "Backdoor: W32" and "Havex.A," Havex is an ICS focused Remote Access Trojan (RAT), created to spy on the infected hosts/servers. It targeted websites of three ICS vendors. It also has the potential to cause a DoS attack on Open Platform Communications (OPC) based applications [152]. Attackers used three attack vectors to entice the victims to install the software

on their systems including spam emails, exploit kits and use of watering hole attacks, i.e., software installers on prominent vendors' sites were infected with RAT thus any user downloading the software, or an update would automatically download and install the trojan. The malware exploited the vulnerabilities in vendors' websites to trojanize the software installer. The trojanized installer comprised a malicious file named "mbcheck.dll," which was the actual malware. This file was dropped and executed as part of the standard installation. RAT would then communicate with a CCS and download numerous plugins for further attacks. Various versions of RAT plugins had different tasks like enumerating LAN and listing down connected resources and servers using OPC [153].

2.6.1.6 Stuxnet

A targeted computer worm designed to sabotage CPS installed in the Iranian Nuclear Enrichment Facility was discovered in 2010. It was delivered through an infected USB flash drive. Stuxnet exploited four zero-day vulnerabilities in Windows-based systems to gain an initial foothold. The malware consisted of multiple modules including Windows and PLC rootkits, anti-virus evasion techniques, complexed process injection and hooking code, network infection routines, P2P updates and a CCS interface [154]. Stuxnet specifically targeted PCs running WinCC/PCS-7 control software used for programming the PLCs [155]. It could act as a MITM attacker and mask the malicious code execution by replaying twentyone seconds of legitimate process input signals. It also had the capability of self-propagation by exploiting print spooler and LNK vulnerability (CVE-2015-0096) in Windows-based systems. Stuxnet comprised rootkits that could hide its presence and was also equipped with stolen digital certificates to appear legitimate. The payload altered the frequency converter drives' (from specific vendors including Fararo Paya from Iran and Vacon from Finland) speed to cause physical damage to over nine hundred centrifuges [156]. To contain the threat spectrum of such malware, Microsoft released a security update MS10-061 to fix print spooler and MS-15-018 for Windows shell vulnerability.

2.6.2 Attack Methodology

By analyzing characteristics of numerous malware discussed in the previous section, it can be deduced that in the last decade or so, malware attacks have not only affected the IT infrastructure but have caused physical damage to IoT/ICS as well. Hence, keeping in view the operating mechanism and functionalities of the malware, we have formulated an attack methodology (shown in Figure-2.9). It illustrates all possible steps taken by the attackers in various phases to successfully compromise an IoT system.

2.6.2.1 Preparatory Phase

In this phase, attackers carry out reconnaissance and collect information about the potential target. The information can be obtained through social engineering, corporate websites, and by using various penetration testing toolkits such as Metasploit, Wireshark, Nmap, Social Engineering Toolkit,

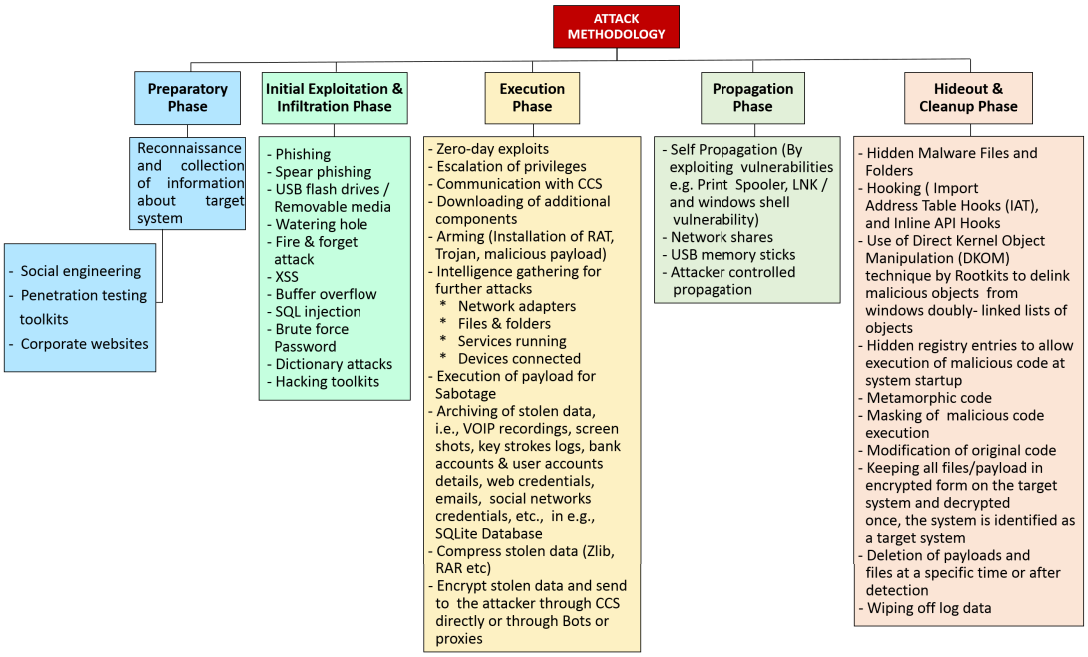

Figure 2.9: Methodology of a malware attack targeting IoT/ICS

Kali Linux, and Nessus. Penetration testing is done to find weaknesses in the target system. The testing can be performed on networks, websites, and servers. Based on this information, attackers plan their attack vectors and develop malware.

2.6.2.2 Initial Exploitation and Infiltration Phase

After gaining information about the potential target, the attackers decide on the type of exploit, its functionalities, and the attack vectors to deliver the exploits to the target systems. In most of the organizations, not only the administrative staff but even the technical staff is not sound on information security practices. Therefore, attackers utilize this weakness and resort to phishing, spear phishing, watering hole attack, and use of infected USB flash drives to gain an initial foothold in the target systems. There are some other exploitation methods as well, such as XSS, buffer overflow, SQL injection, brute force and dictionary attacks for password recovery and use of hacking toolkits.

2.6.2.3 Execution Phase

After intruding into the target system, the attackers can steal information or perform a malicious action either by remote access or through a sophisticated malware that installs a trojan on the host system. The malware can be installed by exploiting zero-day vulnerabilities for which no security update is available, or by attaining root/admin privileges. Most of the latest malware versions keep their files in an encrypted format to avoid detection by anti-virus or any other security mechanism. As soon as the malware identifies the target system based on the particular file system, filename

keywords, pathname or some other attributes, the payload is decrypted and executed.

In many cases, the payload installs a RAT, which then communicates with a CCS and downloads additional components of the payload or other toolkits/exploits. Some of the functions performed by a RAT include intelligence gathering on network adapters, files and folders, services in operation, and connected devices. In addition to espionage, a RAT can enable an attacker to perform any function on the host system from the escalation of privileges to physical damage to the hardware. The RAT is also capable of archiving the stolen data files, VOIP recordings, key logs and financial information. The current breed of RATs uses SQL Lite Database that archives the data in a compressed format. The stolen data is usually encrypted before being sent to the CCS. The data may be delivered directly to the CCS or through bots to increase complexities for later forensics. Some of the most notorious RATs currently in use are; Sakula, Sub7, KjW0rm, Havex (ICS specific), ComRAT (can target ICS), Heseber BOT, Dark Comet, and Shark.

2.6.2.4 Propagation Phase

The common attribute in both, "Targeted" and "In the Wild" malware is the capability to reproduce or to move from the infected system to a new host. Because of this functionality, the malicious software is also termed as self-propagating malware. These malicious programs exploit security vulnerabilities at various levels, i.e., application layer, network layer, and web servers to infect systems and then scan the internet/LAN for more vulnerable systems. Such weaknesses include print spooler, LNK/Windows-shell vulnerability, network shares and USB memory sticks. The installation of the RAT also facilitates attacker-controlled propagation in the victim network.

2.6.2.5 Hideout and Clean-up Phase

Malware use multiple techniques to keep themselves invisible while operating on a victim system. Usually, they keep their files and folders hidden or in encrypted form. The encrypted files are decrypted once the malware reaches the target system or at the time of execution. Malware, such as rootkits, remain invisible by faking the output of API calls through hooking techniques. The hooking can be achieved by intercepting function calls, altering import tables of executables and use of a wrapper library. The two most common methods of hooking being implemented by malware are the Import Address Table (IAT) Hooks and Inline API Hooks. The rootkits also resort to Direct Kernel Object Manipulation (DKOM) technique that hides its processes, drivers, files, and intermediate connections from the object manager/task manager. For clandestine operation, this sophisticated malware is also capable of making hidden registry entries to allow the execution of malicious code at system startup. To remain undetected from anti-virus, the malicious software are designed to be metamorphic, i.e., to re-write their code after each execution. Also, to avoid forensics and reverse engineering, the malware can delete their payload and files at a given time or attacker-controlled instances. They are also capable of removing log data to wipe-off their footprints.

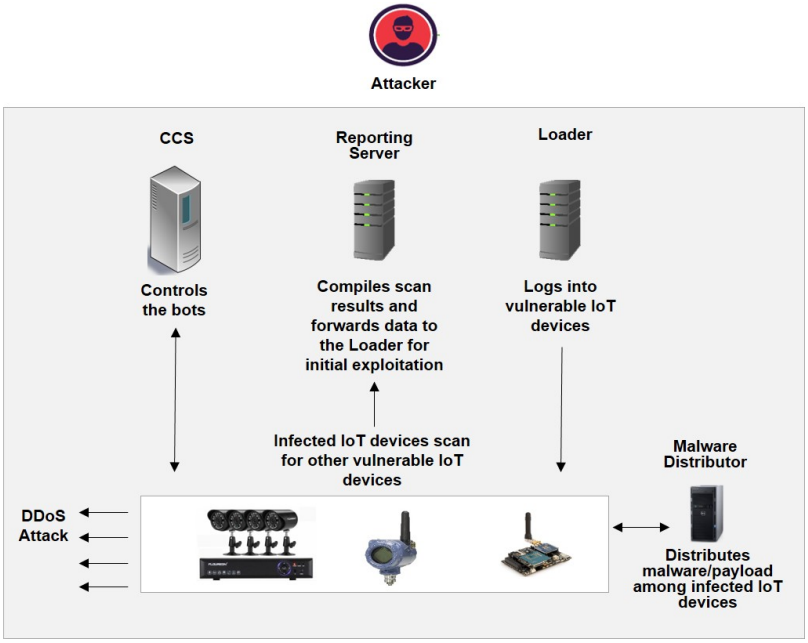

Figure 2.10: IoT botnet

2.7 Gap Analysis and Security Framework

An exponential increase in the number of IoT devices is expected in the next four years. However, due to the lack of secure architecture and weak security mechanisms in commercial IoT devices, these will continue to be a lucrative target for the attackers. Keeping in view the latest trends in malware-based cyber-attacks, there is a high probability that IoT devices may be used to create a botnet army to launch various other attacks such as DDoS and distribution of ransomware/spyware. Based on the malware attack methodology described in Section-2.6, we have deduced an attack methodology of a DDoS attack on IoT devices, which turns the victim devices into bots. One of the probable architecture of a botnet controlled by an attacker is shown in Figure-2.10. A typical IoT botnet [157] comprises a CCS that controls the bots, a Reporting Server that compiles the data about vulnerable IoT devices and forwards it to the Loader module. The Loader gains an initial foothold into the victim devices by exploiting the weaknesses such as hardcoded default login credentials. Once the Loader logs into the victim device, it instructs the victim device to contact the Malware Distributor (MD), a server in the botnet, to download additional malware payload. The infected IoT devices such as CCTV cameras, DVR, smart meters, or sensing nodes are then used to launch DDoS attacks. The chronology of this DDoS attack is shown in Figure-2.11.

In the preparatory phase, the attacker carries out the reconnaissance and finds out specific vulnerabilities in IoT devices. The vulnerabilities may include open hardware ports (UART, JTAG, etc.), weaknesses in the software/OS of the device, weak security implementation, i.e., hardcoded login credentials, weaknesses in the web interface or APIs, and last but not the least open telnet ports. After gaining information about IoT device's vulnerabilities, the attacker plans to get

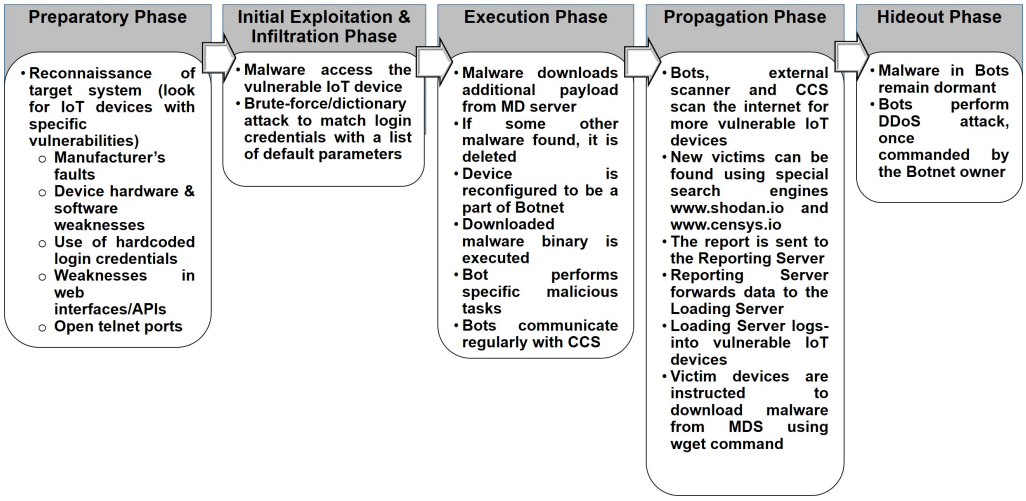

Figure 2.11: DDoS attack on the IoT

an initial foothold into the vulnerable devices by selecting/preparing appropriate exploits. In this case, the exploit can be in the form of malware that establishes a telnet connection with the victim device and logs into the device by using brute-force or dictionary attack to find out the requisite username and password out of the list of probable default credentials that could be used by that specific device manufacturer.

In the execution phase, the infected IoT device downloads additional malware payload from the MD. The malware scans the infected IoT device for other malicious codes, if found, they are deleted, and the victim device is reconfigured to be a part of the IoT botnet. The IoT bot is then used to launch specific attacks such as the DDoS attack on targeted websites or servers. During their lifetime, IoT bots communicate regularly with the CCS and receive instructions for further attacks. The infected IoT devices also scan the internet or the internal network for vulnerable devices and send the scan results to the Reporting Server. In the case of the internet, a list of vulnerable devices can be found using specialized search engines such as *www.shodan.io* and *www.censys.io*. The Reporting Server forwards the list of vulnerable devices to the Loader module, which logs into the vulnerable IoT devices and then instructs them to download additional malware/payload. Usually, the additional payload is downloaded using wget command. The malware can remain dormant to hide its presence and performs the DDoS attack only when commanded by the attacker through CCS.

Based on the above mentioned DDoS attack, which is just one of the numerous threats /attacks scowling IoT, it is evident that current IoT security standards and protocols being implemented by the IoT device manufacturers fail to protect against modern era's sophisticated malware attacks. Although existing IoT communication protocols including CoAP, RPL, 6LoWPAN and 802.15.4 do provide communication security at various layers of the IoT protocol stack (shown in Table-2.4). However, the communication protocols alone, cannot protect against malware/code-modification attacks [34, 40].

Table 2.4: Security provided by the IoT communication protocols

IoT Layer	Protocol	Security Measures
Physical	802.15.4	Nil [34]
MAC	802.15.4	Data confidentiality, data authenticity & integrity, replay protection, access control mechanism [34]
Adaptation	6LoWPAN	Nil [34]
Network	RPL (Routing Protocol for Low Power & Lossy Networks)	Data confidentiality, data authenticity & integrity, replay protection, semantic security, key management [158]
Application	CoAP (Constrained Application Protocol)	Data confidentiality, data authenticity & integrity, replay protection, Non-repudiation [159]

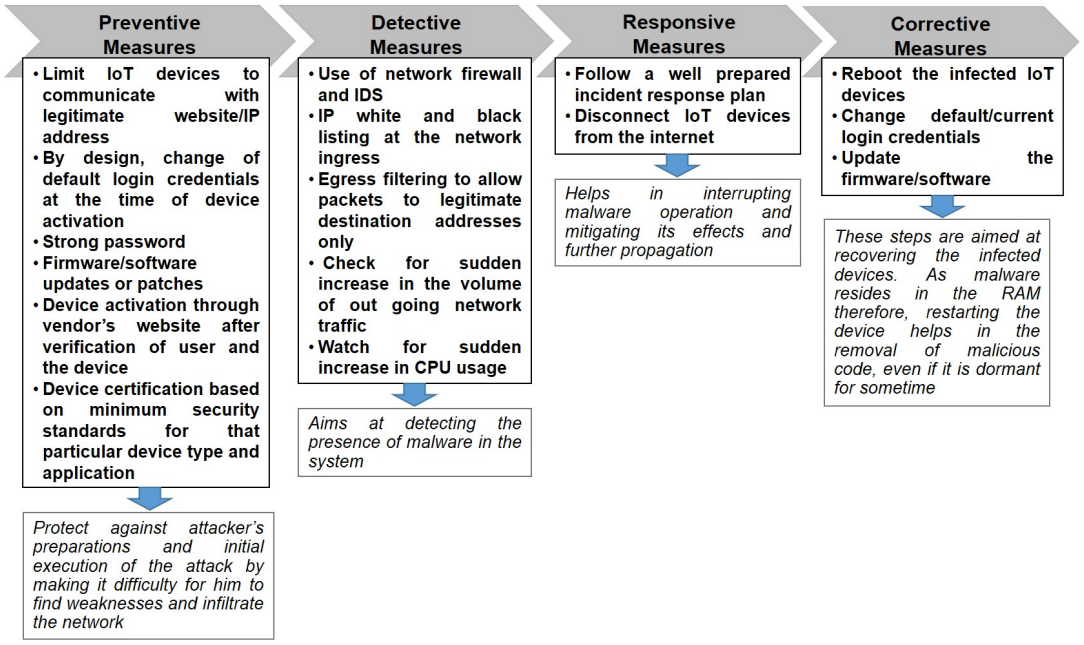

Figure 2.12: IoT security against DDoS attacks

Hence, we propose a security mechanism (shown in Figure-2.12) against IoT botnet malware, comprising preventive, detective, responsive, and corrective measures. In addition to the security measures, the proposed security model also illustrates the impact on an attacker's methodology of attack in various phases, as shown in Figure-2.11. However, in a realistic world keeping in view the plethora of IoT devices' vulnerabilities and related threats as discussed in Section-2.3 and 2.4, the proposed security mechanism shown in Figure-2.12 is insufficient. Therefore, security of IoT ecosystem needs to be dynamic, innovative and wholesome so that it is always one step ahead of the adversaries. A comprehensive security mechanism means proactive approach towards the security of devices, data, applications, networks, and users. Hence, there is a requirement for developing a dynamic IoT security framework that can detect contemporary threats, predict future security events, and respond swiftly to mitigate the risks and take remedial actions.

2.8 Summary

In this chapter, most of the known threats to the IoT systems have been highlighted by quoting examples of related successful attacks. These threats range from simple message interception to sophisticated malware attacks. A comprehensive attack methodology for some of the most significant real-world attacks and an attack strategy of a DDoS attack through the IoT botnet was also discussed. This chapter also proposed a defense mechanism to protect against botnet-based DDoS attacks, followed by a need to develop a comprehensive security framework to prevent a broad spectrum of the IoT threats.

”Security by design is a mandatory prerequisite to securing the IoT macrocosm, the Dyn attack was just a practice run.”

- James Scott

3

Defense-in-Depth Approach

Sequel to chapter-2, in this chapter, a defense-in-depth approach comprising preventive, detective, responsive, and corrective measures is proposed. One of the primary objectives of the conceived security framework is to provide a guideline to the IoT standardization bodies to formulate a baseline security standard for the IoT systems. It is followed by a comprehensive discussion on lessons learned and pitfalls observed concerning the IoT security domain. It is also highlighted that this chapter has been published as a part of a tutorial paper titled “*Anatomy of Threats to the Internet of Things*,” in *IEEE Communications Surveys & Tutorials* [26].

3.1 Guidelines for IoT Security Framework

To prepare a composite set of guidelines for edifying IoT security, we have reviewed the best practices currently being deployed by some of the technical giants of the world such as IBM (IBM Watson IoT), Cisco, American Telephone & Telegraph (AT&T), and Trusted Computing Group (TCG). A graphical illustration of these guidelines is shown in Figure-3.1, and Figure-3.2. Moreover, Table-3.1 glances over some of the compelling security measures and their impact/protection against various threats. These security measures are discussed in detail in the succeeding sections.

3.1.1 Risk Assessment and Threat Modelling

The first step in the development of a security policy for any organization is carrying out the risk assessment for all processes, equipment (hardware & software both), stakeholders and information assets at each layer of IoT architecture. E.g., starting from the manufacturing, transportation, installation, and commissioning stage to the operation and management of the IoT system. The

primary objective of this assessment is to identify what all security incidents can happen in the organization, and subsequently initiating the risk treatment process to minimize the damage of such events. Almost all the information security standards enforce risk management as an integral part of the overall controls.

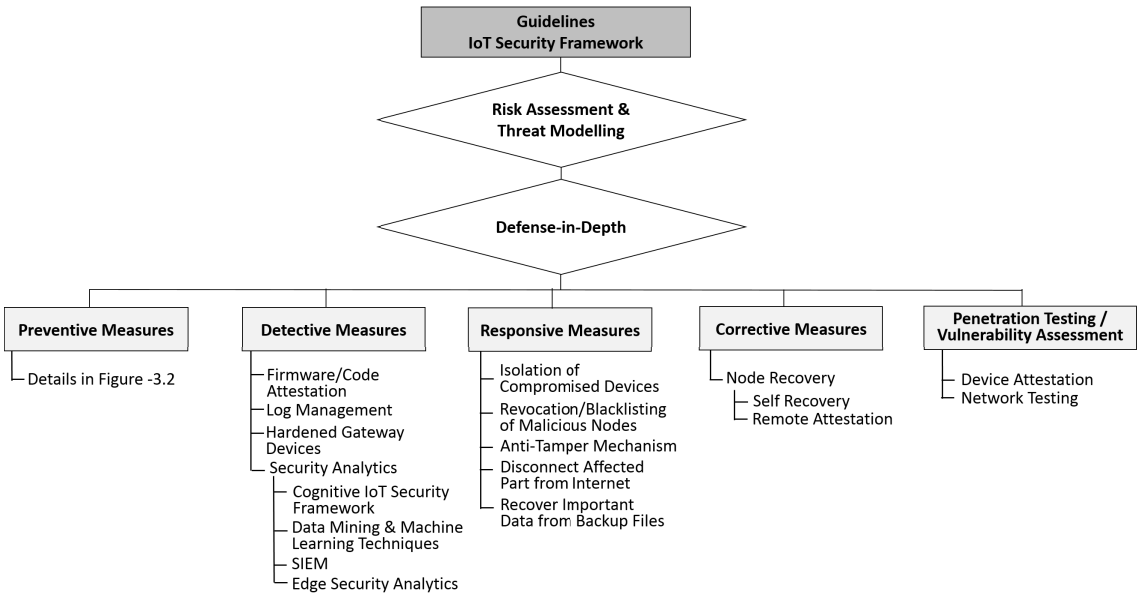

Figure 3.1: Guidelines for the IoT security framework

ISO-27001 [160], an international standard for Information Security Management System (ISMS), outlines seven steps to an effective risk assessment. The first step is about how the organization is going to define its risk methodology. The methodology includes risk ownership, means of measuring the impact of risk on confidentiality, integrity, and availability of data and the method of calculating the effects of the identified risks. The second step involves determining all possible information assets, failure of which can cause some loss to the organization. The third step focuses on the identification of threats and the potential vulnerabilities that can be exploited. In the fourth step, organizations are required to map risk impacts against the likelihood of their occurrences. The fifth step is the most important, as it involves the implementation of measures to avoid, mitigate, transfer or accept the risks. The sixth and seventh step includes preparation of a risk treatment plan and continuous monitoring of the ISMS for dynamic changes to the overall security plan. National Institute of Standards and Technology (NIST) has also issued a special publication 800-30 [161] as a guide to conduct a risk assessment for the security of information systems. Any such standard can be followed until there are some IoT specific standards on board.

3.1.2 Defense-in-Depth

Due to the increasing sophistication and complexity of cyber-attacks, no IT infrastructure can be termed “Safe.” Likewise, a particular security measure cannot prevent 100% attacks. There-

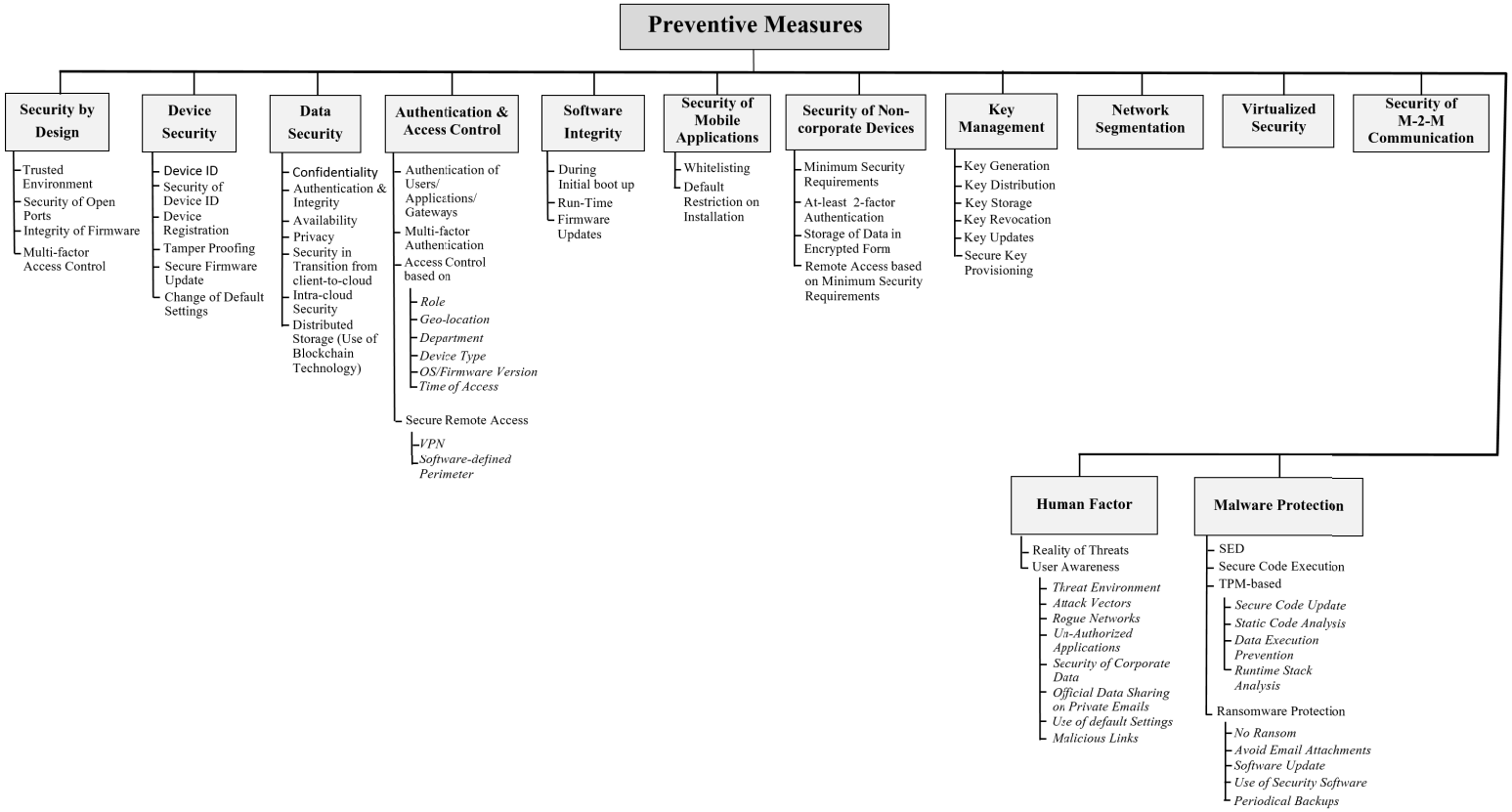

Figure 3.2: Guidelines for the IoT security framework - Preventive measures

fore, the "Defense-in-Depth" mechanism requires substantial preventive, detective, responsive, and corrective actions. However, at the same time, implementation and practice of security measures should not be so complicated that users avoid and go around them. Hence, a comprehensive defense mechanism should be planned based upon risk profiles of the information assets of the organization. Cisco has issued a straightforward and handy defense-in-depth strategy checklist [162] that helps in evaluating the overall security framework of an organization. Moreover, the defense-in-depth approach requires the organizations to take all possible preventive, detective, reactive, and corrective measures. All of these steps are discussed in detail in subsequent sections.

3.1.2.1 Preventive Measures

a. **Security by Design.** The architects of the IoT systems should consider the non-zero likelihood of device compromise while developing security protocols. Therefore, security should be enabled by design, and users should have the leverage to change the security settings as per their requirements [17, 163].

b. **Device Security.** Allocation of a unique device identifier to each IoT device and its continuous validation is essential to ensure platform integrity and controlled access to system resources [164]. The devices should prove their unique ID to set up secure communication with their respective neighbors. The neighbor can be a node, a gateway device, or an application server. The security of device ID against spoofing attacks is critical for sensitive organizations. Moreover, currently, the device ID is required for most of the network security protocols such as IPSec, TLS, and Secure Shell (SSH). Similarly, there should be some mechanism for safe storage of keys, passwords, certificates and other security-critical information on the device that cannot be tampered by the adversary [53].

   To solve the problem of the secure device ID, TCG proposes the use of TPM-based keys as device IDs, which comply with the IEEE standard for local and Metropolitan Area Networks and Secure Device Identity (802.1AR) [165, 166]. The TPM provides enhanced security for device identifiers by protecting these keys in the hardware. Therefore, these keys are protected against unauthorized disclosure during malware and hardware tampering attacks. Another advantage of this technology is that being based on TPM, the cryptographic ID is bounded to the particular device [166], which makes it almost impossible for an attacker to spoof that particular ID using different hardware. However, it is a general opinion that the use of cryptographic identifiers results in privacy issues. Therefore, to avoid long-term user keys/IDs that may lead to unwanted tracking, TCG proposes the use of TPM-based attestation ID keys or direct anonymous attestation.

   There is also a requirement of device registration so that devices can be added or removed as and when required, and only authorized devices are included in the network. The device registration may encompass maximum information about the device such as device ID, its role/capabilities, type, level of security/authorization as per sensitivity of data, public key, software/firmware version, and the authorized period of activation. One of the possibilities to

ensure a transparent and immutable device registry is the use of blockchain technology [53].

IoT devices often operate in an untrusted environment without any physical protection such as traffic light sensors, environmental sensors, agriculture sensors, smart city sensors, and a lot more. Therefore, the end devices in an IoT system should be environmentally rugged and tamper-proof to protect against any malicious forging and access to device hardware. However, in case an adversary tries to tamper the device hardware physically, it should fail safely [167]. Such that upon detection of any tampering attempt, the device memory should automatically wipe off all the data stored in it. This may protect against illegal access to sensitive corporate data, cryptographic primitives (passwords, keys, unique identifiers of neighboring nodes, etc.) or any intellectual property. Some of the embedded systems manufacturers implement end-to-end security in their devices, such as ARM mbed [168] provides secure boot and built-in cryptographic and protocol support to ensure secure network connection. Whereas Juniper Networks [169] make use of the Integrity Measurement Architecture (IMA)/Extended Verification Module (EVM) to detect any accidental and malicious file modifications. The files are attested before they are accessed. The attestation can be done locally or via remote attestation. NXP is yet another manufacturer and developer of various solutions for embedded systems [170]. It offers secure authentication and anti-counterfeiting technology in the form of a tamper-resistant CPU and a secure memory that can store cryptographic keys and a device identifier.

Given the dynamic threat spectrum, the firmware of IoT devices also continuously evolve by installing periodic security and other operational updates. Therefore, it is imperative that all the IoT users especially the critical infrastructure owners such as smart grid, ICS, traffic control systems, nuclear power plants, air travel and railway systems, keep the software/firmware of their devices up to date to protect against any security vulnerability identified by the device manufacturers. Another important aspect of any distributed IoT system based on heterogeneous devices is authenticated and secure broadcast of security updates and control messages.

Similarly, change of default device configuration, especially the security settings such as username and passwords, should be implemented immediately upon the first installation of the IoT devices. In today's hostile environment, security should not be an optional feature; instead, it should be implemented by design [167]. Hence, the device firmware should prompt the user to change the default security settings before it starts functioning.

c. **Data Security.** The security of data mostly refers to the triad of information security, i.e., confidentiality, integrity, and availability of data. To ensure the security of data, organizations must classify their data based on its sensitivity and then grant access to users according to their authorization to access the respective class of data [17]. Moreover, in the current era of IoT, the privacy of data must not be ignored such that personal information should not be disclosed publicly or to an entity that does not have necessary authorizations. In this context, users' data must be handled as per privacy regulations of respective countries/regions such as the General Data Protection Regulation (GDPR) is enforced in the European Union (EU) [171]. Similarly, in this age of data-driven business development policies, the security of Personally Identifiable

Information (PII) in medical and financial records also requires due consideration. IoT business owners or cloud service providers should continuously weigh the utility of users data they are maintaining against the risk of holding it. Whenever the said ratio gets out of proportion, i.e., the risk of keeping large privacy-sensitive user data is more than its further utility; the companies should delete old data. Authors in [167] state that in the case of corporate sector data theft, the unauthorized disclosure of intellectual property may create conflicts in the ownership of such data. To ensure the security of private data, researchers in [15] suggest the use of ephemeral and separate identifiers during communication and while in storage.

In a cloud environment, there should be a secure device-to-cloud interaction. In a similar effort, IBM Watson IoT uses TLS 1.2 for authenticated and encrypted IoT device interactions, which ensure a secure exchange of data over the network. The data sent from the end device to the cloud must be encrypted to preserve the confidentiality of user information [17]. However, the encryption of user data restricts intra-cloud processing and data analytics. To overcome such an issue, the use of homomorphic encryption is recommended [172]. Authors in [17] also suggest the use of a Cloud Access Security Broker (CASB) that not only helps in maintaining a secure link between corporate network and the cloud services provider but also gives organizations insight into cloud applications and services being used by its employees.

Irrespective of the type of storage, data availability to authorized users, is a critical requirement for any organization. Moreover, in the wake of a recent surge in ransomware attacks, the security of relevant personal/corporate data is equally vital. It is recommended that a distributed storage mechanism should be preferred over centralized storage to avoid a single point of failure. Blockchain provides a secure, unforgeable and transparent mechanism for distributed storage, in which every transaction is validated by network consensus [24]. IBM Blockchain [173], Microsoft Azure [174] and Hyperledger Fabric by Linux Foundation [175] are a few examples of multi-purpose blockchain platforms.

d. **Authentication and Access Control.** Authentication for controlled access to an IoT system is not limited to the devices only. The same applies to the applications and gateway devices as well [17, 164]. It is required to protect sensitive information against malicious applications downloaded by users from unauthorized sources. Similarly, gateway devices are to be authenticated to protect against the introduction of a forged gateway device in the network. Depending upon the desired security level, multi-factor authentication may be used, i.e., a combination of password/passkey and a biometric identifier. Moreover, mutual authentication between IoT devices and IoT services/devices can prevent masquerading of IoT services by malicious parties. In addition, it can further help in accountability and forensic analysis.

   Considering the importance of network access control, authors in [176] proposed a traffic flow based network access control. It implements access control based on numerous traffic flow identifiers, such as MAC address, source, and destination address (IP address). Similarly, IBM Watson IoT uses IBM Bluemix that implements role-based controls for users, applications, and gateways to realize the security of data and access to other services/resources [94]. Such a distinction between roles helps in the implementation of unified security policies across the

complete network. In addition to the role, geolocation [177], department, device type, OS/-firmware version and the time of the day at which user seeks access [17] can also form the basis of access control policies.

Correspondingly, authors in [178] propose an ID-based cryptographic authentication scheme without the need for a Key-escrow mechanism to secure M2M interactions in CPS. The scheme saves upon precious computation and communication resources by averting the process of signature generation, transmission and verification. The proposed scheme is also claimed to be robust against MITM, impersonation, replay, DoS, and node compromise attacks. In a similar endeavor, security researchers in [179] have designed a novel mutual authentication and key establishment scheme to secure M2M communication in 6LoWPAN networks. The proposed scheme duly caters to the static as well as the mobile nodes in a 6LoWPAN network. Respectively, [180] suggests a certificate-less anonymous authentication scheme based on hybrid encryption to secure multi-domain M2M communications in CPS. The proposed solution is considered to be tolerant against MITM, replay, impersonation, DoS, and node compromise attacks.

Controlled access to user data by third parties is an important issue. Currently, user data owned by most of the online services are made available to third parties in the form of APIs. The possibility of an unauthorized entity besides the generator of the information and the host service accessing the user information cannot be ruled out. Such an event can result in various privacy and ethical problems. Hence, authors in [181] propose an OAuth-based external authorization service for IoT scenarios. Instead of smart objects/devices storing the authorization related information and performing the computation-intensive verification process, the verification of a request by a service is delegated to an external OAuth-based authorization service. Such an arrangement provides flexibility to the service provider (hosting user data) to remotely configure the access control policies. However, the delegation of authorization logic to an external service demands strong trust between the service provider/smart object and IoT-OAS (OAuthorization-based Authorization Service). There is also a requirement of a secure communication link between the service provider/smart object and IoT-OAS. Moreover, if the smart object directly offers its data as a service, then there is a likelihood of a DoS attack if the smart object receives a large number of simultaneous requests. The proposed scheme is also vulnerable to a MITM attack if the attacker uses an untrusted HTTP/CoAP proxy. In this way, an attacker can not only intercept the communication between endpoints but can also get hold of the authorization information. Based on the apprehended authorization information attacker can spoof the service requester's ID. The scheme also does not protect against a physical compromise of the device.

In another work, to facilitate and securely manage remote access by users to corporate networks/sites, [17] recommends a software-defined perimeter to restrict access to legitimate users. In addition to mere user authentication, such a security perimeter ensures that the user accesses the applications, services and data as per his authorization only.

e. **Software Integrity.** It is to be made sure that only legitimate software is running on IoT

devices, during initial bootup, at runtime, and during firmware updates. Software integrity is one of the important pillars in IoT security as cryptographic algorithms, network security protocols, secure storage, and other such tasks are implemented by software [53].

f. **Mobile Applications.** It is being covered as a separate entity because downloading of mobile applications from unauthorized stores is one of the primary sources of corporate networks' infection. The organizations are advised to enable the installation of only whitelisted apps on corporate devices and should provide a list of the same to its employees for implementation on their devices as well [17].

g. **Security of Non-Corporate Smart Devices.** Increase in the use of smartphones, wearable smart devices such as fitness trackers/bands, smartwatches and smart home appliances including a smart thermostat, intelligent lighting system, smart TV, smart cooling system, smart doors, etc., has added another dimension to IoT ecosystem. It is a common belief that mobile phones, wearable or smart home devices do not contain sensitive information, so they do not require security [17]. Resultantly, manufacturers do not pay much heed towards the security of these devices [11]. Due to this lack of security consciousness, IoT devices have recently been subjected to massive DDoS attacks [54]. It is also viewed that in the future, nation states can sponsor the sale of apparently legitimate IoT devices for cyber espionage [17] or sabotage of target systems.

   Therefore, it is recommended that a minimum security standard should be set for mobile/wearable smart devices with an emphasis on following: Access to device based on at least two-factor authentication, i.e., password and a biometric identifier, limited access to corporate data (only viewing option without any modification rights), storage of sensitive data such as health and financial information in encrypted form. Correspondingly, the corporate networks should provide remote access to those devices only that meet the minimum security requirements. It is also recommended that enterprises should enable mobile access to their systems through VPNs based on multi-factor authentication.

h. **Key Management.** Secure key management is the baseline for the security of any IoT system. It includes key generation, key distribution, key storage, key revocation, and key updates. TCG provides a hardware-based secure key management system that supports various options for provisioning of keys during the IoT device lifecycle, i.e., during chip manufacturing, assembly of the device, while enrolling with a management service, and during owner-personalization. It also provides secure key updates over an untrusted network [166]. Besides, there are other key management systems proposed for IoT systems [182–184].

i. **Network Segmentation.** Network segmentation or segregation is an effective methodology to curtail the impact of a node or a part of a network compromised by an adversary. It not only protects networks and systems of different security classifications but also protects systems of the same classification with varying security requirements. Depending on the system architecture and configuration, network segmentation can be achieved by various methods. Some of these techniques include implementation of demilitarized zones, physical isolation, use of VLANs, software-defined perimeter, application firewalls, application and service proxies, user and ser-

vice authentication and authorization, and last but not the least content-based filtering [185].

j. **Virtualized Security.** The shift from hardware to Software Defined Networks (SDN) has revitalized the flexibility in the implementation of effective security measures. Virtualized security has enabled the protection of data irrespective of its location. Another benefit of this virtualization is that instead of maintaining dedicated hardware for numerous security protocols such as encryption, secure routing, and secure gateways, software-based security solutions can be implemented on a single shared platform. Such a dynamic security solution will enable organizations to enforce security policies with persistence in every type of IoT system, i.e., private or cloud-based IoT architecture.

   An example of SDN-based security enhancement for IoT systems has been demonstrated in [35]. The researchers believe that SDN can be used to augment IoT device-level protections by implementing dynamic security rules at the network level. To achieve this goal, researchers in [35] have proposed a software-based Security Management Provider (SMP) that provides appropriate access control functionality to the users of IoT systems such as smart lighting, smoke alarm and baby monitor, to preserve their privacy and further improve the security. SMP exercises dynamic configuration control over the ISP network and the home router on behalf of the user. It communicates with the ISP network via APIs and also interacts with the IoT system users via GUIs. The proposed security solution thus motivates the manufacturers to concentrate less on User Interface (UI) development and instead focus on the development of APIs that allow a third-party, i.e., SMP, to configure IoT behavior at various layers of IoT architecture.

   In yet another work, [186] proposes an SDN-based security architecture for heterogeneous IoT devices in an Ad-Hoc network. The proposed architecture comprises smart nodes, OpenFlow enabled nodes, OpenFlow enabled switches, and distributed SDN controllers. The multiple SDN controllers are synchronized to provide a granular network access control and network monitoring. Hence, all network devices are first authenticated by the controllers before they start accessing network services as per their authorization.

   Conclusively, it is the SDN controller that monitors and manages all aspects of the network, including security, and the interface between SDN applications and the hardware components [187]. Hence, the SDN controller, being a focal point of all the control activities, can be termed as a lucrative target for the malicious attacks. Thereby, a successful attacker may gain unauthorized access to the controller and insert viruses or malware in the network thus threatening the confidentiality, integrity, and availability of data and other network services [67]. Similarly, authors in [187] also identify various threats to SDN such as unauthorized access, data leakage, data modification and misconfiguration. The authors also highlight the eavesdropping and jamming threats on the physical layer of Software-Defined Optical Networks (SDON). However, they also underline a security measure to protect against eavesdropping and jamming in optical lightpath based on a hopping mechanism. But such a mechanism also suffers some shortcomings concerning the secure exchange of hopping sequence between the transmitter and the receiver and protection against MITM attacks. It is, therefore, imperative to protect SDN against such a single point of failure and attacks on centralized controllers.

k. **Adaptive Security Management.** Most of the IoT applications, such as eHealth monitoring comprising BSN with dynamic network topology, require adaptive security management. Authors in [51] propose a metrics-driven adaptive security management model for eHealth IoT applications. The proposed security model monitors and collects the security contextual information from within the system as well as from the environment. Based on collected data, it measures the security level and matrics, analyzes the received data and responds by changing the security parameters such as encryption scheme, authorization level, authentication protocol, level of QoS available to various applications and reconfiguration of the protection mechanism.

l. **Security of Automated M2M Communication.** In an IoT ecosystem, M2M communication is an important pedestal of industrial and critical infrastructure automation such as power plants, intelligent traffic control system, railways, smart grids, and smart cities. This type of communication ranges from information sharing between robotic/intelligent controllers and smart actuators/appliances to data sharing between smart vehicles. The automated exchange of information between unknown entities must meet the security and privacy requirements. Taking the example of the Internet of Vehicles (IoV), it is recommended that any proposed solution should meet specific security requirements such as data authentication, data integrity, data confidentiality, access control based on authorization, non-repudiation, availability of the best possible communication link, and anti-jamming measures [188].

m. **Protection Against Malware Attacks.** There is an increasing trend in ransomware attacks over the last four years in which the number of attacks has risen to 638 million in 2016 from 3.8 million in 2015 [189] and are still being counted in 2020. As per Symantec Corporation [190], ransomware attacks increased by 4500% in 2014, being too profitable for cybercriminals. Symantec Corporation has proposed few dos and don'ts for the consumers and businesses to protect themselves from such attacks. The preventive measures include: Do not pay the ransom, avoid clicking attachments in unknown emails, keep software up to date, must use security applications, and finally, the most important step is to take a periodical backup of valuable data.

   A common security measure against most malware attacks is not to use hardwired/default usernames and passwords. Also, use only authenticated and encrypted protocols for inbound connections, i.e., SSH for telnet, Secure File Transfer Protocol (SFTP) for File Transfer Protocol (FTP), and https for http. Finally, keep all external interfaces of the administrative connections closed. Security at lower layers should be complemented by application-level access control, use of multi-factor authentication protocols, use of OPC tunneling technologies, installation of update patches, deployment of software restriction policy (application white-listing), white-listing of legitimate executable directories, use of IPSec or VPN for remote access [152], implementation of ingress and egress filtering, restricted number of entry points to ICS Network, maintenance of logs and use of configuration management tools to detect changes on field devices.

   Similarly, numerous security solutions proposed by TCG technologies [165] help to prevent unauthorized access to security-critical programs and data. To solve this issue, Self Encrypting

Drives (SED) based on TCG specifications are in common use for embedded systems such as ATMs, secure mobile phones, corporate copiers, and printers. In these drives, encryption is implemented in the hardware, and data is automatically encrypted in a transparent way to the user. The drives can be safely sanitized for reuse without any need for rewriting multiple layers of garbage data. The user is just required to delete the cryptographic key. As a result, the data stored is made illegible. The hardware-based automatic encryption is termed efficient and secure than simple software-based encryption, which can be turned off anytime by the user [191].

In addition to restricting unauthorized disclosure to sensitive data, the malware should be prevented from execution from the beginning. The two best techniques for this purpose are whitelisting, and execution of manufacturers' signed binaries only. TCG offers TPM-based secure software updates, static code analysis, data execution prevention, and runtime stack analysis. Any combination of such techniques can ensure the integrity of a runtime environment [165]. Although hardware-based security protections are always efficient and more secure than software-based solutions, however keeping in view the cost effect and hardware complexity, these techniques may not be feasible for resource-constrained embedded devices such as wireless sensors and actuators. In such cases, the best way is to program the device to reboot periodically and make use of boot time protections. However, rebooting a sensor or actuator periodically may degrade the performance of resource-constrained devices. Such devices are usually battery operated and have limited energy. Hence, frequent restarts may drain the device's resources. Another limitation of restart-based recovery mechanism is that it can destabilize Real-time IoT systems (RT-IoT) that need consistent actuation with tight timing constraints. To address this issue, authors in [192] propose a runtime restart-based security protocol "ReSecure" for Real-time Systems (RTS) . ReSecure is a blend of hardware and software mechanisms that enable a tradeoff between the security guarantees and control performance while ensuring the safety of the physical system at all times.

n. **Human Factor.** Any level of security is not sufficient until the users of the respective organization are security conscious and believe in the reality of the threats. Correspondingly, an unintended action like connecting an infected USB flash drive to a company's private network can cause a disaster for that enterprise. The organizations should deploy network-wide security policies to implement controls based on authentication, authorization, role and even incorporating geolocation of the users. Enterprises should organize periodic security updates and awareness lectures for its employees, covering the following dimensions:

- Current threat environment.
- Attack vectors being used by hackers/adversaries.
- Implications of sharing sensitive corporate and personal information on public/rogue networks.
- Download and installation of applications/software from unauthorized sources.
- Storage of corporate data in personal laptops/flash storage devices without encryption.
- Use of private email accounts for official purposes.
- Disposal of important official documents in open bins, thus giving an invitation to the

attackers for dumpster diving.

- Use of default settings for smart devices.
- Sharing of sensitive data over social media, that too with default (lowest) security settings.
- Avoid malicious links in unknown emails.

Table 3.1: Security measures and their impact

Ser	Security Measure	Impact / Threat Protected Against	References
1.	Risk assessment and threat modelling	Identification of all possible threats, vulnerabilities and risks. Helps in the development of a risk mitigation plan and formulation of a composite security framework	ISO-27001 [160], NIST Special Publication 800-30 [161], Cisco [162]
Preventive Measures			
2.	Security by design from the vendors (Change of default security settings on device startup, security of all debug ports/interfaces)	Users' unawareness, unauthorized access to the devices through backdoors, firmware and software modification	AT&T [17], IBM [163]
3.	Device ID management	ID spoofing and device replication attacks. Compliments network security protocols (IPSec, TLS, SSH)	IBM [164], TCG [165, 166]
4.	Tamper-proofing of IoT devices	Unauthorized disclosure of cryptographic keys and passwords, modification of code/-firmware and replication/cloning of devices	IBM [164], NXP [170]
5.	IoT device registration and management	Unauthorized or illegal device joining the network	IBM [53]
6.	Secure boot and builtin cryptographic protocols support	Unauthorized access to device and modification of the boot sequence to execute malicious codes	ARM Mbed [168]
7.	Use of Integrity Measurement Architecture (IMA) or Extended Verification Module (EVM)	Accidental and malicious modifications of files	Juniper Networks [169]
8.	Data classification and requisite user authorization	Unauthorized disclosure and access to data	AT&T [17]
9.	Use of ephemeral identifiers for communication and storage of data	User privacy in the context of PII	IBM [15]
10.	ID-based authenticated encryption and mutual authentication schemes for CPS	Impersonation, MITM, eavesdropping, data forgery, replay and modification attacks	[178–180]
11.	Homomorphic encryption	Privacy issues in cloud-based IoT during data processing/analytics	[172]
12.	Cloud Access Security Broker (CASB)	Security issues in cloud-based IoT systems	AT&T [17]
13.	Blockchain technology	Data integrity issues including data modification and forgery, replay attacks, malware attacks targeting data security, integrity and availability such as cryplocker, ransomware and wiper	Bitcoin Blockchain [24], IBM Blockchain [173], Microsoft Azure [174] and Hyperledger Fabric by Linux Foundation [175]
14.	Authentication and access control in applications (including white/black listing)	Downloading of malicious applications	IBM [164]

Continued on next page

Table 3.1 – *Continued from the previous page*

Ser	Security Measure	Impact / Threat Protected Against	References
15.	Endpoint and gateway device authentication and access control	Introduction of forged end/gateway devices in the network by an attacker	IBM Bluemix IBM [94]
16.	Authentication between devices within an IoT system	Masquerading of IoT services by malicious parties. It also facilitates accountability and forensic analysis	
17.	Role-based access control for the users of an IoT system (In addition to role, access control policy can also consider geo location, department, device type, OS/-firmware version and time of the day)	Security and privacy issues related to data and unauthorized access to the network services	IBM [94], Cisco [177]
18.	Ensure software integrity during initial boot up, at runtime and during firmware/software updates	Code modification and malicious code execution	IBM [53]
19.	Security of data in personal IoT devices (Smart watch, smartphone, health monitor, fitness tracker) by using lightweight cryptographic protocols	Unauthorized access/disclosure to personal information	[193–195]
20.	Secure remote access to corporate networks from smart IoT end devices using VPN and limiting access to end devices meeting minimum security standards	Attacks on corporate networks, security issues related to business data/intelligence	US-CERT [152]
21.	Key management (including key generation / distribution / storage / revocation / updates)	Masquerading attacks and device compromise	[166, 182–184]
22.	Network segmentation using Demilitarized zones, physical isolation, VLANs, software defined perimeter, application firewalls/proxies and content-based filtering	Curtail impact of a node or a part of network compromise	Australian Signals Directorate [185]
23.	Virtualized security based on SDN	Augment IoT device-level protection by implementing security at the network level. Hence, reducing burden of cost related to the development of security protocols for low-cost IoT devices for the manufacturers	[35, 186]
24.	Use of self-encrypting devices/drives (SED)	Unauthorized disclosure of data	TCG [165, 191]
25.	Adaptive security management	Provides dynamic re-configuration of security parameters	[51]
26.	Execution of signed binaries, TPM-based secure software updates, static code analysis, runtime stack analysis	Malware attacks	TCG [165]
27.	Runtime restart of RT-IoT devices with tight timing constraints	Malware attacks	[192]
28.	Security awareness workshops and lectures for the employees	Social engineering attacks, phishing/spear-phishing attacks, download of infected/malicious apps	
Detective Measures			
29.	Runtime verification of firmware/code	Malicious code, corrupt software	

Continued on next page

Table 3.1 – *Continued from the previous page*

Ser	Security Measure	Impact / Threat Protected Against	References
30.	Log management	Facilitates detection of security breaches	
31.	Network security analytics	Detects security breaches, malfunctions and anomalies	Cisco [177], IBM-CIoT [196–198]
32.	Edge security analytics	Facilitates isolation of security events at the source and limit attack spectrum	IBM [53]
33.	Network level security measures to enforce cross-device security policies	Manipulation of actuator actions based on malicious/modified sensors data	[48]
Responsive Measures			
34.	Incident response plan	To streamline the response in case of a security incident and facilitate in recovering from the attack by adopting requisite corrective measures	
Corrective Measures			
35.	Self-recovery and diagnostics, and remote attestation	To recover from the security incident by reconfiguring the devices and removing all remnants of the attack	TCG [165]
36.	Secure reboot of RT-IoT devices	To recover from malware that resides in the RAM	[192]
Penetration Testing and Vulnerability Assessment			
37.	Penetration testing and vulnerability assessment	Detect/identify weaknesses at all layers of IoT architecture to facilitate respective countermeasures	[199, 200]

3.1.2.2 Detective Measures

a. **Firmware/Code Attestation.** Runtime verification of firmware/code installed on an IoT device is an important means of detecting the execution of malicious code installed remotely on a device.

b. **Auditing (Log management).** A record of all changes made to the system and devices be maintained to enable periodic audits to detect security breaches.

c. **Hardened Gateway Devices.** Security hardened gateway devices can be used to monitor sensors' data feed to determine the health of communication between devices and service-based applications.

d. **Security Analytics.** It helps in gaining visibility of the IoT ecosystem and ultimately controlling all the network components, including the hardware and software, to detect and rectify any malfunction or a threat [177]. IBM uses a Cognitive IoT (CIoT) Security Framework named Security-360. All the network components, including devices, users, applications, business processes and even workload, contribute to form a 360-degree view of the security posture. Based on data provided by the entire environment, the security mechanism assesses the changes in the security posture of the network and plans a defense. In this regard, various data mining and machine learning techniques can provide automated methods to track normal behavior and flag anomalies [196–198]. Moreover, Security Information and Event Management (SIEM) is also considered a vital component of a defense-in-depth approach to network security. It is

therefore concluded that intelligent threat analytics should be able to protect the IoT ecosystem against all sorts of threats based on known signatures, predictable malicious behavior [17], and correlation of security incidents/events.

A subset of overall system security analytics is "Edge Security Analytics." It is implemented by deploying security intelligence gateways. These intelligent devices provide swift responses to security incidents by faster detection of anomalies and re-mediation by isolation of events at the source and limiting attack spectrum. They also help in preserving the privacy of sensitive data by carrying out processing locally [53].

e. **Redefining Network-Level Security for IoT.** Today, IoT device manufacturers focus on novel functionality, ease in operation, and being the first one to launch a new product in the market. Hence, they do not give attention to device security [201]. This lack of manufacturers' attention to security coupled with constraint resources, IoT devices are not suitable for traditional host-based protection (anti-virus and security patches). Hence, researchers in [48] proposed a network-level security architecture to secure IoT devices. Their security architecture employs an IoTSec (security controller), μmboxes (gateways for IoT devices), and IoT end-nodes.

   The IoTSec controller centrally monitors the network and records security contexts and environmental variables for each end-device to form a global view of a set of possible states of the system. Based on the set of states, IoTSec decides or controls the flow of commands to the end devices. The proposed system is claimed to be equally useful to enforce cross-device security policies. E.g., in a smart home, if an attacker hacks into a fireplace and commands it to ignite the fire to cause an accident. To address this vulnerability, the IoTSec controller ensures that the fireplace is turned on only if the camera detects that someone is present in that room. The status of camera output, i.e., the presence of a person in the room, can be read from the current global state of the smart home maintained by the IoTSec controller itself. However, certain issues related to the centralization of the IoTSec controller and the limitation of using different μmboxes for every other kind of IoT device needs to be addressed.

3.1.2.3 Responsive Measures

An effective incident response plan begins even before any security incident occurs. In an IT environment, the response team is usually called the Computer Emergency Response Team (CERT). These teams comprise skilled cybersecurity professionals, auditors, legal experts, IT administrators, and other specialized members. The goal of CERT is to develop and physically practice a comprehensive response plan against any security breach so that all the stakeholders are clear about their responsibilities. An organized and well-planned incident response can make or break any business. Similarly, IoT can also design and employ a comprehensive response strategy. The response measures are also termed as after-incident reactive measures, which include:

- Action against compromised devices/parts of the system allowing the rest of the system to run its routine functionality.
- Revocation and blacklisting of malicious nodes.

- Initiation of anti-tamper mechanism, in which, as soon as the hardware of the node is interfered with, the node's memory containing firmware and the code should immediately be wiped off, and the node should only join the network after being activated by personalization instead of Over-The-Air-Activation (OTAA).
- Disconnect all the systems from the internet.
- Isolation of compromised sub-systems so that the healthy part of the network remains available.
- Recover important official and personal data from backup.

3.1.2.4 Corrective Measures

Once a compromised IoT device is detected and isolated from the network, the next step is node recovery, i.e., secure firmware/code update and reactivation of the device. There are two methods of node recovery. The first one is self-recovery, in which the device itself performs the integrity check of the code running on it and the last best configuration stored in read-only storage. If the validation fails, the device deletes the current code and reinstalls the last best configuration. The device then restarts and performs validation of all its modules. The second method is remote attestation; the device sends the integrity report to the controller/gateway device for remote validation [165]. The verifier then initiates a secure firmware update process if the validation fails.

3.1.2.5 Penetration Testing/Vulnerability Assessment

a. **Device Attestation.** Periodic device-side code analysis should be performed to check for the presence of any malicious code or modification in the original code. The successful code verification helps in shrinking the attack surface [15].
b. **Network Testing.** It mostly includes the use of penetration testing toolkits and other vulnerability assessment measures adopted by ethical hackers to secure the network. The most common tools are Metasploit, Wireshark, Nmap, Social Engineering Toolkit, Kali Linux, Nessus, etc. The penetration testing is done to find weaknesses in the target system. The testing can be performed on networks, websites, and servers. The weaknesses are then fixed by installing security patches, improving security configurations, making changes in the IDS and firewall rules, and security of open ports/interfaces.

3.2 Cost-Benefit Analysis for the Selection of Suitable Security Measure

After a deliberate discussion on the defense-in-depth approach for IoT comprising various preventive, detective, responsive, and corrective security measures, a question arises about the complexity and cost comparison of various security measures. In response to this question, authors in [202] illustrate that the security requirements of two distinct IoT systems and even the security features of two different technologies cannot be compared using a single measure. The

security measures are adopted, keeping in view the technical resources (computational power, battery life, memory, and available bandwidth) of end devices and the threat environment. However, at the same time, the traditional host-based security solutions such as anti-virus, frequent security updates/patches, secure execution environment, OS virtualization, etc., are difficult to be implemented on resource constraint IoT devices. Hence, a relative cost-benefit analysis of security measures providing the same level of security is essential to select the suitable technology. E.g., as discussed in Section-3.1.2.1, the allocation of a unique device identifier is essential to protect against ID spoofing and device replication attacks. However, just the allocation of an identifier is not enough, the safe storage of device ID and other associated cryptographic primitives such as private keys and symmetric keys require additional measures such as TPM-based keys [165, 166]. However, any additional security measure comes at the cost of additional overheads in the form of special hardware, high computation and energy costs, etc.

Similarly, blockchain, a distributed ledger technology (DLT), is recommended to replace centralized cloud platforms. Both blockchain and cloud store data for further processing. These technologies ensure authentication and integrity of data. However, there are few differences that play a key role in the selection of suitable technology for IoT. Cloud services are provided under the centralized control of one trusted entity. Hence, the cloud is vulnerable to the single point of failure concerning security and privacy issues [22] including data manipulation [203, 204], and the availability of cloud services. Concerning data manipulation, the cloud service provider has to be a trusted party as it has control over the data stored in the cloud and related services. Therefore, the cloud provider can manipulate user data [204]. Whereas, blockchain is orchestrated in a way that all the miner and full nodes in the blockchain network maintain the same copy of the blockchain state and the trust is distributed among all the network nodes. Hence, if one device's blockchain data is altered, the system will reject it, and the blockchain state will remain un-tampered. Correspondingly, a single point of failure also concerns the availability of the services when the cloud servers are down because of software bugs, cyber-attacks, power problems, cooling and other issues; users find it difficult to access the cloud services [203]. Whereas in the blockchain, data is replicated on many computers/nodes, and problems with few nodes do not disrupt the blockchain services. Cloud is also vulnerable to unauthorized data sharing. E.g., in the recent past, private data of 87 million users was provided by Facebook to a British political consulting firm “Cambridge Analytica” without users' permission [27, 205]. Such a data breach results in irreversible data security and privacy issues. Whereas, blockchain with its smart contract technology gives users the freedom to restrict access to their data to authorized entities only, without placing trust in any third party or a cloud service provider [206].

Currently, blockchain is considered to be computationally and energy-intensive in the backdrop of PoW-based consensus protocol used in Bitcoin Blockchain. However, considerable research is being done to design and develop IoT-specific blockchain technologies that infer low computational and energy costs [207–210], are scalable [204, 211] and also offer privacy-preserving computations on user data [212]. Hence, it is the cost-benefit analysis, the resourcefulness of end devices, and security requirements that holistically determine an

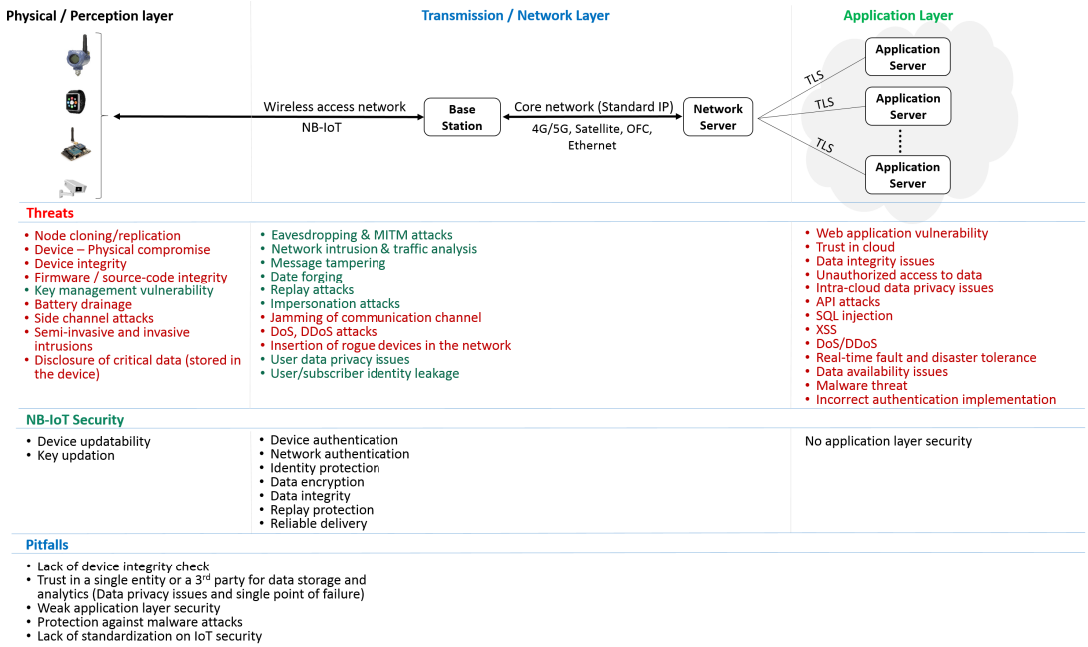

Figure 3.3: NB-IoT security in the IoT threat environment

appropriate security framework for an IoT system/use case.

3.3 Conclusions, Lessons Learnt and Pitfalls

To reach some logical conclusions/lessons and identify pitfalls concerning IoT security, we have projected a snapshot of the impact of security provided by one of the selected real-world IoT technologies on IoT threats (discussed in Chapter-2, Section-2.3 and Section-2.4,), in Figure-3.3. Although, there are many IoT communication technologies such as Zigbee, BLE, RFID, LTE-M, LoRaWAN, etc., that connect IoT devices with the gateways or base stations. However, LPWA (Low Power Wide Area) is considered to be a suitable technology for many IoT use cases due to its low power consumption, wide-coverage, long-range, low latency, reliability, low cost, better QoS, and considerable security [60, 202, 213]. Therefore, we have carried out a comparison of various LPWA technologies. As shown in Table-3.2, there are various options for LPWA technology in both licensed and unlicensed spectrum with varying security features. However, not all of the technologies can be discussed here in detail. Therefore, we have only mapped NB-IoT security features in Figure-3.3. Under the threats sub-section of the Figure-3.3, the points shown in red color are the threats/attacks that are not prevented by the NB-IoT security features. Whereas, the points shown in the green color are addressed by NB-IoT. It is evident that NB-IoT protects against the majority of the transmission/network layer attacks and only a few perception layer threats. Moreover, the application layer threats make it essential for the application developers to embed requisite security measures in the applications. It is evident from Figure-3.3 that the cryptographic security provided by the NB-IoT cannot protect against device capture and device tampering attacks. Moreover, there is also no mechanism to detect any forging or change in the

Table 3.2: Comparison of LPWA technologies

Feature	LTE-M	NB-IoT	LoRaWAN	Sigfox
Licensed spectrum	Yes	Yes	No	No
Device / subscriber authentication	UICC/eUICC	UICC/eUICC	Yes	Device only
Network authentication	Yes LTE-AKA	Yes LTE-AKA	Optional	No
ID protection	TMSI	TMSI	Partial	No
Data confidentiality	128-AES	128-AES	Yes (AppSKey)	No
Data integrity	Limited	DoNAS (Optional)	Yes	Yes
Control signal integrity	Yes	Yes	Yes	Not known
End-to-Middle security	No	No	Yes	No
Forward secrecy	No	No	No	No
Replay protection	Yes	Yes (Optional)	Yes	Yes
Reliable delivery	Yes	Yes	No	No
Device updatability	Yes	Yes	Limited	No
Keys updatability	Yes (Optional)	Yes (Optional)	Limited	No
Updation of long term keys	Yes (OTA)	Yes (OTA)	Limited	No
Requirement of certified equipment	Yes	Yes	Optional	Yes
IP network	Yes (Optional)	Yes (Optional)	No	No

device code, hardware configuration, and system files. Such protection is critical to detect remote code execution attacks that convert the devices into bots. The pitfalls observed in NB-IoT security are also shown in Figure-3.3.

As shown in Table-3.2, LTE-M, and NB-IoT operate in a licensed frequency band, whereas, LoRaWAN and Sigfox operate in an unlicensed spectrum [202]. Hence, it is imperative to discuss the impact of a licensed and an unlicensed frequency spectrum on the operational performance and security of an IoT system. The main advantage that NB-IoT has over LoRaWAN and Sigfox is that being in a licensed frequency band, NB-IoT is based on an international standard defined by 3GPP [61]. Therefore, NB-IoT can be termed as a mature technology with good QoS and less susceptibility to interference. Although the cost of a licensed frequency band is very high, i.e., more than USD 500 million per MHz, yet, the security and the performance benefits outweigh the cost effect. Being operating in a licensed spectrum, the end devices get access to the network after due authentication and authorization only. Therefore, it is difficult for an attacker to introduce a forged device in the network. Moreover, a regulating authority can control and manage a licensed spectrum with much ease as compared to an unlicensed one.

On the other hand, LoRaWAN is a non-standard proprietary technology with low QoS and no message delivery reliability. Being in an unlicensed frequency band, LoRaWAN and SigFox are at high risk of service degradation as the frequency band is shared with a lot of other radio devices. Moreover, the use of an unlicensed spectrum in most countries is regulated with some restrictions on the service providers concerning the maximum power of the transmitted signal and the duty cycle. However, still, it is difficult to control and regulate the unlicensed spectrum as at times, there can be a large number of ad-hoc networks operating in the said band. Correspondingly, the limitation on the duty cycle makes it difficult to support firmware updates over-the-air [214].

Whereas, IoT devices without any software updates or security patches are a security hazard. The brief discussion on the impact of real-world IoT technologies on the security threats and the previous discussion on IoT threats in Chapter-2, and IoT security framework guidelines discussed in Chapter-3, Section-3.1.2 has led us to draw certain lessons which further helped us to identify the pitfalls in the current IoT security environment.

Lessons Learnt and Pitfalls

a. IoT threats at various layers such as physical, MAC/Network, and application layer exploit different vulnerabilities and use different attack vectors to achieve malicious objectives. E.g., a device manufacturer leaves some open interfaces in the device hardware. These open interfaces can be exploited by the attacker to gain unauthorized access to the device and manipulate its operation [80]. Similarly, jamming of a communication channel targets the availability of the network or network services. Whereas, anti-jamming protection requires a different approach as compared to merely protecting against eavesdropping. Hence, attacks at various layers will have different impacts on the overall security of an IoT system and will require different security measures depending upon the IoT use case and threat environment.
b. According to the discussion in Chapter-2, Section-2.3, attacks at physical layer such as device capture, jamming of a wireless channel, hardware exploitation, node cloning, invasive intrusions, device configuration and firmware modification cannot be prevented only by cryptographic security provided by IoT communication protocols. Therefore, security has to be viewed as a whole and supplementary measures need to be taken at different layers based on the security requirements of IoT use cases. These additional security measures may infer some additional costs in the form of hardware, software, bandwidth, computation or storage.
c. The discussion in Chapter-2, Section-2.4, infers that depending upon the type and physical environment of IoT applications, end devices are vulnerable to physical attacks including device capture, tampering, invasive hardware attacks, side-channel attacks, reverse engineering, sensitive data leakage, and firmware/source code modification attacks [52].
d. DDoS attacks are mostly launched through compromised IoT devices [76]. Therefore, there is a requirement of an effective ingress as well as egress filtering, especially where IoT is connected to the internet.
e. Cyber attacks are considered as one of the biggest threats to IoT applications [215], and mostly the network and the application layers are the focus of the attackers [215].
f. No operation in an IoT system can be termed safe unless the integrity of the code installed on the IoT device and the integrity of the data being shared between devices is ensured [9].
g. Absence of anti-virus/malware detection mechanism in IoT is one of the causes of successful attacks on the integrity of the code/software of an IoT end device [8, 9].
h. Secure firmware update is one of the effective solutions against malware attacks in IoT. However, low downlink data rate, very short duty cycle, and lack of firmware integrity verification measures make it hard for an IoT technology to implement an effective firmware update mechanism [214].

i. Not all the IoT technologies meet the security needs of all possible IoT use cases. Instead, all technologies have adequate security for some specific applications. However, if the security provided is not enough for a particular use case, additional security measures can be taken but at the cost of some additional hardware, computation, energy or bandwidth cost, etc.

j. Security features of two different technologies cannot be compared using a single factor/measure.

k. The ideal LPWA technologies have some important security features as optional. These features are required to be enabled by the network operators. Hence, the user organizations/network operators need to have a clear understanding of what security features they require for which IoT use cases [202].

l. To effectively provide comprehensive security and privacy solution, it is necessary to analyze the IoT application and associated threats. Although similar, a smart home is different from a smart work environment. The solutions, especially the ones involving classic cryptography and physical layer security, must be tailored for the specific threats. The goal is to provide a cost-effective solution, while also taking into account the energy requirements (many devices can be battery-operated) [216].

m. Mostly, security is not the primary concern while designing IoT technologies or products [201]. Instead, the manufacturers focus more on the performance aspects such as low cost, low power consumption, more coverage, high data rate, ease of implementation, and service delivery.

n. Standard IT security protocols cannot be deployed on resource constraint IoT devices. However, selected standard security protocols can be optimized by removing various optional features.

o. Security is a holistic property. Hence, it should not be considered in isolation. E.g., LPWA technologies are developed with the primary objective of improving upon the performance and reliability concerning low power consumption, wide-coverage, long-range, low latency, reliable data transmission, low cost, and better QoS security [60, 203, 215]. Therefore, some compromises have to be made between the security and performance of the system. E.g., the use of lightweight cryptographic solutions to reduce computation overhead and power consumption. Similarly, efficient use of available bandwidth implies the use of security measures with less communication complexity.

p. Based on the discussion in Chapter-2 (Section-2.3, and Section-2.4) and Section-3.1 in this chapter, it is deduced that considerable research and development is being done in both academia and the corporate sector to mitigate threats to IoT. These threats fall in the domain of security triad, i.e., threats to confidentiality, integrity, and availability of data/information. As highlighted in Section-3.1.2, that security has to be viewed as a whole, and for a defense-in-depth approach against IoT threats, we need to deploy various preventive, detective, responsive, and corrective security measures. Hence, Table-3.1 shows that there are many Commercial off-the-Shelf (COTS) and academic security solutions available/proposed to provide preventive, detective, responsive and corrective measures. For instance, issues concerning device security such as device ID [164–166], tamper-proofing [164, 170],

registration and management [53], and secure boot [168] have been addressed by various tech giants including IBM, AT&T, TCG and Juniper Networks. Similarly, issues concerning data security and network access including authenticated encryption [178–180], privacy-preserving computation (homomorphic encryption) [172], secure cloud access [17], mutual device and gateway authentication [94], and secure network access control [94, 177] have also been meticulously tackled.

q. Whenever we talk about cryptographic security, key management is an associated challenge, and it is always considered to be an open research issue [216]. Similarly, after device, data, and network security, application layer security is also very essential as mostly the network and the application layers are the focus of the attackers [215].

r. The constrained resources in IoT devices and corresponding lack of strong security measures result in certain shortcomings that need to be addressed in the future. These include; absence of an international IoT standardization body that should govern minimal security standards as per the sensitivity and nature of IoT applications. Next is the lack of security mechanisms to ensure the integrity of IoT devices. Similarly, the protection of IoT devices against malware attacks and related secure firmware updates are still open challenges. Another critical aspect is that most of the data processing and analytics are performed under the centralized control of a third party/cloud provider that has to be the trusted one [106]. However, trust in a single entity results in various security and privacy issues. Finally, more work is required to be done in intra-cloud and distributed privacy-preserving data analytics. Similarly, the exploitation of zero-day vulnerabilities, especially at the application layer, is a persistent threat. Some of these vital open issues are discussed in detail in the next section.

3.4 Open Research Challenges

3.4.1 Baseline Security Standards

Because of the current lack of standardization on IoT products, diverse IoT applications and heterogeneity of IoT products, there are issues of security, interoperability, and compatibility. Most of the IoT products are being manufactured without any baseline security standard [37]. Whereas, keeping in view the existing threat spectrum, there is a requirement of various integrated security measures in IoT devices. These measures include requisite user authentication and authorization, encryption of data at rest and in transit, hardware security against tampering, and OS/application security. However, taking into account the constraint resources of many IoT devices such as sensors, Arm core or like microcontroller-based devices, CCTV cameras, baby monitors, home lighting systems, and the high computation and memory requirements for traditional cryptographic authentication and encryption solutions, there is a need to develop lightweight fully optimized cryptographic security protocols for IoT devices [217]. Application-specific functionality vis-a-vis low manufacturing cost and low energy consumption are also considered to be the limiting factors in developing a generalized solution for all the IoT products. Correspondingly, there is a

requirement of an international IoT standardization body to enforce minimum security standards in IoT products.

3.4.2 Privacy-Preserving Data Aggregation and Processing

Privacy is a critical security requirement for IoT users. Although considerable research has already been done concerning the user as well as data privacy, however, certain issues like privacy in data collection, data aggregation, data sharing, and data management warrant further attention [33]. E.g., data aggregation is done at the gateway devices to reduce the communication overhead between end devices and the cloud/servers. To preserve data security and privacy, the aggregation or processing is done over encrypted data by employing additive [218, 219] or multiplicative homomorphic encryption schemes. There are some full homomorphic encryption schemes as well [220, 221]; however, due to heavy computation load, it is difficult to use full homomorphic encryption schemes in IoT. Apart from data encryption, users' signatures aggregation is another approach to contain the communication overhead, given p signatures on p distinct messages from the same user. However, it is quite challenging to design a multi-key homomorphic signature to aggregate p signatures on p distinct messages generated by p users [113].

3.4.3 Software/Code Integrity

Numerous solutions to ensure the integrity of IoT end devices exist. However, the most dependable solutions are hardware-based that require execution of complete attestation process in a secure environment. But keeping in view the scale of deployment and low cost of IoT devices, manufacturing of secure hardware-based IoT products for usages besides critical infrastructure is not practical. Hence, there is a need to explore a secure software-based solution that can be easily deployed in resource constraint IoT devices with the flexibility of timely upgradation. Another foreseeable problem is that the next generation of IoT will consist of a large number of heterogeneous devices. Therefore, to detect and correct any malicious software modification efficiently, a swarm attestation mechanism for large dynamic and heterogeneous networks of embedded systems is still a challenging task [222].

3.4.4 Blockchain - An Instrument to Augment IoT Security

The success of Bitcoin brought the attention of the world to its underlying blockchain technology [24]. The blockchain is considered to be an unforgeable digital ledger that cannot be manipulated and changed. Although Blockchain was initially developed for financial technology (fintech), yet it is being adopted by many organizations to provide secure distributed services, such as smart city security [223], supply chain management [224], data sharing [225], data security [226], and decentralized and distributed web services [227]. However, blockchain's adaptation in the IoT ecosystem requires further evaluation. Figure-3.4 shows the inherent benefits of Bitcoin Blockchain in blue blocks, its limitations in pink blocks, and the blockchain features that IoT can leverage in green blocks. Whereas the open research issues are shown in yellow blocks.

BLOCKCHAIN for IoT	
Bitcoin Blockchain Pros & Cons	Features Suited for IoT & Reserch Challenges
Transaction integrity & authentication	Transaction integrity & authentication
Non-repudiation	Non-repudiation
No double spending/ avoids duplication	No replay
Prevents data forgery	Prevents data forgery
Decentralized control	Decentralized control
Pseudonymous identities	Identity management vis-a-vis user privacy
Neutralizes affects of ransomware & Cryptlocker	Needs to neutralize affects of ransomware and Cryptlocker
Ideal for untrusted environment	Needs untrusted environment
Transparency	Transparency of transactions
No encryption	Encryption (data security at rest and in transit)
Latency & low thorughput	Near real-time transaction confirmation
Energy intensive consesus protocol	IoT focused consensus with low overheads
Scalability issues	Scalable solution
Financial value based transaction validation	Needs IoT-centric transaction validation rules

Figure 3.4: Blockchain for the IoT

Although IoT can inherit some of the core benefits of blockchain such as decentralized and unforgeable digital ledger, transaction integrity and authentication, no double-spending, trustless operation, and by design protection against ransomware and cryptlocker type attacks. However, to make blockchain a reliable and secure platform for IoT, certain aspects need further research and evaluation. Such challenges include, ID management with due consideration for user privacy, user data privacy (both, on-chain and in transit), minimum latency in transaction confirmation for near-real-time IoT systems (smart vehicles, autonomous traffic management, smart grid, health monitoring), IoT focused transaction validation rules, IoT-centric consensus mechanism with low energy, low computation, and low communication overhead. The research on IoT-centric consensus mechanism must focus on consensus finality and fork prevention, which is a key to minimize latency in transaction confirmation and a critical requirement for real-time IoT systems.

3.4.5 Challenges to Fog Computing in IoT

One of the challenges in fog computing is to realize ID authentication while ensuring low latency of real-time services, the mobility of users, decentralized fog computing nodes and avoiding de-

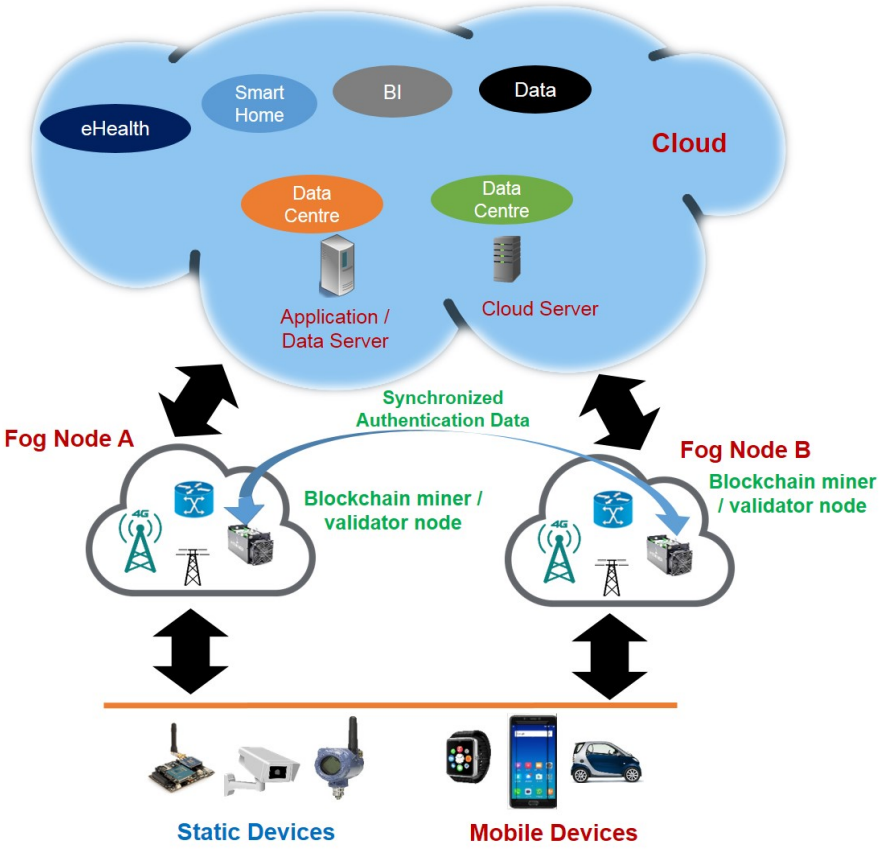

Figure 3.5: Blockchain-based ID authentication in fog computing

anonymization attacks [228]. Currently, there are many ID authentication schemes [229–231]. However, they do not cater to the mobility of the end devices. The probable solution to this challenge lies in the blockchain-based access control for fog computing. As shown in Figure-3.5, all the fog computing nodes can be the full blockchain nodes and can securely share and maintain the users' authentication and authorization information using group keys or attribute-based encryption [232, 233].

Another challenge is the consistency of the access control policy when multiple devices are used by users to access real-time services. The policy may involve device authentication and management mechanism for the users and key management mechanisms for the fog nodes. Although security is an essential part of any IoT system, however, the limited computational and power capability of IoT devices, makes it difficult to employ conventional cryptographic solutions. Hence, there is a requirement to design lightweight security protocols to support real-time services for fog-assisted IoT applications.

3.5 Summary

In this chapter, we presented a comprehensive set of security guidelines based on industry best practices that can help IoT standardization bodies to design minimal security standards based on types of IoT applications and devices. It was followed by an illustration of some important lessons learned from our discussion on threats to IoT in Chapter-2, and requisite defense-in-depth approach delineated in this chapter. Finally, some open research challenges related to IoT security were discussed. As for today, the inherent security provided by the IoT communication protocols does not protect against malware and node compromise attacks. Moreover, in the backdrop of a recent upsurge in the number of Ransomware attacks, the leading cause of their detrimental effects can be attributed to the centralized network architecture or cloud-supported operation and management of most of the IoT application scenarios. In these cloud-based IoT systems, network functionalities and security operations are controlled centrally. Such architectures are not only costly to set up but also present a single point of failure. Moreover, the IoT users, who are mostly the data owners, question the trustworthiness of the cloud service providers due to numerous data security and privacy issues. In addition, there is a lack of user-defined access control to data and transparency in the handling of data inside the cloud environment.

Hence, apart from other techniques, blockchain technology, with its inherent cryptographic security and unforgeable distributed architecture, is being evaluated and tested to address the IoT's security and privacy issues. It is believed that blockchain can resolve most of the IoT data integrity issues due to its ability to run distributed applications in the form of smart contracts and storing data on multiple nodes. Therefore, there is a dire need for an in-depth assessment of the blockchain technology to ascertain its suitability for augmenting the IoT security, identify the limitations, and recommend appropriate measures.

”The blockchain symbolizes a shift in power from the centers to the edges of the networks.”

- William Mougayar

4

Blockchain's Adoption in the IoT

The underlying technology of Bitcoin is blockchain, which was initially designed for financial value transfer only. Nonetheless, due to its decentralized architecture, fault tolerance, and data immutability, it has been weighed to resolve data integrity issues in IoT. However, presently, not much work has been done to assess blockchain's viability for IoT and the associated challenges. Hence, to arrive at intelligible conclusions, this chapter carries out a systematic study of the peculiarities of the IoT including its security and performance requirements. Subsequently, the gaps are identified by mapping the security and performance benefits inferred by the blockchain technologies and some of the blockchain-based IoT applications against the IoT requirements. This research also highlights some of the practical challenges to the integration of IoT with the blockchain. In the end, a way forward is proposed to resolve some of the significant challenges to the blockchain's adoption in IoT. This chapter has been published in the *Journal of Networks and Computer Applications* as a paper titled, “*Blockchain's Adoption in IoT: The Challenges and A Way Forward*,” Imran et al. [234]. In addition, an initial version was also presented at the 15^{th} International Conference on Security and Cryptography (SECRYPT) [235].

4.1 Introduction

Although blockchain was initially conceived as a financial TX protocol in the form of Bitcoin, but due to its cryptographic security benefits such as pseudonymous IDs, decentralization, fault tolerance, TX integrity and authentication, researchers and security analysts around the world are focusing on the blockchain to resolve security and privacy issues of IoT. However, default limitations of Bitcoin blockchain, such as scalability, latency in TX confirmation, large storage, intensive

computation and energy requirements, and privacy leakage, infer that blockchain technology has to be assessed deeply before it can be used securely and efficiently in an IoT environment.

4.1.1 Related Work

Till date, numerous surveys and some research on blockchain-based IoT technology [25,236–244] has been published but either these papers focus on general applications of the blockchain or discuss technical aspects concerning digital currencies. They do not give an insight into blockchain challenges related to IoT. For instance, [236] highlights various security, privacy, and performance issues such as DDoS attacks, 51% attack, data malleability, authentication, energy consumption, cryptographic, and usability problems. However, these issues have been discussed concerning cryptocurrencies such as Bitcoin, Ripple and Bitcoin exchanges. This work also identifies some of the research areas such as scalability, smart contracts, licensing, IoT, security, and privacy, which have been neglected in current research. However, for most of the part, [236] presents the methodology of its research and broadly highlights the current research topics. Moreover, if we look from the IoT perspective, [236] does not focus on this issue. Similarly, [237] carries out a detailed survey of blockchain technologies and their impact on society and the economy. It discusses the problems associated with Bitcoin blockchain. It also draws attention to the wide utilization of blockchain technologies, but IoT is just a point in the long list of potential use cases of the blockchain. Finally, it addresses the issues related to administration and policy guidelines.

In another work [238], authors give an overview of blockchain technology, discuss its variants such as Ethereum [245], Ripple [246], Gridcoin [247], etc., and present a gist of some non-financial applications of the blockchain. It also does not address issues concerning blockchain's adoption in IoT. Similarly, [239] presents a wholesome survey on the technical aspects of digital currencies. It discusses the Bitcoin characteristics and related concepts especially the consensus protocols in much detail but with respect to digital currencies. Although the papers mentioned above have covered various aspects of digital currencies and blockchain in detail, they are not focused on IoT. Moreover, authors in [240] present a lightweight architecture of a smart home. However, the paper just focuses on the limitations of Bitcoin blockchain and proposes a solution to avoid Bitcoin's issues of computation intensiveness, latency in TX confirmation and scalability. Correspondingly, the authors compare the security and performance efficiency of their solution with Bitcoin blockchain only.

Similarly, authors in [241] propose one of the use cases of the blockchain for IoT, i.e., configuring and managing IoT devices using blockchain smart contracts. By doing so, the authors aim to avoid the security and synchronization issues involved in a client-server model. Where, if a server gets malicious, then all the connected devices will be vulnerable to security issues. Therefore, taking advantage of blockchain's trust-free distributed architecture, the IoT devices are proposed to be configured and managed through Ethereum smart contracts [245]. Moreover, [242] carries out a literature review of blockchain applications beyond cryptocurrencies and their suitability to IoT. The review also aims at finding a solution to Bitcoin blockchain related vulnerabilities, such as integrity attacks, de-anonymization techniques, and adaptability of Bitcoin blockchain in IoT

concerning high TX input in IoT. Whereas, [25] gives an insight into the working of blockchain and smart contracts [245]. The authors prudently highlight the blockchain-IoT use cases such as a market place for sharing services and resources between IoT devices, P2P market for renewable energy and supply chain management (SCM). The paper also highlights some issues about the use of blockchain in IoT. These issues include low TX throughput, high latency in PoW-based blockchains, the privacy of users and TX contents, legal matters associated with smart contracts and the need for changes. Similarly, authors in [243] have also made a valuable contribution to the Bitcoin research. They have carried out an in-depth analysis of numerous Bitcoin properties, stability issues, and Bitcoin forks. The authors also gave an overview of alternatives to Bitcoin consensus and user anonymity/privacy techniques. Finally, [244] presents a systematic literature review of blockchain-based IoT solutions. The researchers also commented on the evaluation methods, evaluation metrics, and evaluation results presented in the reviewed studies.

Therefore, to cover the gaps in the literature concerning blockchain's adoption in IoT, there is a requirement of carrying out a comprehensive survey to find out how does existing blockchain technologies impact IoT. Similarly, how can IoT leverage blockchain to resolve its security issues, and what are the impediments in doing so. This chapter thus carries out a methodical review of the IoT threat environment, resultant IoT security and performance requirements and the impact of progression in blockchain technologies on IoT. The benefits afforded by the blockchain technologies and some of the blockchain-based IoT applications are pitched against IoT security and performance requirements to identify the voids. We also present a comparison of some of the notable blockchain consensus protocols based on certain security and efficiency factors to determine a suitable technology for the IoT.

Moreover, to discover some practical issues involved in the integration of IoT devices with the blockchain, we implemented an Ethereum blockchain-based IoT supported supply chain monitoring system in an experimental setting. We discovered that there are some challenges in securely sending sensor data from the IoT devices to the blockchain. It is also noticed that currently, there is no mechanism to perform a device integrity check to ascertain the validity of IoT devices. Whereas, it is an important security requirement since the IoT devices mostly operate in an unprotected environment and are vulnerable to physical compromise, which can result in malicious device operation. We also establish that there is a requirement for IoT-oriented TX validation rules and IoT-focused consensus protocols to meet the specific needs of the IoT environment. In the end, a way forward is recommended to address some of the significant blockchain issues. Hence, there are many factors that make our work distinguished from our predecessors.

4.1.2 Contributions of this Chapter

The primary objective of this research is to identify unscaled challenges that hamper the total adoption of blockchain in an IoT environment. The major contributions of this chapter are:

a. A comprehensive review and analysis of blockchain consensus protocols in the context of their implementation in IoT.

b. Detailed analysis of progression in blockchain technology and its impact on the IoT in view of security and performance requirements of IoT.
c. Identification of some unique and practical challenges to the blockchain's adoption in IoT.
d. Analysis of few existing blockchain applications and related voids.
e. A way forward to address some of the critical IoT related blockchain issues.

4.1.3 Organization

This chapter is organized as follows: Section-4.2 illustrates some critical security and performance requirements of IoT systems. In Section-4.3, some important blockchain concepts and the consensus protocols are explained. Progression in blockchain technology, and its impact on IoT is highlighted in Section-4.4. Whereas, Section-4.5 presents current challenges to the blockchain's adoption in IoT. The latest trends in blockchain-based IoT applications and related issues have been covered in Section-4.6. Gap analysis and a way forward to address some of the significant challenges are presented in Section-4.7 and Section-4.8 respectively. Finally, the chapter is concluded with a hint of future work in Section-4.9.

4.2 IoT Requirements

As discussed in Chapter-2, security flaws in IoT are resulting in attacks on device integrity, data integrity, secrecy and privacy, attacks on the availability of network and attacks on the availability and integrity of services, e.g., DoS and DDoS attacks [10]. The current security issues in IoT can be attributed to the poor security-aware design of devices, scarcity of memory, power, and computational resources, and trust in cloud-based applications. Based on the resource constraint peculiarities of IoT devices and IoT threat environment presented in Chapter-2, we have deduced some security and performance requirements for future IoT systems. Hierarchical models of these requirements are reflected in Figure-4.1 and Figure-4.2 respectively.

4.2.1 Security Requirements

The design and development of future IoT systems and devices is envisaged to be somewhat standardized as per the security requirements depicted in Figure-4.1. The essential security requirement of an IoT system is to be able to operate in a trustless environment. Moreover, most of the IoT applications rely on sensors' data. Hence, unforgeable storage and security against data manipulation and unauthorized sharing are also required. Furthermore, most of the IoT devices, such as smart city environmental sensors (temperature, humidity, gas, etc.), surveillance cameras, and ITS sensors being deployed in public places without much protection, are vulnerable to physical compromise [81, 83]. Resultantly, no operation in an IoT system can be termed safe unless the integrity of the code installed on the IoT device and the integrity of the data being shared between devices is ensured [9]. Therefore, device security is another important aspect that needs attention from the manufacturers and the security researchers. To protect the network against node compromise and malware attacks, the IoT systems need to authenticate devices before adding them to

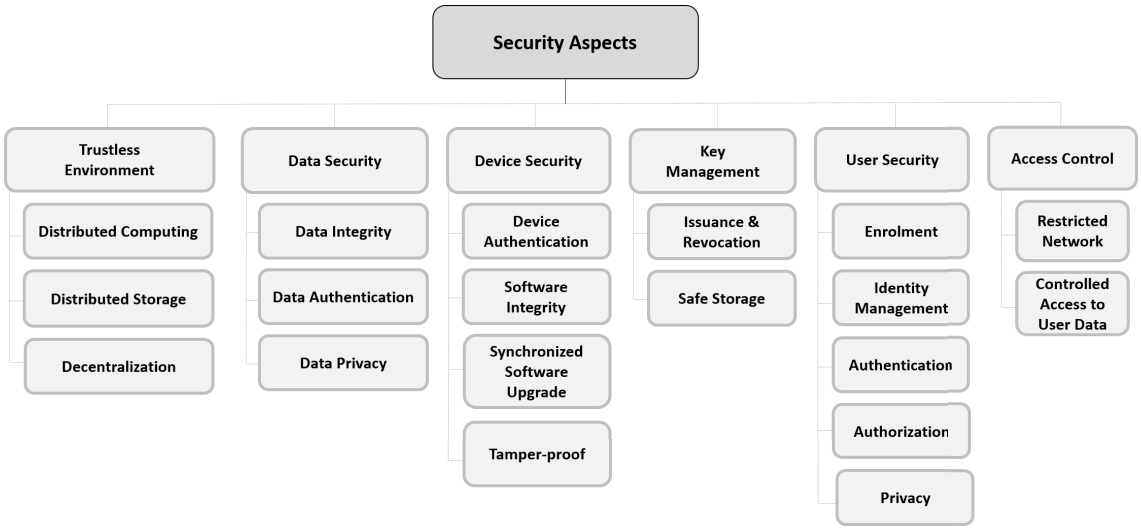

Figure 4.1: Security requirements for the IoT systems

the network. Similarly, there should be frequent checks to attest the integrity of the code installed on the devices. In case of any suspicion about the device software, the respective node should be revoked temporarily until the secure software update is performed.

IoT devices should also be tamper-resistant concerning both hardware and software modifications. Another vulnerable issue is that due to the scarcity of memory, power, and computation resources, redundant cryptographic security measures cannot be implemented in IoT devices [36]. However, still, IoT devices need some lightweight cryptographic security along with an efficient key management system, in which compromised keys should be revoked and updated as and when required. Another important requirement is user security, including enrolment, ID management, authentication, and authorization. In addition, a secure IoT system requires protection against unauthorized access to the network and user data.

4.2.2 Performance Requirements

Due to reliance on real-time data sharing by most of the IoT systems like VANETS, WSN, ICS, smart grids, smart homes, and SCM, the performance efficiency of the IoT system is as important as its security. Some of the performance requirements desired in IoT systems are shown in Figure-4.2. To protect future IoT systems against human errors, they need to be self-regulated and self-managed. An efficient IoT system must cater to the constraint resources of end devices, including low memory, low power consumption, and low computational ability. However, an increase in performance efficiency should not be on the pretext of compromising the security of the system. Moreover, a rise in the number of users/IoT devices in the future will result in the generation of more data. Therefore, the respective IoT system must be able to accommodate future network expansion and handle a large number of messages with high throughput.

The existing threat spectrum coerces the need for a sophisticated security mechanism for IoT.

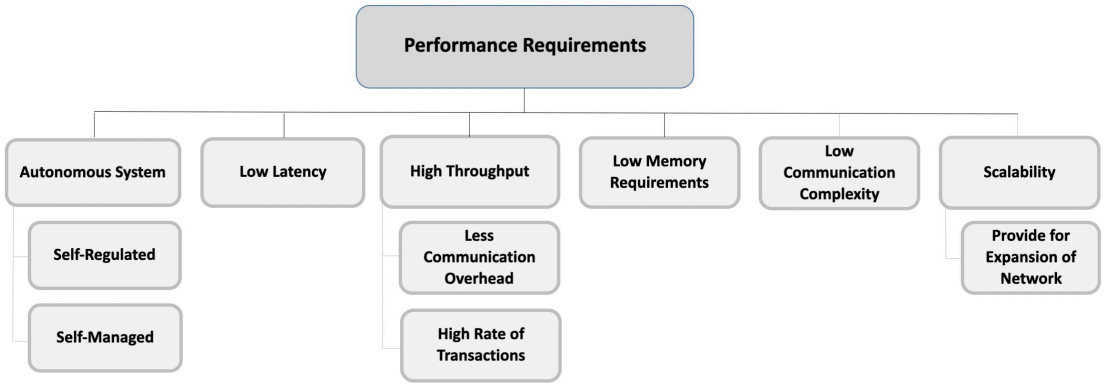

Figure 4.2: Performance requirements for the IoT systems

Many security researchers visualize blockchain as the silver bullet to augment IoT security. Therefore, before proceeding further, it is essential to get familiarized with the blockchain technology.

4.3 Blockchain: An Overview

The Bitcoin [24] has very innovatively transformed the method of financial value transfer without any trusted third party. The underlying technology of Bitcoin is blockchain. In simple terms, blockchain comprises a series of blocks in such a way that every new block is cryptographically connected to the previous block. In the case of Bitcoin, the blocks contain a record of financial TXs between Bitcoin users. Due to its inherent benefits, such as immutability, auditability, TX integrity and authentication, fault tolerance, and above all, trust-free operation, blockchain is being envisaged to play a vital role in the security of IoT ecosystem. Various benefits of Bitcoin blockchain and how they are achieved are enumerated in Table-4.1.

Blockchain avoids a single point of failure by employing DLT. None of the nodes in the blockchain has centralized control as the TXs are validated through network consensus. Moreover, the replication of blockchain state on all the nodes protects against data forgery, and node compromise attacks targeting data integrity. Blockchain provides a transparent public view of all the TXs, which presents an immutable log of events for future auditing. To ensure user anonymity, the real-world IDs are replaced by pseudonyms (cryptographic hash of the public key). The network consensus feature offers a trust-free environment without reliance on a single party. Blockchain employs various cryptographic algorithms to ensure TX authentication and integrity. The TXs in the blockchain are protected against double-spending based on the concept of Unspent Transaction Output (UTXO), where each input of a TX can spend the satoshis paid to the previous output, only once. Additionally, some important concepts concerning blockchain technology are illustrated in succeeding paras.

Table 4.1: Benefits of Bitcoin Blockchain

Ser	Benefit	Achieved by
1.	Avoids single point of failure	Distributed public ledger and decentralization
2.	No central authority or third party mediation	Validating the TXs with the consensus of network nodes
3.	No central database	Distributed public ledger
4.	Resilience to node compromise	Network consensus and state machine replication
5.	Auditable and immutable TXs	The recording of validated TXs in an unforgeable blockchain with a timestamp makes them always available for the audit. However, if an attacker acquires 51% or more hash power then he can change the history of the blockchain and double-spend the TXs
6.	Transparency	TXs are publicly announced to enable all nodes of the blockchain network to maintain the same copy of the order of TXs. Moreover, the TXs are published on the blockchain in cleartext
7.	Pseudo-anonymity	Hash of Public Keys
8.	Trust-free operation	Validation of each TX by network nodes
9.	TX authentication and non-repudiation	Signing of TXs by the user's private key using Elliptic Curve Digital Signatures Algorithm (ECDSA) [248]
10.	TX integrity	Taking SHA-256 hash of a TX
11.	Protection against double spending	Using UTXO, where each input of a TX can spend the satoshis paid to the previous output, only once

4.3.1 Key Concepts

1. **Transaction (TX).** A process that results in the change of state of the blockchain. Depending upon the blockchain platform, a TX ranges from the transfer of financial value to the execution of arbitrary code in the form of a smart contract [249]. Moreover, in the case of an IoT environment, a TX may be a means of sharing user or environment sensors' data.
2. **Block.** It is a set of TXs that happened in the recent past and have not been confirmed yet. The block also has a block header that contains the blockchain version number, hash of the previous block, a random nonce, timestamp, and Merkle Root Hash of all the TXs included in the block.
3. **Blockchain**. It is a distributed public/private ledger that keeps a record of all the TXs/blocks [24]. Vitalik Buterin in [250] gives another perspective that the essence of the blockchain is informational and processual, and does not relate directly to the monetary sphere.
4. **Mining.** It is the process of adding validated TXs to a block and then broadcasting that block

on the blockchain network, to be known by all the nodes. Miner nodes do the mining, and the selection of a node to mine a new block is done based on certain lottery schemes. In the case of Bitcoin, miners compete to solve a cryptographic hash puzzle, and whosoever finds the solution (also known as Proof-of-Work) first is eligible to mine the next block. When a block is mined and added to the blockchain, then the TXs in that block are confirmed [251]. Irrespective of the type of blockchain platform, usually some lottery scheme is required to randomly select a miner to propose or mine a new block.

5. **Simple/Normal Node.** There may be different types of nodes in a blockchain network depending upon their capabilities and resources, such as computational capability and memory size. A node may be a simple node, which can only send and receive a TX and does not store the complete copy of the blockchain. In the case of an IoT environment, a simple node can be an Arduino-based sensor node that can only send a sensor reading to the gateway device or receive some commands.
6. **Full Nodes.** These nodes maintain a complete copy of the blockchain, but they do not mine a block. However, full nodes validate TXs based upon the consensus rules of the respective blockchain and contribute to accepting or forking out a block [252]. A double-spending or a malicious TX may not even be routed or relayed by a full-node. This implies that full nodes are capable of TX and block propagation. Hence, full nodes are essential for the security of the blockchain. In an IoT environment, a Raspberry Pi (Rpi) with more computational and memory resources as compared to an Arduino, can be a full node [253]. We tested this functionality by running a Go Ethereum version geth-linux-arm7-1.8.3 on a Rpi-3 based sensor node.
7. **Miner/Validator Nodes.** These are the full nodes that have the additional capability to mine or validate a new block, thus extending the blockchain [252]. Moreover, mining nodes are selected as per specific criteria based upon the type of consensus protocol being used in the blockchain. E.g., In Bitcoin, the mining nodes have to solve a cryptographic puzzle, and the node that does it first is eligible to mine the next block. The miner node has to submit a PoW along with the mined block so that the rest of the nodes can validate that the puzzle has been correctly solved. If the rest of the network accepts the block, the miner node then earns a block reward and TX fee in the form of respective cryptocurrency. Whereas, in Proof-of-Stake (PoS) consensus protocol, miner nodes are selected based on their coinage, i.e., the number of coins they own and the time since they have those coins. However, in most of the Byzantine Fault Tolerance (BFT) based consensus protocols, the miner/validator is elected in a round-robin fashion to propose a new block. The rest of the member nodes of the quorum, vote on the validity of the block and its TXs. In most of the cases, the block is validated and included in the blockchain upon getting 2/3 majority votes in its favor.
8. **TX/Block Finality.** It is related to the final confirmation or approval of a particular TX or a block by the consensus protocol of respective blockchain. It is an important aspect as it infers delay in TX confirmation and ultimately affects the TX throughput of the blockchain. E.g., In Bitcoin, a TX gets one confirmation/approval after ten minutes, i.e., once the block

containing that TX is mined. However, to get final confirmation, the TX has to wait until an additional five blocks are mined and appended to the block containing that particular TX. Hence, it takes sixty minutes to finally declare a TX confirmed/approved in Bitcoin blockchain. Whereas, in other blockchains such as Hyperledger Fabric [175] and Tendermint [254], the TX gets instant confirmation.

9. **Permissioned vs. Permissionless Blockchains.** Before defining "Public" and "Private" blockchain types, it is imperative to highlight that a blockchain can be a permissioned or a permissionless blockchain based on the restrictions to process the TXs, i.e., creating new blocks of TXs. In a permissionless blockchain, any node can create new blocks of TXs, whereas, in a permisisoned blockchain, TX processing is performed by selected nodes only. As far as the terminology of a public and a private blockchain is concerned, it relates to the access to the blockchain data [255].
10. **Public Blockchain.** It may be a permissionless digital ledger that allows free and unconditional participation by any node [255]. Mining in public blockchains is mostly incentive-based so that miners are encouraged to mine a block. Hence, public ledgers bear more TX costs than private ledgers [250]. Whereas, the connectivity between nodes in public blockchain is less than in private blockchain; therefore, it takes a longer time to finalize the TXs [238]. Moreover, to achieve transparency in permissionless blockchains, all the TXs are visible to the public. Hence, issues related to user anonymity and data privacy emerge. Moreover, public blockchains have low TX throughput because of poor TX finality, especially in PoW-based blockchains [256]. Real-world examples of public blockchains are; Bitcoin [24], Ethereum [257], IOTA [258], Litecoin [259], Lisk [260], etc.
11. **Private Blockchain.** It can be a permissioned ledger, in which the number of the miner nodes is limited, and their IDs are known. Hence, TX processing is restricted to the selected/pre-defined miner or validator nodes only. Moreover, a user may have access only to those TXs that are directly related to him [255]. E.g., Hyperledger Fabric enables competing businesses and groups to maintain the privacy and confidentiality of their TXs, using "Private Channels" [261]. Private channels can be termed as restricted messaging paths that can be used to provide TX privacy and confidentiality for specific subsets of network members. All data, including TX, member, and channel information, are invisible and inaccessible to network members, not part of that particular channel. Hence, comparing to the public ledgers, there can be more privacy of user information in the private blockchains.

    Another difference between public and private blockchains is the extent to which they are centralized or ensure anonymity [238]. TX costs in private ledgers are also low amid less number of nodes [250]. Due to immediate TX finality, permissioned blockchains have high TX throughput [256]. Therefore, it can be attributed that private blockchains are faster than public blockchains. However, private blockchains with BFT-based consensus protocols suffer from poor scalability issues in terms of the number of validator nodes. Also, according to [248], the TX record in these types of blockchains can be tampered with due

to its partial centralization (known and less number of mining nodes). Concerning IoT systems, which are mostly private, a permissioned blockchain is the appropriate ledger technology. Some of the examples of real-world implementation of private ledgers include; Hyperledger Fabric [262], Multichain [263], Quorum [264], etc. The key differences between public and private blockchains are shown in Table-4.2.

Table 4.2: Public vs. Private blockchains

Public (may be Permissionless) Blockchain	Private (may be Permissioned) Blockchain
Permissionless participation	Permissioned participation
IDs of nodes are not known (Use of pseudonymous IDs)	IDs of nodes are known [255]
Unlimited number of nodes	A limited number of nodes
Less data privacy	Options available for data security
Poor consensus finality [265]	Instant consensus finality (Mostly in BFT-based blockchains) [265]
Low TX throughput [256]	High TX throughput [256]
Good scalability (concerning the number of miner nodes) [256]	Poor scalability (In BFT-based blockchains) [256]
Vulnerable to 51% attack (In case of PoW and PoS blockchains)	Vulnerable to node collusion (In BFT-based blockchains) [255]

12. **Hybrid Blockchain.** Being a balance between public and private blockchain, it is also called as "Partially Decentralized" or "Consortium Blockchain," [238]. E.g., In a consortium of ten industrial organizations, every organization maintains a mining/validating node in the blockchain network. In this case, a block may be valid only if it has been signed by minimum seven nodes. All the nodes may have open read access to the blockchain, or it can be restricted to specific nodes only [266]. However, there is a possibility of tampering with blockchain record due to reduced decentralization [248].
13. **Blockchain Forks.** Most of the public blockchains are prone to forks, i.e., if a miner node mines a block and the rest of the network rejects that block due to consensus rules violation, then the small chain extending from the rejected block onwards is forked, and the other longest chain extending from the correct block will be accepted as the valid chain. One of the main reason of forks in public blockchains is due to the consensus mechanism such as PoW, PoS, PoET, and PoA, in which there is no consensus finality once a block is mined. The consensus is reached subsequently once succeeding blocks keep on extending the chain leading from the older block. For example, to consider a TX as confirmed in Bitcoin Blockchain, it has to wait until six more blocks are appended to the block containing subject TX. Hence, temporary forks occur until the unconfirmed TXs are finally accepted or rejected by the main chain. Besides, the forks can be soft and hard depending upon acceptance and removal by the upgraded (following new consensus rules) and non-upgraded nodes (following old consensus rules) [251]. A soft fork is formed when a block violat-

ing new consensus rules is rejected by the upgraded nodes but accepted by non-upgraded nodes. In other words, a soft fork is backward compatible. Similarly, it is possible to keep the blockchain from permanently diverging if upgraded nodes control the majority of the hash rate [251]. An example of the soft fork is the adoption of "SegWit" to increase the TX speed in Bitcoin Blockchain.

In comparison, the hard fork is created intentionally once a system is upgraded, or an important change in consensus rules is deemed necessary. Hence, the latest version of consensus rules is not compatible with the older version. It means the hard fork is not backward compatible. Therefore, a block following the new consensus rules is accepted by upgraded nodes but rejected by the non-upgraded nodes. Correspondingly, when the mining software gets blockchain data from the non-upgraded nodes, it refuses to build on the same chain and accepts data only from the upgraded nodes. This creates permanently divergent chains, one for non-upgraded nodes and one for the upgraded nodes. A real-world example of the hard fork is the split between Bitcoin Cash (BCH) and Bitcoin due to a disagreement on the adoption of SegWit. Some other precedents of Bitcoin Blockchain hard forks include Bitcoin Gold (BTG), Bitcoin Diamond (BCD), Bitcoin Private (BTCP), and Bitcoin Interest (BCI) [267].

From IoT perspective, blockchain forks are not desired as they cause a delay in TX confirmation. E.g., In Bitcoin, due to the blockchain forks, a TX has to wait for six additional blocks to be mined over its respective block, to be considered confirmed. This wait time of six blocks infers a delay of sixty minutes in a TX confirmation. Whereas, in case of near-realtime IoT systems such as smart cars, ITS, drones, health monitors, a delay in TX confirmation can lead to a substantial physical damage and financial loss.

14. **Smart Contracts.** Exploiting the Bitcoin blockchain's ability to execute autonomous scripts, developers have created new versions of the blockchain that can perform arbitrary computations other than transferring coins. E.g., Ethereum blockchain [245] implements scripts called smart contracts [245] that can run any algorithm encoded in them as a part of the TX [268]. Being deployed on the blockchain, the smart contracts are also called as "Decentralized Applications (DApps)." Since smart contracts reside on the blockchain, they have a unique address. A smart contract can be triggered by addressing a TX to it under some rules that govern the contract. Smart contracts can be used in applications like auto-pay (shopping, parking, route management, tolls, fuel payment), digital rights management, financial services including loan, inheritances, escrow, cryptocurrency wallet controls, capital markets, mortgage, automatic payment of insurance claims [269], SCM, smart grid [25, 270], and etc.

    The key idea behind smart contracts is the development of autonomous objects or IoT devices that not only rent or sell their data but also maintain their operability by paying for the maintenance services. Such an autonomous system is likely to contribute to the development of an overall "Economy of Things" with the goal of providing efficient and consistent services without any intermediary.

15. **Consensus Protocol.** It is the mechanism or set of rules that enables all the full nodes to reach an agreement over the order of TXs. There are many types of consensus protocols being used in different blockchain applications. E.g., PoW, PoS, Practical Byzantine Fault Tolerance (PBFT), etc. Some of the notable consensus protocols are being discussed later in subsection 4.3.2.
16. **Consensus Finality.** It means, the convergence of the blockchain consensus process on a particular block/order of TXs. However, in reality, a consensus process may result into a permanent block or a stale block that may be forked out later. This aspect is further illustrated by Vitalik Buterin in [265], that the finality of a TX is always probabilistic. However, it may stand true for a PoW, PoS or PoET consensus protocols [271], but other consensus protocols may have different finality guarantees. Such as Casper [265] offers stronger finality guarantees as compared to PoW consensus and similarly, BFT-based consensus protocols provide immediate consensus finality [271,272], and the TXs once confirmed are not forked out later. From an IoT point of view, consensus finality is an essential requirement in most of the IoT systems as it also influences latency in TX confirmation.

4.3.2 Blockchain Consensus Protocols

1. **Proof-of-Work (PoW).** It is the computation of a cryptographic hash function with some degree of difficulty [24], i.e., selecting a nonce such that the computed cryptographic hash has a specific number of zeros in the start as defined by the level of difficulty. PoW forms the basis of consensus tactics in Bitcoin and other cryptocurrencies. When a miner node solves the PoW, it is eligible to mine a new block. Whereas, other full nodes in the network mutually confirm its correctness [248]. PoW protects against double-spending attacks. Since it is computationally intensive, it is challenging for a single attacker to solve the difficulty for all the modified blocks before the honest nodes in the network [24]. It is a common perception that if a malicious miner or a pool of miners gain 51% of the total network hash power, they can control the network [273]. However, authors in [274] prove that the malicious/dishonest miners resorting to selfish mining strategy can gain more revenue by only 25% of the total hashing power. Therefore, minimum 2/3 of the network nodes need to be honest to prevent selfish mining; a simple majority is not enough. Moreover, public networks with pseudonymous user IDs are prone to Sybil attack. Therefore, Satoshi Nakamoto conceived PoW-based consensus for Bitcoin blockchain to make Sybil attacks more expensive to be launched [207, 272].
2. **Proof-of-Stake (PoS).** It was conceived based on an idea described in [275] to improve upon PoW's high latency, high computation, and high energy costs. PoS implies that the people with high stakes are less likely to attack the respective network. Hence, an entity with the highest coinage, i.e., number of coins times the days, will be eligible to mine a new block. Moreover, the mining difficulty is inversely proportional to the coinage [239]. However, once the miners claim the reward, the coinage is reset so that other miners/stakeholders also get the chance to mine a block. Therefore, if an attacker wants to launch an attack

similar to 51% attack, he must own enough coins so that even when the coinage is reset, he can still gain more than half of the odds [239]. In addition, Nicolas Houy in [276] proves that PoS is vulnerable to a 51% attack, as the few rich stakeholders can collude to manipulate the state of the ledger. Nevertheless, the probability of a 51% attack in PoS is considered to be lower as compared to the PoW [277]. Moreover, the maximum TX rate a PoS protocol has achieved is a few hundred Transactions Per Second (TPS) as compared to Visa's peak capacity of 56000 TPS [278, 279]. Due to the lack of consensus finality, PoS-based consensus can also lead to blockchain forks [278]. A variation of PoS named "Delegated Proof-of-Stake" (DPoS) [280, 281] implemented in Bitshare, a digital currency, is considered to be more efficient than PoS in terms of TX confirmation time. Moreover, it can tolerate up to 50% malicious nodes [248, 273].

3. **Proof-of-Activity (PoA).** A combination of PoW and PoS [282], has been developed in the wake of an assumption based on an economic phenomenon called "Tragedy of the Commons." Which implies that over the period the block reward in PoW-based cryptocurrencies will subside, hence, the miner nodes will have less interest in ensuring the security of the network, thereby making it vulnerable to various attacks. Therefore, the proposed PoA protocol aims to increase the cost of an attack for a malicious user by forcing it to achieve eight times faster hash rate than the honest miners in the network. In addition, it reduces the computation complexity to 1/10th of the Bitcoin PoW, thus, minimizing the energy consumption as well. However, PoA also aims to secure only cryptocurrency applications.
4. **Proof-of-Authority.** Based on PoS, Proof-of-Authority is developed as an alternative to PoW in private blockchains. It has been implemented by Parity [283]. In this protocol, the authorities are pre-determined and each authority is assigned a fixed time slot within which it can generate blocks. Every authority is known based on its true ID; therefore, instead of monetary value at stake, Proof-of-Authority implies validator's ID at stake. Hence, any misconducting validator will be publicly known [284]. Proof-of-Authority makes a strong assumption that the authorities are trusted, and therefore, it is only suitable for permissioned ledgers. Ethereum test network Kovan [285] also employs the same.
5. **Proof-of-Elapsed-Time (PoET).** To address the problems of high power consumption and latency in the PoW-based consensus protocols, Intel developed a lottery-based consensus protocol named "PoET" for Sawtooth Lake, a blockchain-based distributed application platform. According to this protocol, the miner node, which presents the least waiting time, is selected to mine the next block. The PoET leader election protocol meets the criteria for a good lottery algorithm, i.e., fairness, investment and verification. It randomly distributes leader election across the entire population of the validators. PoET is secured by Trusted Execution Environment (TEE) through Intel's Software Guard Extension [286]. Except for leader election based on PoET for which specialized hardware is required, the rest of the protocol works like Bitcoin protocol. The trust is also placed in the hardware that generates the random wait time.
6. **The Proof of Burn (PoB)**. It implies that the users send coins to a verifiable but an unspend-

able address, thus burning the coins, to be eligible to mine a block [287]. The difference between PoW and PoB is that PoB has no energy costs, and its economic implications add towards the stability of the network. PoB has been adopted by a cryptocurrency named "Slimcoin" [288].

7. **BFT-based Consensus.** BFT is a family of state machine replication protocols [208, 209] that protects against arbitrary faults by replicating the services on multiple nodes. The safety and liveness property of BFT protocols can tolerate no more than $(n-1)/3$ faulty replicas over the lifetime of the system [289], where n is the total number of replicating nodes. However, in reality, any number of nodes can get malicious or show abnormal behavior [18]. In contrast to the PoW, BFT-based protocols require the IDs of the consensus nodes to be known, hence making it suitable for permissioned blockchains [272]. BFT-based state-machine replication protocols are considered to have poor scalability as they have never really been tested for the scalability beyond ten to twenty nodes [290]. Similarly, authors in [291] state that BFT-based protocols are not considered suitable for a network with more than a hundred nodes. The leading cause of the scalability issue seems to be the network communications which often involve $O(n)^2$ messages per consensus request [289]. Some of the variations of BFT-based protocols, which are currently being used in various blockchain platforms are mentioned in succeeding paras.
8. **Practical Byzantine Fault Tolerance (PBFT).** It is designated to be more efficient than a PoW concerning latency and energy costs, but it can only tolerate up to 33% malicious nodes. PBFT [289] is considered to be an expensive protocol concerning the number of messages required for consensus. The client's request is processed through five different stages, i.e., initially broadcast from client to all the replicas, then processed through pre-prepare, prepare, commit, and execution stage. Hence, in a network with four replicas, a single request requires thirty-two messages between client and replicas, i.e., four in stage-1, three in stage-2, nine in stage-3, twelve in stage-4 and four in stage-5 respectively.

   Moreover, in every stage of PBFT protocol, the decision is based upon no of certificates received for the previous stage. The number of certificates required to make a decision depends upon the estimated number of faulty nodes, e.g., to commit a message/request the replicas have to receive at least $2f$ prepared certificates for that request and to finally execute the request, the replicas need at least $2f + 1$ commit messages. Where f represents the number of faulty nodes. This means that the number of faulty nodes has to be pre-determined. PBFT protocol guarantees liveness based on weak timing assumptions. It operates in a primary-backup mechanism, and replicas move through a succession of configurations called views. Replicas initiate a change-view request, i.e., elect a new primary when a respective primary fails or does not respond in a set timeout period [289, 292]. Such weakly synchronous protocols are expected to degrade significantly when the underlying network behaves unpredictably. Therefore, the asymptotic communication complexity of PBFT in worst conditions can rise to ∞ [207]. Moreover, such a mechanism is expected to be vulnerable to less throughput in case of frequent network failures, and even DoS attacks,

where a persistent adversary causes network interruptions.

In a demonstration of such a DoS attack, authors in [207] implemented a malicious network scheduler to intercept and delay all view change messages of a PBFT protocol. They concluded that due to network interruptions and weak synchrony property of PBFT, the replicas remained stuck in view changes and never moved forward. They also deduced that such behavior is not restricted to PBFT. Instead, all protocols that rely on weak timing assumptions to tackle crashes can be affected by DoS attacks.

9. **Delegated Byzantine Fault Tolerance (DBFT).** DBFT has been implemented by NEO [210], an open-source blockchain project. NEO aims to realize the goal of the smart economy by employing the triad of digital assets, digital ID and smart contracts. DBFT consensus protocol is based on proxy voting and the NEO holders select the delegates/bookkeeper nodes that maintain the digital ledger. A speaker is selected amongst all the bookkeeping nodes, and together these nodes reach an agreement and generate new blocks. The protocol is tolerant to $f = (n-1)/3$ faults [210, 293], where, n is the total number of delegates/bookkeeper nodes and f is the number of faulty nodes in a consensus process. NEO provides efficiency by generating a block in 15-20 seconds with a throughput of 1000 TPS [210, 294, 295]. A new block is generated at the end of each round based on at least $n-f$ signatures by the bookkeeping nodes [293, 294]. During the consensus process, DBFT also depends upon weak-synchrony (weak timing assumption). Hence, a view change is requested by the nodes if the consensus does not take place in a particular view [293]. Therefore, DBFT is also vulnerable to DoS attacks based on network failures/interruptions. However, it provides consensus finality without any risk of blockchain forks [278]. As far as communication complexity is concerned, for one client and four validator nodes, DBFT consensus requires ten messages to process a TX.
10. **Honeybadger-BFT.** It is designed and optimized for a cryptocurrency scenario with restricted bandwidth but significant computing power [207]. It employs a BFT atomic broadcast protocol that provides optimal asymptotic communication complexity of $O(n)$ in the asynchronous network setting. Therefore, it does not rely on timing assumptions to make progress whenever messages are delivered regardless of actual clock time. As per experimental results, Honeybadger-BFT provides better throughput in terms of TPS, than PBFT. However, it has been tested for the tolerance of up to $f = n/4$ faulty nodes only. Moreover, the latency in TX confirmation also increases with the rise in the number of validating nodes. Hence, while expanding the network, there is a need to maintain a balance between the number of nodes, bandwidth utilization, and latency tolerance level of the users/applications.
11. **Tendermint.** Based on BFT, Tendermint employs a consensus protocol without mining. A block is initiated by a proposer, which is selected in a round-robin fashion from dedicated validators (with voting power equal to their bond deposit). TX validation is done based on majority voting, i.e., honest validators should have a majority vote of greater than or equal to $2/3$ of the total votes. There are three standard and two special steps in each validation

round. The consensus process in deciding the next block can be extended to many rounds (no bound on maximum rounds is given) depending upon certain conditions [254]. Some of these conditions include: the designated proposer is not online, block proposed by the designated proposer did not propagate in time, and even if there is a valid block, but greater than $2/3$ pre-votes or greater than $2/3$ pre-commits were not received by enough validators in time. This dependence on time can be exploited by any MITM adversary who can delay the messages from the proposers, thus forcing the protocol to go for so many rounds that the system experiences delays in computing new block heights. To curb false block propagation by the proposers, Tendermint employs a concept of punishment by confiscating the bond deposit of the faulty proposers. For one client and four validator nodes, the Tendermint consensus protocol needs to share twenty-one messages to process a single TX. It can also tolerate at the most less than $1/3$ faulty/malicious nodes.

12. **Algorand.** Algorand is a new cryptocurrency developed to overcome the issues of TX latency and blockchain forks in PoW, and PoS cryptocurrencies [296]. By using a Byzantine agreement protocol, a block is finalized at the end of the consensus process. Hence, TX confirmation time is brought down to an order of a minute. It also protects against Sybil attack by randomly selecting committee members for the consensus agreement based on their weight. Where weights are derived from the amount of money/cryptocurrency, one owns. Thus, as long as the honest users own more than some fraction (over 2/3) of the money, Algorand can avoid forks and double-spending. It addresses the issue of scalability concerning BFT protocols such as PBFT, which are considered to be communication-intensive and can scale merely to a dozen of nodes/servers. It achieves this by randomly selecting a small set of committee members for each step of the consensus protocol.

    Algorand avoids targeted attacks against the committee members by not using a fixed set of members. It selects the members in a private and non-interactive way. The users compute a Verifiable Random Function (VRF) on their public and private keys. The result of the function indicates to the users whether they are selected to participate in the consensus process or not. In this non-interactive way of selection, an adversary does not know exactly who the committee members are. Algorand, makes it further secure by selecting new committee members for each step of the consensus process. In this way, even if the attacker comes to know about a committee member once he starts participating in the consensus process, his attack efforts are futile, as that member will not participate in the next step. Algorand is claimed to be resilient to DoS attacks and it can continue to operate even in the absence of some of the users/nodes. As far as TX throughput is concerned, Alogrand commits a 2 MB block in 22 seconds and on the average (avg) commits about 750 MB of TXs in an hour, which is approximately 125 times the Bitcoin's throughput.

13. **IOTA.** It is a blockless distributed ledger developed to enable micropayments in IoT industry [297]. It employs tangle, a Directed Acyclic Graph (DAG), to store TXs instead of a blockchain. It is believed to be a successor of blockchain technology, as it addresses the issues of scalability and high TX fee. Latency in TX confirmation is reduced by making

consensus (TX validation) parallelized, and an integral part of the TX generation process. IOTA does not require a miner to mine a block of valid TXs; rather, every node approves/-validates randomly selected two previous TXs, before initiating its own TX. However, for the TX to be valid, the node must solve a PoW-based puzzle (similar to Bitcoin). IOTA is believed to be suitable for asynchronous networks, as all the nodes may not see the same set of TXs. Therefore, nodes do not have to achieve consensus on which valid TXs have to be included in the ledger. Instead, a specific node decides between two conflicting TXs by running a tip selection algorithm based on Markov Chain Monte Carlo (MCMC) method, which selects a TX based on acceptance probability. E.g., A user runs MCMC 100 times for a particular TX, and if that TX is accepted 51 times, then it means that the TX was approved with 51% confidence. For high-valued TXs, the threshold can be set as high as 99% acceptance probability. However, IOTA does not have consensus finality. Hence, it is prone to forks as well, which causes latency in TX confirmation. It is also not clear that after how many direct or indirect approvals a TX is safe to be declared confirmed?

For better performance efficiency, even if a node does not initiate any TX, it still has to work by relaying new TXs to other nodes, as each node maintains a record of TXs received from its neighboring nodes. As far as security is concerned, to protect against spamming attacks, every TX is weighted based on the amount of work done during PoW by the initiating node. Authors of IOTA claim that it protects against double-spending and quantum computing attacks by capping maximum own weight that can be assigned to a TX by the initiating node. Secure and authenticated data sharing between multiple nodes is also one of the core features of IOTA [280]. In spite of all these features, IOTA's security is questionable as security researchers from MIT Media Lab were able to break into IOTA's customized hash function "Curl" [298].

4.4 Progression of Blockchain Technology and its Impact on IoT

Bitcoin blockchain has revolutionized the distributed ledger technology with its significant cryptographic security and immutability. IoT can leverage the key benefits of the blockchain (as shown in Figure-4.3 to resolve its ever-growing security and privacy issues. E.g., The challenge of secure data sharing between heterogeneous IoT devices and guarantee of the trustworthiness of their data can be met by the common blockchain platform that provides immutability of data. Therefore, the blockchain with its decentralized architecture and unforgeability provides an ideal solution for IoT systems mostly operating in an untrusted environment.

IoT systems can also leverage blockchain technology as a secure, unforgeable, and auditable log of events and TXs, as per the type of the application. It can also be used to set policies, control and monitor access rights to user/sensor data and execute various actions autonomously based on pre-defined conditions using smart contracts [245]. However, in the past few years, due to IoT

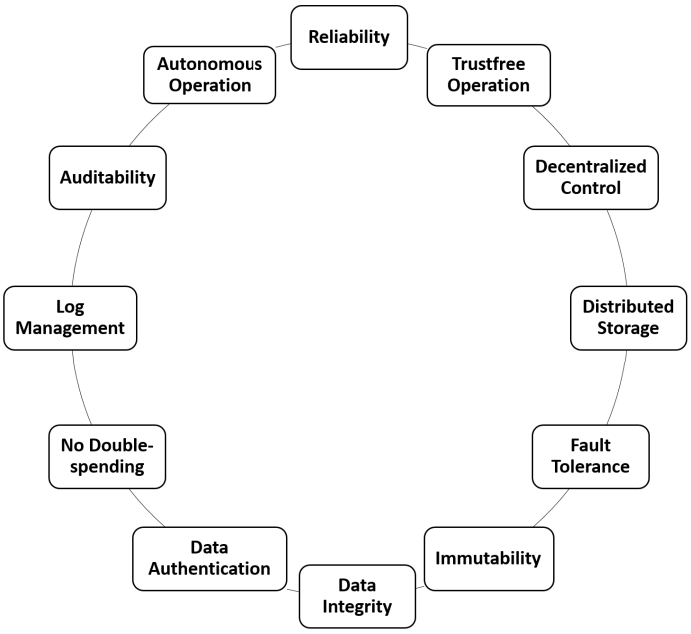

Figure 4.3: Benefits of the blockchain

devices' constraint resources, storage, processing and limited power, the use of cloud services has been on the rise to leverage cloud's computational and storage capabilities. But as discussed in chapter-2, Section-2.5, the cloud has its weaknesses. Therefore, it is imperative to highlight the major differences between the cloud and the blockchain.

As shown in Table-4.3, cloud services are provided under the centralized control of one trusted entity. Hence, the cloud is vulnerable to the single point of failure concerning security and privacy issues [22], including data manipulation [203,204], and the availability of cloud services. Regarding data manipulation, the cloud service provider has to be a trusted party as it has control over the data stored in the cloud and related services. Therefore, the cloud provider can manipulate user data [204]. Whereas, blockchain is orchestrated in a way that all the miner and full nodes in the blockchain network maintain the same copy of the blockchain state and the trust is distributed among all the network nodes. Hence, if one device's blockchain data is altered, the system will reject it, and the blockchain state will remain un-tampered. Correspondingly, a single point of failure also concerns the availability of the services when; the cloud servers are down because of software bugs, cyber-attacks, power problems, cooling and other issues; users find it difficult to access the cloud services [203]. Contrarily, in blockchain data is replicated on many computers/nodes and problems with few nodes do not disrupt the blockchain services. The blockchain is therefore good for data security and availability. However, blockchain has a limitation that with every passing day the size of the blockchain increases, e.g., the current (in Aug 2020) size of the Bitcoin blockchain is around 290 GB [299], and all the miner and full nodes are required to store the complete blockchain. In the case of IoT, this challenge is more pronounced, e.g., in a smart city IoT scenario, the sensor data coming from hundreds of thousands of IoT nodes will result in a rapid increase in the blockchain size, and the constraint resources of IoT devices concerning data

Table 4.3: Cloud vs. Blockchain

Cloud	Blockchain
Centralized architecture	Decentralized control
Trust is placed in the cloud provider	Trust is distributed in the network
Single point of failure (due to the possibility of data manipulation by the cloud provider)	Distributed architecture with blockchain state replicated on all the miner and full nodes
Vulnerable to data manipulation	Immutable
Prone to un-authorized data sharing	User-defined access control based on smart contracts
User data under control of cloud provider	Offers autonomous data sharing between users/devices through smart contracts
Users are never clear about intracloud TXs	Complete transparency by maintaining an unforgeable log of events and TXs
Not ideal for high data availability and low latency requirements of IoT	Provides edge storage and computing in terms of miner nodes that store the full copy of the blockchain
High network infrastructure and maintenance costs	Less expensive

storage make it difficult to handle large volumes of data. Hence, this limitation affects the utility of IoT devices as full or validating nodes in a blockchain network.

Cloud is also vulnerable to unauthorized data sharing. E.g., in the recent past, private data of 87 million users was provided by Facebook to a British political consulting firm "Cambridge Analytica" without users' permission [27, 205]. Such a data breach results in irreversible data security and privacy issues. Whereas, blockchain with its smart contract technology gives users the freedom to restrict access to their data to authorized entities only, without placing trust in any third party or a cloud service provider [300]. Here a question arises on how the data is stored and managed by the miners without compromising its confidentiality. In this regard, a blockchain technology "Hyperledger Fabric" follows a unique execute-order-validate architecture. To support this architecture, there are three types of nodes in the Hyperledger Fabric based on their roles, i.e., clients, peers, and orderers. The clients submit TXs in the form of chaincodes for execution. Whereas, peers execute TX proposals for the validation and endorsement as defined by the endorsement policy. An endorsement policy states that which, and how many peers are required to endorse the correct execution of a smart contract. Finally, the Ordering Service (ODS) nodes establish the total order of all the TXs and output a block containing previously unconfirmed TXs. ODS nodes are entirely unaware of the application state, and they neither execute the TXs nor participate in the TX validation process [301].

Hence, the execution of chaincodes by limited peers defined through endorsement policy restricts the exposure of TX payload and client ID to selected peers only. Moreover, to keep private data completely confidential from all the unauthorized users, the data values within the chain-

code/smart contract can be encrypted before sending TXs to the ODS and appending blocks to the ledger [302]. The encrypted data written to the ledger can be decrypted only by a user in possession of the corresponding decryption key. E.g., if a user wants that his financial or health-related data should not appear in plaintext, he can encrypt the data with the public key of the other user who is entitled to view that data. The authorized user can then decrypt the ciphertext using his private key. Data can also be encrypted/decrypted using Symmetric-key encryption algorithms such as AES. In addition to data encryption, role-based access control can also be built into the chaincode logic [303].

As far as issues concerning bandwidth are concerned, due to the state replication mechanism in the blockchain, every full/miner node must store a copy of complete blockchain [304]. Moreover, the decentralized nature of the consensus process infers that nodes in the blockchain network interact with other nodes to exchange information about the blockchain to participate in the consensus process, validate TXs, and create new blocks [305]. Therefore, Bitcoin-derived blockchain employs a gossip protocol so that all state modifications to the distributed ledger must be broadcast to all the nodes participating in the consensus process. The bitcoin blockchain is public and permissionless. Thus any node can join the network and participate in the consensus process. Hence, there is a great likelihood that the node with the smallest available bandwidth may become the network bottleneck. Moreover, as the size of the blockchain grows, the requirements for storage, bandwidth, and compute power required for participating in the consensus process increases. Hence at some point in time, it may not be feasible for all the nodes to process a block thus leading to the risk of centralization. In a traditional cloud-based system, such a situation can be addressed simply by adding more servers, using load balancing techniques or by increasing the bandwidth to handle the added TXs. Additionally, in the decentralized public blockchains, it is very difficult to control the public nodes [306]. However, in the case of private blockchains, which are mostly permissioned, only some selected nodes participate in the consensus process. Hence, it can be easily ensured that every node in the network has high computation power along with a high bandwidth internet connection. [306].

Moreover, due to the imminent increase in IoT devices connected to the internet, there would be an explosion in the volume of data produced by smart devices. Whereas, the existing cloud-based storage and computing solutions cannot handle such a large scale data due to the IoT requirements of high availability, real-time data delivery, scalability, security, resilience, and low latency [307]. Therefore, it is believed that blockchain due to its P2P distributed network architecture and state replication on all the nodes can augment the security and real-time data availability of fog nodes as an alternative to centralized cloud storage and computing [307]. However, still, blockchain's scalability issue concerning the ever-increasing size need to be resolved. Coming over to the progression in blockchain technology and the suitability of a blockchain platform for an IoT environment, we carried out a comparison [235] of some of the most prominent blockchain platforms, including Bitcoin [24], Ethereum [245], Hyperledger Fabric [175] and IOTA [297].

Although, IOTA is not as mature at the moment as compared to Ethereum and Hyperledger Fabric yet we have included it because its architecture is different than blockchain, it offers

Table 4.4: Comparison of blockchain platforms

Ser	Features	Bitcoin	Ethereum	Hyperledger Fabric	IOTA
1.	Fully developed	√	√	√	In Transition
2.	Miner participation	Public	Public, Private, Hybrid	Private	Public
3.	Trustless operation	√	√	Trusted validator nodes	√
4.	Multiple applications	Financial only	√	√	Currently, financial only
5.	Consensus	PoW	PoW, PoS ("Casper")	PBFT (for deterministic TXs), SIEVE, and KAFKA (Prototype)	Currently a coordinator approves the TXs through a Tip Selection Algorithm
6.	Consensus finality	X	X	√	X
7.	Blockchain forks	√	√	X	Not exactly forks, but a tangle that can be faded out later
8.	TX Fee	√	√	Optional	Feeless
9.	Run smart contracts	X	√	√	X (Not presently)
10.	TX integrity and authentication	√	√	√	√
11.	Data Confidentiality	X	X	√	X
12.	ID management	X	X	√	X
13.	Key management	X	X	√ (through CA)	X
14.	User authentication	Digital Signatures	Digital Signatures	Based on enrolment certificates	Digital Signatures
15.	Device authentication	X	X	X	X
16.	Vulnerability to attacks	51%, linking attacks	51%	$> 1/3$ faulty nodes	It is in Beta Testing
17.	TX throughput	7 TPS	8-9 TPS	$>$ 3500 TPS (depending upon the number of endorsers, orderers and committers)	Currently, the Coordinator being the bottleneck, the throughput varies between 7-12 TPS
18.	Latency in single confirmation of a TX	10 mins (60 mins for a confirmed TX)	15-20 secs	Less than Bitcoin, Ethereum & IOTA	Being in a transition phase, the TX confirmation time varies from minutes to hours
19.	Is it scalable?	X	X	X	Yes (Scalability concerning unapproved/pending TXs improves with the increase in the size of the network)
20.	References	[24, 251, 308]	[245, 257, 309, 310]	[175, 261, 262, 301, 311–313]	[258, 297]

fee-less TXs and is designed for M2M interactions. It also has the potential to resolve blockchain's scalability issues concerning low TX throughput with an increase in the number of network users. As shown in Table-4.4, the main security and performance considerations to ascertain the most suitable blockchain platform for an IoT system are as follows; the blockchain platform should provide a hybrid network concerning validating nodes' participation. As some IoT networks such as smart cities may have a large number of stakeholders willing to contribute to the security of the public blockchain network and on the other side, there may be a private network such as a smart home, where the owner would be validating the TXs via a couple of home miner/validator nodes. Currently, only Ethereum [245] provides such a hybrid technology, whereas, Bitcoin [24] and IOTA [297] support public participation. It is also imperative to mention that the level of decentralization in permissioned ledgers is affected by the lack of public access to the TX validation process, as limited miner/validator nodes currently do it. Whereas, the limited number of validating nodes is vulnerable to malicious collusion [255].

IoT systems are deployed for multiple applications, varying from smartwatches to ICS, and again its the Ethereum and Hyperledger Fabric that support multiple blockchain applications beyond fintech. Another important factor for an IoT system is low latency in TX confirmation which leads to the requirement of instant consensus agreement without blockchain forks. It is evident from Table-4.4 that Hyperledger Fabric based on PBFT/SIEVE consensus protocols [289] addresses this issue with greater reliability. Another essential aspect is that IoT systems especially the sensors operating in a smart city environment, would be generating millions of TXs per day. Therefore, an ideal IoT-oriented blockchain platform should not have a TX fee or gas requirement, e.g., Hyperledger Fabric has made TX fee optional.

The modern IoT systems not only require M2M micropayment methods but also need controlled access to user data, easy management of sensor policies and much more. Correspondingly, IOTA [297] is designed purely for M2M micro or even nano payments. However, currently, IOTA has not yet implemented smart contracts [245] which are essential for user-driven policy setting and access control rights. However, Ethereum and Hyperledger Fabric meet the requirement of smart contracts. Another important requirement for many IoT systems sharing private data of the users is the confidentiality of data. In this regard, only Hyperledger Fabric provides data confidentiality and also ensures the limited privacy of user data by allowing the creation of private channels [261] and encryption of data values in chaincodes [302]. Private channels are restricted messaging paths that provide TX privacy and confidentiality for specific subsets of network members. All data, including TX, members' profile and assets being traded on that channel, are invisible and inaccessible to network members not part of that particular channel. Moreover, the execution of users' TXs/chaincodes for validation is not performed by all peers. Instead, only one or more specific endorsing peers, as defined by the endorsement policy for a particular chaincode, execute the TX/chaincode for validation [301]. Hyperledger Fabric also supports ID management and TX authorization through public-key certificates (from a trusted Certificate Authority (CA)), which are vital requirements for IoT.

As far as performance is concerned, Hyperledger Fabric provides higher TX throughput than

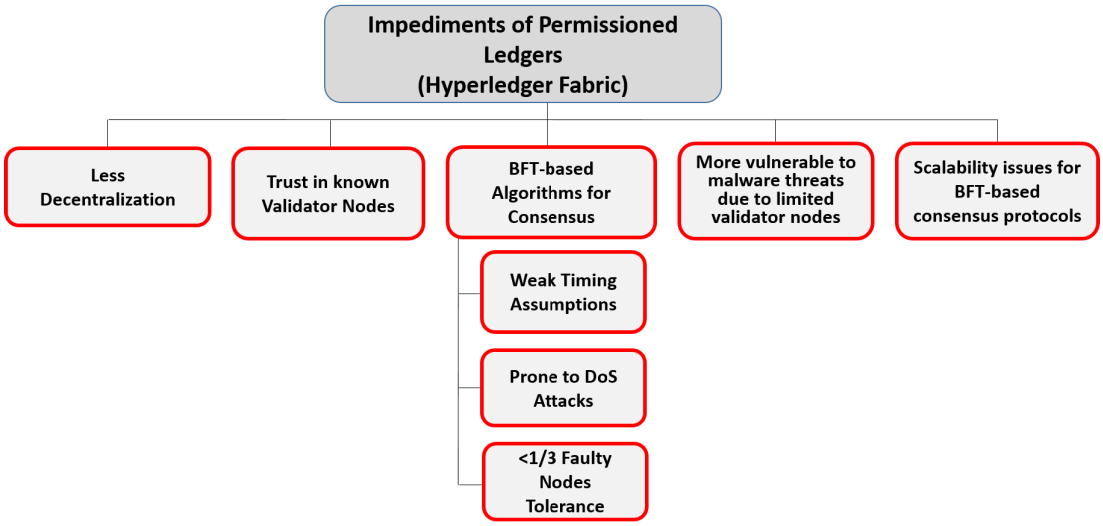

Figure 4.4: Impediments of permissioned blockchains

Bitcoin, Ethereum, and IOTA. Hyperledger Fabric consumes minimal energy and computation resources by using PBFT and SIEVE (a variation of PBFT) for validation of TXs, i.e., low energy and low computation cost [311]. Unlike Ethereum blockchain, it does not require any gas to process the TXs. Based on a BFT-based consensus protocol Hyperledger Fabric is a preferred technology for a permissioned ledger. However, as shown in Figure-4.4, there are some limitations in permissioned blockchains. Being partially-decentralized, trust is placed in some known miner/validator nodes. Hence, in the case of a successful malware attack such as Mirai [54], which can infect and compromise a large number of nodes for malicious purposes, the chances of TX and block validation process in permissioned ledger to be affected are more than in a permissionless or a public ledger with a huge number of miner nodes. Moreover, the user enrolment, authentication, and authorization based on public-key certificates are currently dependent on a trusted CA, which brings some degree of centralization. However, a Decentralized Key Management System (DKMS) for Hyperledger Fabric is under testing for release shortly [314]. Moreover, permissioned ledgers mostly use BFT-based consensus protocols. Whereas, such protocols are prone to DoS attacks and also cannot tolerate more than $1/3$ faulty nodes. BFT-based protocols such as PBFT are believed to have high communications complexity. Similarly, they perform poorly in adverse network conditions. Moreover, BFT-based consensus protocols have poor scalability, as the TX throughput decreases badly with an increase in the number of validator nodes, e.g., if the number of endorser nodes is increased from one to fourteen in Hyperledger Fabric, the TX throughput decreases to less than 1500 TPS [315]. However, still, BFT-based protocols provide low latency and much higher throughput than permissionless blockchains.

To conclude, Table-4.5 presents a recap of what all IoT security and performance requirements are met by the advanced blockchain technologies and what is still outstanding. Concerning IoT security requirements, many data and user security aspects have been addressed

by the blockchain platforms except privacy-preserving computation on sensitive user data, and most of the issues related to device security including device authentication, software integrity check, runtime/synchronized software update, detection of compromised device, IoT-centric consensus protocol and IoT-focused TX validation rules. As far as IoT performance requirements are concerned, some of these requirements are addressed by Hyperledger Fabric. However, low communication complexity and scalability should also be kept in view while designing an ideal IoT-oriented consensus protocol.

Table 4.5: IoT requirements vs. Progression in blockchain technologies

Ser	IoT Requirements	Blockchain Technology
IoT Security Requirements		
1.	Trust-free Operation	√ (All)
2.	Distributed Storage	√ (All)
3.	Decentralized Control	√ (All)
4.	Data Integrity	√ (All)
5.	Data Authentication	√ (All)
6.	Data Confidentiality/Privacy	√ (Hyperledger Fabric)
7.	Pseudonymous IDs	√ (All - based on Pseudonymous IDs)
8.	Privacy-preserving Computation	None
9.	User Enrolment	√ (Hyperledger Fabric)
10.	ID Management	√ (Hyperledger Fabric)
11.	User Authentication	√ (All)
12.	User Authorization	√ (Hyperledger Fabric)
13.	Key Management (Key Issuance & Revokation)	√ (Hyperledger Fabric)
14.	Restricted Network Access	√ (Ethereum & Hyperledger Fabric)
15.	Device Authentication	None
16.	Software Integrity Check	None
17.	Runtime/Synchronized Software Update	None
18.	Detection of Compromised Device	None
19.	IoT-centric Consensus Protocol	None
20.	IoT-focused TX Validation Rules	None

Continued on next page

Table 4.5 – *Continued from previous page*

Ser	IoT Requirements	Blockchain Technology
21.	Consensus Finality	√ (Hyperledger Fabric)
22.	No Forks	√ (Hyperledger Fabric)
IoT Performance Requirements		
1.	Autonomous System	√ (Ethereum & Hyperledger Fabric based on Smart Contracts)
2.	Low Latency in TX Confirmation	√ (Hyperledger Fabric)
3.	Low Communication Complexity	√ (Bitcoin, Ethereum, IOTA)
4.	Scalability	√ (IOTA - TX confirmation rate increases with the increase in network size)

Concerning suitability of an appropriate blockchain platform for IoT, as discussed in Section-4.3, BFT-based private/permissioned blockchains due to potentially improved performance and user security are suitable for IoT environment. Moreover, the IDs of the nodes that can control and update the shared state are known in permissioned blockchains [316]. Overall, private/permissioned blockchains offer more security and comparatively better performance than public/permissionless blockchains. The benefits of the permissioned ledger (Hyperledger Fabric) are shown in Figure-4.5. It is imperative to mention that unlike other permissioned and even permissionless blockchains such as Ethereum, Tendermint, Quorum and Chain, Hyperledger Fabric has a unique TX lifecycle of execute-order-validate. In which, although all peers validate the TXs to update the ledger, but not every peer executes the smart contract TXs. Hyperledger Fabric uses endorsement policies to define which peers need to execute which TXs. This means that a given chaincode can be kept private from peers that are not part of the endorsement policy [301]. However, it is recommended that any proposed solution should meet IoT security and performance requirements already illustrated in Section-4.2.1, 4.2.2, and the challenges to the blockchain's adoption in IoT (Section-4.5).

4.5 Challenges to the Blockchain's Adoption in IoT

To identify some real issues concerning blockchain's adoption in IoT, we implemented a test case scenario of an IoT-based supply chain monitoring system [235]. The customer orders frozen food products and also decides a temperature threshold that has to be maintained during the shipment by the seller. An alert is generated for the customer whenever the temperature threshold policy is violated during shipment. The test scenario and the challenges discovered while integrating IoT devices with the blockchain are explained in chronological order as labeled from 1 to 6 in Figure-4.6.

a. A Rpi-3 based sensor node (Scenario-1) can be connected directly to the blockchain as a full

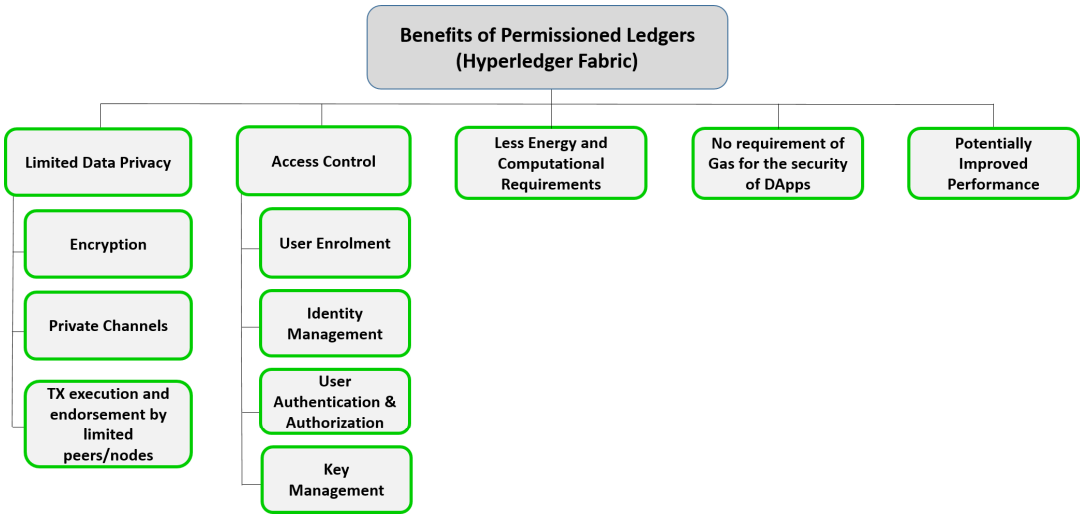

Figure 4.5: Benefits of permissioned blockchains

node [253] or a lite blockchain client [317]. A full node can validate other TXs, but a lite client can keep track of only its TXs.

b. The temperature sensor senses the environment, and its value is extracted via a web UI or a mobile app. The web UI or mobile app connected to the blockchain node pushes the sensor reading to the blockchain through a smart contract. Hence, a mobile or a web app is the interface between IoT devices and the blockchain.

c. In Scenario-2, an IoT device can be a resource-constraint Arduino device or any other embedded system capable of just sensing and transmitting the temperature sensor readings to a gateway device.

d. The Arduino-based sensor node communicates with the gateway device through slower and less secure wireless communication media such as 802.15.4 [318], 802.11 (WLAN standards) [319], LoRa [60], ZigBee [320], NB-IoT [60], and SigFox [321]. Resultantly, IoT systems are prone to data leakage and other privacy attacks [36]. Moreover, this arrangement also limits the blockchain-based device-to-device interaction, as now only the gateway device can access the blockchain or smart contracts.

e. Just like in Scenario-1, the gateway also connects to the Geth node through a web3 provider and pushes sensor data to the blockchain through a smart contract using a web UI or a mobile app.

f. However, there were certain challenges observed during this setup. Firstly, there is a question of how to ensure secure input of sensor data to the blockchain? Secondly, currently, none of the blockchain platforms implement IoT-focused TX validation rules and IoT-oriented consensus protocol. Lastly, an intermediary between the sensor node and the blockchain is the UI, which cannot leverage the cryptographic security provided by the blockchain. Instead, additional device, web, and application security measures have to be taken.

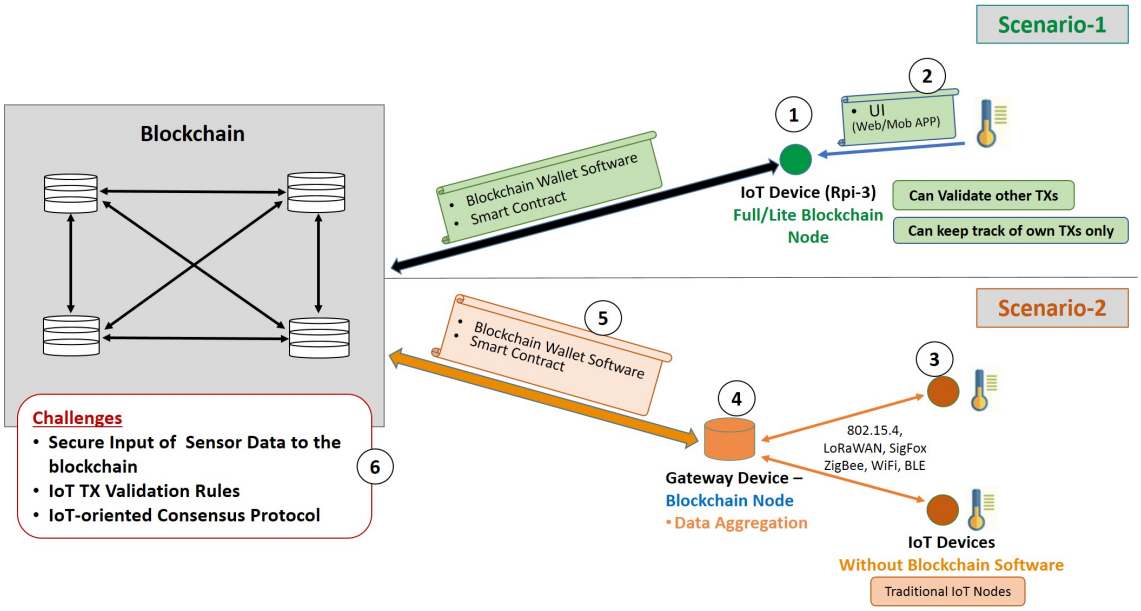

Figure 4.6: Challenges for a blockchain-based IoT system

As mentioned above, the primary challenge observed is the non-availability of an IoT centric consensus protocol. It also has some embedded issues such as TX/block validation rules, consensus finality, resistance to DoS attacks, low fault tolerance, and scalability concerning high TX volume, protection against Sybil attack, and communication complexity. Another related issue is the secure integration of IoT devices with the blockchain. These issues are being discussed in detail in succeeding paras.

4.5.1 Lack of IoT-Centric Consensus Protocol

Figure-4.7 presents a comprehensive comparison of some noteworthy blockchain consensus protocols. The points shown in green color are suitable for an IoT system whereas, points shown in red color are not appropriate for IoT. The current consensus protocols such as PoW [24], PoS [275], PoET [286], and IOTA [297] are designed for permissionless blockchains with a focus on financial value transfer. However, PoS and PoET can also be used in permissioned blockchains [271]. These consensus protocols share a common issue that the consensus process is probabilistic and does not end in a permanently committed block. Hence, they are prone to blockchain forks [278]. The lack of consensus finality results in delayed TX confirmation, which is not suitable for most of the real/near-real-time IoT systems requiring instant TX confirmation. Moreover, PoET requires special hardware and the enclave that allocates wait time has to be the trusted entity. PoET is also proved to be vulnerable to node compromise [322]. Also, as IOTA is currently in the open beta testing phase, it is assumed that some questions related to its security and performance efficiency will be answered in due course of time. E.g., Firstly, will it be an efficient IoT micro-payment system only? or It will also support smart contracts like in the Ethereum and Hyperledger Fabric blockchains. Secondly, does it provide confidentiality of data? and lastly, what is the faulty node tolerance level of IOTA?.

Consensus Protocol → / Features ↓	PoW	PoS	PoET	PBFT	DBFT	HoneyBadger-BFT	Tendermint	Algorand	IoTA
Utility	Fintech	Multiple Applications	Multiple Applications	Multiple Applications	Multiple Applications	Fintech	Multiple Applications	Fintech	Currently for Financial value transfer
Energy Costs	High	Low (as compared to PoW)	Low (as compared to PoW)	Low	Low	Low	Low	Low	Yes
Computation Costs	High	Low (as compared to PoW)	Low (as compared to PoW)	High communication complexity	Low	High (As compared to other BFT protocols)	Low	Low	Low
Consensus Finality	Probabilistic	Probabilistic	Probabilistic	Instant	Instant	Instant	Instant	Instant	Probabilistic
Prone to Forks	Yes	Yes	Yes	No	No	No	No	No	Yes
Latency in TX Confirmation	High	Low (as compared to PoW)	Low (as compared to PoW)	Low (Fast TX confirmation and high throughput)	Low (Fast TX confirmation and high throughput)	Low (Fast TX confirmation and high throughput)	Low (Fast TX confirmation and high throughput)	Low	Low Latency (No Fee, Parallelized Consensus)
Vulnerabilities	Prone to 51% attack	• Prone to 51% attack • Prone to malicious collusion of rich stakeholders	Node compromise	• Vulnerable to faulty nodes > (n-1)/3 (n = total nodes) • Vulnerable to DoS Attack • Poor Scalability concerning number of validating nodes	• Vulnerable to faulty nodes > (n-1)/3 (n = total nodes) • Vulnerable to DoS Attack • Poor Scalability concerning number of validating nodes	• Vulnerable to faulty nodes > (n-1)/3 (n = total nodes) • Poor Scalability concerning number of validating nodes	• Vulnerable to faulty nodes > (n-1)/3 (n = total nodes) • Vulnerable to DoS Attack • Poor Scalability concerning number of validating nodes	Vulnerable to dishonest nodes holding more than 2/3 of the total money	Still in beta testing.
Type of Blockchain	Permissionless	Permissionless and permissioned (both)	Permissionless and Permissioned (both)	Permissioned	Permissioned	Permissioned	Permissioned and permissionless (both)	Permissionless	Currently Permissionless
Requirement of Special Hardware	Not essential	No	Yes, Trusted Execution Environment e.g., Intel SGX	No	No	No	No	No	No
Additional Features						Avoids DoS attack (based on timing assumption) faced by other BFT-based consensus protocols	Punishment for dishonest validating nodes	More scalable than other Byzantine agreement protocols	• Avoids Quantum Computing Attacks • Suitable for Asynchronous Networks • Improved TX throughput with the increase in network size

Figure 4.7: Comparison of consensus protocols

On the other hand PBFT [289, 292], DBFT [210], HoneyBadger-BFT [207], and Tendermint [254] are BFT-based protocols. BFT is considered to be the desired protocol for permissioned blockchains, in which IDs of the nodes are required to be known [272], but it also has certain drawbacks. Except for HoneyBadger-BFT, the rest of the BFT-based protocols are prone to DoS attacks due to weak timing assumptions [207]. Correspondingly, the protocols based on timing assumptions are not suitable for unreliable networks, as liveness property of weakly synchronous protocols can fail when the weak timing assumptions are violated due to malicious network adversary capable of launching DoS attacks [207].

The weak synchrony also adversely affects the throughput of such systems [207]. Another major issue with BFT protocols is scalability concerning the number of validator nodes since they are not usually tested thoroughly beyond twenty nodes [272]. It can be attributed to the intensive network communications which often involve as many as $O(n^2)$ messages per block [289]. How-

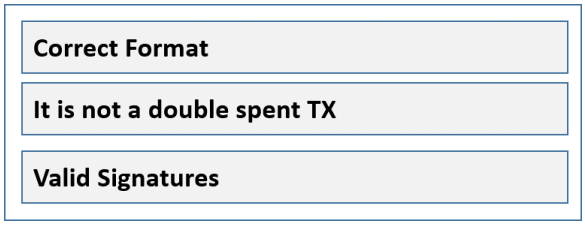

Figure 4.8: Bitcoin TX validation rules

ever, Algorand [296] claims to address the issue of scalability by randomly selecting a small set of committee members for each step of the consensus protocol. It uses VRFs for random selection of the users. It is also imperative to mention that in Algorand, the committee size is dynamic and is dependent upon two conditions, i.e., $\frac{1}{2}g + b \leq T_{step}.\tau_{step}$ and $g > T_{step}.\tau_{step}$, where, g and b are the numbers of honest and malicious committee members respectively, T is the number of votes needed to reach consensus and τ is the expected committee size. Concerning fault tolerance, BFT-based protocols are only capable of masking non-deterministic faults occurring on at the most $f = (n-1)/3$ replicas [289]. Where, f is the number of faulty nodes and n is the number of total nodes.

As far as TX throughput is confirmed, BFT-based protocols can sustain tens of thousands of TXs with practically network-speed latencies [323]. Another major difference between PoW and BFT-based protocols is the notion of availability, which is a critical requirement in real-time IoT systems, i.e., PoW being an incentive-based protocol does not guarantee that a pending TX will be included in the next block, as it is mostly at the discretion of the miners to select TXs based on their fee. Additionally, bandwidth efficiency and low communication complexity are also critical requirements, because most of the devices in an IoT system use wireless communication protocols and a typical smart city IoT network may comprise thousands of sensors. In this regard, PBFT is considered to be an expensive protocol concerning message complexity [324]. Therefore, any current or future blockchain-based solution must be able to sustain a large number of IoT devices and comply with the regulations of wireless communications as per respective country's law [325]. Moreover, despite reduced communication complexity and suitability for asynchronous networks, Honeybadger-BFT is not considered appropriate for IoT systems because of its cryptocurrency centric approach and low fault tolerance of $f = n/4$ faulty nodes only.

To conclude, certain aspects concerning the blockchain consensus protocols are required to be improved for its application in IoT. These aspects include IoT-centric TX/block validation rules, resistance to DoS attacks (exploiting timing assumptions), increased fault tolerance ($> 1/3$ faulty nodes), and low communication complexity.

4.5.2 TX Validation Rules

The TX validation process in Bitcoin (shown in Figure-4.8) validates a TX based on certain rules, including correct TX format, valid signatures and the fact that the TX has not been previously spent [245, 326]. On the other hand, (as shown in Figure-4.9) Ethereum blockchain validates

the format, signatures, nonce, gas, and account balance of the sender's account [245]. Whereas, in Hyperledger Fabric, TX validation is a three-step process [327]; When a client application submits the TX, each endorser executes the TX against the smart contract to check whether smart contract rules are being followed or not. A valid TX is sent back to the client with the endorsers' signatures. In the second step, the Ordering Service verifies TXs for inclusion in the ledger; this validation helps to control what goes in the ledger and ensures the ledger's consistency. Finally, all the committing peers check TX read-write set and endorsement policies before appending blocks to their copy of the blockchain.

However, there emerges a question that can the existing TX validation rules of blockchain platforms be applied to the IoT systems? That usually comprise heterogeneous devices, thus sending sensory values or data in distinct formats and different range of values. Moreover, IoT devices are also vulnerable to cyber-attacks. Hence, a targeted or even a generic malware attack can infect a lot of IoT devices. Subsequently, these devices may be turned into bots and used for further attacks. Therefore, TX validation rules of fintech-oriented Bitcoin and general-purpose Ethereum, and Hyperledger Fabric blockchain may not be suitable for IoT systems [235].

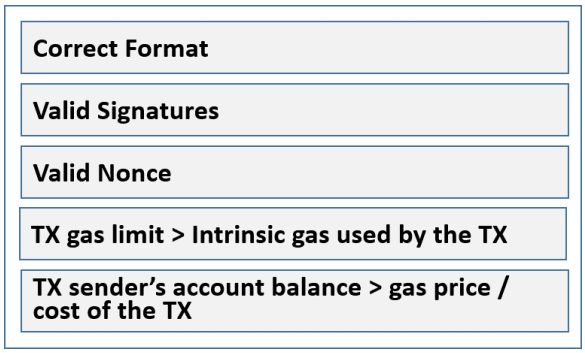

Figure 4.9: Ethereum TX validation rules

4.5.3 Scalability

It not only affects the blockchain size but also indirectly influences the consensus process. E.g., Rise in the number of users will also increase the number of TXs. Hence, if the consensus protocol has less throughput, then the latency in TX confirmation will be increased. Both the issues are being discussed separately in the succeeding paras.

a. **Storage Capacity.** A typical smart city IoT system with thousands of end nodes can generate a huge amount of data in no time. This data is then analyzed to extract information for various applications. Whereas the blockchain is not designed to store such a large amount of data. Moreover, the requirement of storing the complete blockchain by the full and miner nodes limits the integration of resource constraint IoT devices directly with the blockchain. Similarly, with the continuous increase in the size of the blockchain, the storage requirements also increase thus putting more limitations on resource constraint devices to act as full or

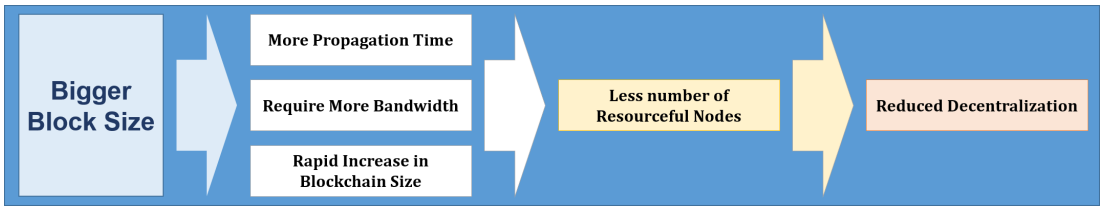

Figure 4.10: Disadvantages of bigger blocks

validator nodes. The increased blockchain size also takes longer to synchronize once new users/devices join the network. Therefore, it is a challenge to design a secure blockchain-based IoT solution which on one side, caters to the constraint resources of IoT devices and on the other inherit maximum benefits of the blockchain.

b. **Inherent Latency of Blockchain.** The real-time data sharing requirements of most of the IoT systems like WSN, ICS, smart vehicles, ITS, and smart grids, demand improvement in TX confirmation time, without compromising on the security and performance of the system. E.g., in a PoW-based blockchain, reducing the block generation time does lessen the TX confirmation time but to achieve the same level of security as with ten minutes block time; a TX has to wait for more confirmations because of less difficulty in mining a block. Moreover, with less block time, there would be more stale blocks thus wasting the computation and energy resources. Another factor associated with TX latency is the block size. There is a belief that by increasing the block size, say from 1 MB to 2 MB in Bitcoin blockchain, the throughput can be increased. But in reality, a bigger block will take longer to propagate in the network. Therefore, nodes with low bandwidth internet connections will suffer, and resourceful miners with more bandwidth will be at an advantage [328]. Also, an increased block size will result in the faster growth of blockchain size, which will affect the number of full nodes in the network, as more resources would be required to store the complete blockchain. Accordingly, Figure-4.10 shows the disadvantages of having bigger blocks.

   It is therefore concluded that to achieve security in a fully decentralized blockchain; there has to be a trade-off between performance efficiency and level of security, to prevent the system bending towards centralization. Correspondingly, a blockchain system with a certain degree of centralized control, may have some security and trust issues.

4.5.4 IoT Device Integration

In the test scenario shown in Figure-4.6, the IoT devices send sensor data to the blockchain through a web UI. The same can also be done by running a JavaScript code in the shell or a mobile App. Presently, smart contracts are only supported by some of the blockchain technologies including Ethereum and Hyperledger Fabric. Although Ethereum blockchain is currently the most acclaimed platform for DApps, yet, it has a major weakness. The smart contracts execute in Ethereum Virtual

Machine and do not communicate directly with the outside world. Therefore, a "web3" library is used as an interface.

In such a situation, the blockchain is only useful as a secure distributed database. However, before the data goes into the blockchain, its integrity is dependent on the security of the device, web UI or mobile app. Keeping in view the current IoT threat scenario, in which the IoT devices can easily be compromised, and malicious code can be executed remotely, the integrity of the IoT devices would always be doubtful. Moreover, the IoT data can also be corrupted due to some hardware/software failure or human error. Such an anomaly in sensor data cannot be detected unless the devices are tested for any hardware failure, software misconfiguration or other malicious modifications. At the moment, the only available solution is "Oraclize" [329]. It extracts data from various sources including web pages, WolframAlpha, IPFS, and any secure application running on Ledger Nano S. To prove the legitimacy of data, a "Proof of Authentication" is provided along with the requested data, i.e., the proof that data has not been changed and is in its original form as obtained from the source. However, it does not support IoT devices.

Another aspect of the IoT device integration with the blockchain is the lack of resources to be a full node or a miner node. Full and miner nodes are required to store the complete copy of the blockchain. Hence, a direct interaction of the IoT device with the blockchain through a blockchain client software will have additional memory and computational costs. Therefore, due diligence is required for enabling IoT devices to have a wide range of interactions with the blockchain [235].

4.5.5 Protection of IoT Devices against Malware/Remote Code Execution Attacks

This issue has two aspects, first is related to ransomware attacks, which has an insignificant effect in the case of a distributed ledger. Therefore, till the time a few nodes are unaffected, the network still has the accurate replica of the distributed ledger. However, the second aspect is that a node compromised due to malware can introduce fake/malicious data in the network. As in sensors-based IoT systems, each sensor has its unique data, which is event-based and is difficult to be linked to old TXs, unlike in Bitcoin. Therefore, it would be very challenging for other nodes to validate a particular sensor data/TX. Hence, there is a requirement of malware-detection/software-attestation in a blockchain-based IoT system that can detect malicious nodes. This aspect is further linked to the availability of a runtime software/firmware update mechanism. For example, an IoT system is hit by a wiper or a ransomware attack that wipes or encrypts all data including the OS/firmware files on end devices, thus making the devices non-functional. One of the recovery mechanism would be to initiate a firmware update procedure.

4.5.6 Secure and Synchronized Software Upgrade

Because of their critical functionalities, most of the IoT devices remain in continuous operation without any firmware or software updates. Hence, they are more vulnerable to cyber-attacks. Therefore, there is a need for a runtime firmware/software upgrading/updating mechanism. How-

ever, due to the decentralized architecture of the blockchain, currently, there is no mechanism to ensure synchronized software upgrades in the end devices.

4.5.7 Additional Issues

In addition to the challenges discussed above, some issues have been identified from the literature review.

a. **User Privacy and Data Security.** As discussed in Section-4.4, most of the blockchain platforms keep on-chain data in plain text, where, every TX can be checked, audited and traced back to the genesis block. Although this level of transparency does help to operate in a trustless environment yet at the same time, it affects users' privacy and data secrecy. Moreover, the pseudonymous IDs used by the Bitcoin blockchain do not guarantee total anonymity and thus are vulnerable to linking attacks [242]. Therefore, the applications running on public blockchains need additional cryptographic security, once dealing with sensitive or private user data along with some additional de-anonymization measures to de-link user ID.

   Concerning, user privacy/anonymity, currently, there are many variations of Bitcoin blockchain that claim to provide anonymous TXs. For instance, Monero [330] ensures user anonymity by using a ring signature scheme to make the TXs untraceable. Similarly, Zerocash [331] lets its users convert Bitcoins into Zerocoins (anonymous coins) and thus make obscure TXs. However, it is to be well thought out how to ensure user anonymity on a blockchain while guaranteeing user authentication and accountability. Whereas, to ensure data privacy on the blockchain, the data can be encrypted. Correspondingly, a blockchain-based smart contract system named “Hawk” [332] stores encrypted TXs on the blockchain. Similarly, for private blockchains, Hyperledger Fabric [261] addresses this issue by providing support for data encryption and sharing of data using private channels. In the same way, Quorum [264] makes use of cryptography and segmentation to ensure the security of sensitive data. However, still, there is a lack of blockchain-technologies that can ensure privacy-preserving computations and data analytics.

b. **Integration of IoT Communication Protocols.** There is an essential requirement for integration of IoT communication protocols such as BTLE, Bluetooth, 6LoWPAN, 802.15.4, Zigbee, LoRaWAN, etc., with blockchain for TX record, future verification, and possible monetization [333].

4.6 Latest Trends in Blockchain-based IoT Applications and Related Voids

Researchers and innovators around the world are developing and investigating ingenious ways to implement blockchain in the IoT environment. These use cases aim to take advantage of the inherent benefits of the blockchain such as decentralized control, immutability, cryptographic security,

Table 4.6: Blockchain applications

Application	Purpose	Blockchain Platform
ADEPT [333]	An autonomous, robust, scalable and secure framework for IoT devices	Ethereum
Security framework for smart cities [223]	Blockchain-based security framework for secure communication between smart city entities	Not mentioned
Secure firmware update [334]	Blockchain-based IoT device secure firmware update and integrity check	Proprietary blockchain with PoW consensus
Smart home architecture [240, 335]	Lightweight architecture of a blockchain-based smart home to control access to devices' data	Proprietary with no PoW
VANETS [336]	Decentralized and self-managed VANET	Ethereum
eBusiness model [337]	Blockchain-based autonomous sharing of data and properties	Ethereum
Transparency of SCM [338, 339]	Object tracking and record of ownership	IBM blockchain based on Hyperledger Fabric
BIFTS [340]	Traceability of perishable food	Not mentioned
Slock.it [341]	Managing things' services through smart contracts	Ethereum
Enigma [212]	Privacy preserving data computation	Proprietary

fault tolerance, data integrity and authentication, and capability to run smart contracts. Table-4.6 shows some of these applications, the purpose of their development and the respective blockchain platform. It is evident that not all applications use open-source blockchain platforms such as Ethereum and Hyperledger. Out of eight applications mentioned here, three applications use proprietary blockchains designed to their specific needs. Additionally, the main characteristics of these applications are shown in Table-4.7. We have tried to highlight the answers to certain questions concerning these applications, such as why is blockchain used? What blockchain platform is used? How is TX validation done? What conventional and blockchain issues are resolved? These applications are further discussed in detail to highlight their functionality, special features, voids and any innovation or cutting-edge feature that aims to resolve some of the challenges discussed in Section-4.5.

4.6.1 Consensus-based P2P Telemetry

Taking advantage of blockchain's ability to run smart contracts and network consensus on the validation of TXs, IBM disclosed a Proof-of-Concept (PoC) for a blockchain-based Autonomous Decentralized P2P Telemetry (ADEPT) system [333]. Based on Ethereum blockchain, ADEPT aims to implement a decentralized, autonomous, robust, scalable, and secure framework for the IoT that does not have a single point of failure. The proposed framework uses TeleHash protocol for P2P messaging, and BitTorrent for distributed file sharing. As shown in Table-4.7, the proposed system aims to resolve the issues in conventional IoT networks concerning trust in a centralized authority/entity, single point of failure, user, and data privacy issues, and errors induced through human interactions. It also endeavors to provide data privacy, user privacy, ID management, user-defined access control for data, and scalability. Certain voids regarding its employment in IoT are:

Voids. It is a PoC and requires further testing to ensure its reliability concerning security and performance efficiency.

4.6.2 Blockchain-based Security for Smart Cities

Key Features. In a conventional setting, due to the non-availability of a universal standard for smart devices, there are issues related to difficulty in sharing data received from heterogeneous devices and the integration of these devices to provide cross-functionality. Hence, Biswas and Muthukkumarasamy in [223] present an overview of a blockchain-based security framework for secure communication between smart city entities. Authors claim that the integration of the blockchain with devices in the smart city will provide a shared platform where all the devices would be able to communicate securely. Moreover, the use of blockchain will prevent data availability and data integrity attacks. It also provides an unforgeable log of TXs, that can be later used for audit purposes.

Voids. There is no qualitative or quantitative analysis of the proposed framework, including computation and transmission overheads. Moreover, it is not clear that what blockchain platform, consensus protocol, and TX/block validation techniques are implemented in the smart city application?

4.6.3 Secure Firmware Update

Key features. It is a blockchain-based IoT device firmware update scheme that lets the devices to securely check the firmware version and its integrity and then download the latest firmware [334]. This scheme vows to mitigate the effects of cyber-attacks targeting known firmware vulnerabilities. It also avoids network congestion issues that may arise due to simultaneous firmware update/download requests by a large number of IoT devices in an IoT network with thousands of devices deployed in a client-server model. It also aims to contain the size of the blockchain by avoiding the storage of updated firmware on the blockchain. Instead, it is done by implementing a P2P firmware sharing network using BitTorrent. However, it is not clear that what all messages

are logged on the blockchain for auditing. If all the messages related to firmware verification are logged, then the proposed scheme does not mention that how it will manage the ever-increasing size of the blockchain?

Voids. The proposed scheme has not been evaluated for communication complexity and energy consumption. Moreover, it is assumed that all the nodes work correctly, whereas, in reality, any number of nodes can be compromised. It is also not stated that how does the request node extracts and pushes the model number and firmware version to a blockchain TX? Another issue is that the nodes do some PoW to reach a consensus on the firmware verification. But it is not mentioned that what measures have been taken to avoid blockchain forks? What is the latency in TX confirmation? And how much time does a single firmware verification/update takes? It is also not mentioned that which nodes can perform PoW and which cannot? The distribution of normal nodes (resource constraint devices) and the miner nodes is also not given.

4.6.4 Blockchain-based Smart Home Architecture

Key features. Ali Dorri and Raja Jurdak in [240] and [335] propose a secure, private and lightweight architecture of a blockchain-based smart home application. The use of blockchain in a smart home differs from a conventional Bitcoin blockchain application in many ways. Unlike the Bitcoin blockchain, the local blockchain in the smart home is centrally managed by its owner. It has a policy header, which also acts as an access control list that allows the owner to control all the TXs happening in his home. For device-to-device communications, the miner issues a shared key between respective devices as per policy defined by the owner. The proposed scheme provides controlled access to IoT data. It also ensures data confidentiality, integrity, and availability along with protection against DDoS attacks. It aims to solve certain blockchain issues such as computational intensiveness, latency in TX confirmation, and energy consumption by forgoing the use of PoW in block mining. To reduce computational overhead and energy consumption, each block is mined without any PoW. Moreover, the latency in TX confirmation is reduced by considering a TX, true, whether it is mined in a block or not. Also, the proposed scheme utilizes cloud storage to ease up the memory requirements for smart home devices. However, some voids have been observed in this scheme.

Voids. Few aspects need further explanation with reasoning. Firstly, the hallmark of blockchain is the decentralized network, whereas, in this scheme, the Home-Miner, Cluster Heads (CH) and the cloud storage are providing a single point of failure at the respective layer. Secondly, most of the blockchain platforms validate TXs and blocks on a consensus decision by all the network nodes. However, in this case, it is at the discretion of the CH, whether to retain a block or reject it. Thirdly, it is only the Home-Miner that mines a block without any PoW, whereas, it is the difficulty level in PoW that protects the blockchain against double-spending and data forgery attacks. Lastly, in contrary to consensus-based TX validation in usual blockchain platforms, the Home-Miner checks all the incoming and outgoing TXs. Therefore, keeping in view the possibility of Byzantine General's Problem [342], if the Home Miner gets corrupted or malicious, the integrity of the blockchain TXs cannot be guaranteed. The nodes use The Onion Router

(TOR) for connection to the overlying network to achieve more anonymity/privacy at IP Layer. The overlay network maintains CHs that store public keys of the requesters, requestees, and the list of TXs forwarded to other CHs. It is up to the CH, whether to keep a new block or not, whereas, in Bitcoin blockchain, it is a consensus decision.

4.6.5 Blockchain-based Self-Managed VANETS

Key features. The conventional VANETS have a centralized managing authority. This arrangement has many drawbacks from a single point of failure to present a lucrative target to the attacker. Moreover, due to centralized management, it has less user privacy. To avoid such issues, Leiding. et al. [336] propose an Ethereum blockchain based decentralized, self-managing VANET with a challenge-response based authentication. The complete VANET is regulated by Ethereum-based applications (smart contracts), which are used to enforce certain rules or provide different services. Each node/user is registered and identified by its Ethereum address, i.e., a hash of its public key. To access services provided by Ethereum-based applications, every node has to pay in the form of Ethers. Thereby, the users fund the network infrastructure. The payments made by the users serve as the incentive for the vendors providing Ethereum-based applications and associated services. In a real-world scenario, the Ethereum account of a user can be used to make automated payments of car insurance, registration, additional services like real-time traffic updates, and payment of traffic violation fines.

Voids. The proposed scheme does not explain how PoW will be performed by the miner nodes to mine a block in the blockchain? There is no discussion about what information about each node will be published on the blockchain? Certain other aspects also need due consideration, like, who will mine the block? How will Vehicle-to-Vehicle (V2V) communication take place in the blockchain-based VANET? and what is the latency in communication? Latency is a critical issue as it is an inherent weakness in most of the blockchain protocols. Whereas, most of the time, the nodes/cars connected to VANET need real-time information about traffic and road conditions.

4.6.6 IoT eBusiness Model

Key Features. In yet another venture [337], Yu Zhang and Jiangtao Wen propose a blockchain-based decentralized electronic business model for the IoT. The proposed model aims to share paid data and smart properties like a car, parking space, house, fuel, e-shopping, commodities, and services, by applying the concept of Decentralized Autonomous Corporations (DAC). The key idea here is that DAC is automated without any intervention by humans and make use of smart contracts for decision making. It enables rapid information exchange among all stakeholders, i.e., sensors, computers, humans, DACs, buyers, sellers, etc. Moreover, each device in IoT can serve as a service provider. The proposed model has been designed and developed by modifying and optimizing basic elements and operating modes of the conventional e-commerce system. The efficiency is increased by removing the third party, working in a low trust environment, and reducing latency. The DAC model can be deployed for each smart device/sensor to trade it's paid data for some

service like power, additional module and software up-gradation, etc. The authors implemented the test case of the proposed model using Ethereum blockchain and aim to further develop an automated transfer of ownership service for smart properties.

Voids. Although authors gave a detailed overview and insight into their proposed e-business model for IoT, yet it was not clear how the constraint resources of IoT devices like less computational power, small memory, and low energy consumption will be met? The proposed solution mostly focused on the working of the e-business model, so there is a lack of discussion on technical aspects. Hence, some issues that need more deliberation include; which are the miner nodes? What data from the blockchain will be stored on the IoT devices? What security measures have been taken to protect against device compromise attacks, and how devices are integrated with the blockchain?

4.6.7 Transparency of Supply Chain Management (SCM)

Key Features. The blockchain is an ideal platform to ensure product authenticity and transparency during its complete supply chain cycle. It will help in tracking the origin and the transformations undergone by a product in the supply chain by maintaining a formal registry. The digital ledger can be connected to a supply chain sensor network connecting cargo trucks, storage coolers, etc., to keep track of product location and its environment parameters like temperature and humidity [224].

In a similar endeavor, Everledger, a UK-based global startup, has launched a Global Digital Ledger based on IBM Bluemix [343] to digitally certify diamonds to assist in the prevention of frauds. The digital ledger stores complete data about diamonds including their ownership and TX history. The immutable ledger aims to support owners, insurance companies, banks, and law enforcement agencies to verify the complete life cycle of a diamond since its discovery in the mine until its sale in the market and subsequent ownership. To date, Everledger has certified more than one million diamonds. The company has not disclosed any technical details about Everledger. However, it claims to use a hybrid blockchain model to take advantage of permissioned controls as in the private blockchains [344]. The company is also aiming to apply the same solution for the security of fine arts, vintage cars and wine [338, 339].

Voids. Irrespective of the practical manifestation of the blockchain in SCM, there is an inherent issue of interfacing blockchain and different types of physical devices. Moreover, there are questions related to the status update regarding the location and condition of a product in transit to a customer. Which is currently done manually by a human or by a sensing device. Now in a distributed environment, no other sensor node knows about the exact condition of this product once it has reached the warehouse, except the node reporting upon it. Therefore, there has to be some element of trust in that sensor node, such that its input data is accepted in the blockchain. Hence, if all the nodes are trusted, then there is no need of a blockchain. Moreover, if there is no trust, then the complete supply chain is compromised, and any malicious node can inject false data [344].

Table 4.7: Main characteristics of blockchain-based IoT applications

Characteristic	Applications									
	ADEPT	**Smart Cities**	**Firmware Update**	**Smart Home**	**VANETS**	**IoT eBusiness**	**SCM**	**BIFTS**	**Slock.it**	**Enigma**
Why is blockchain used?	Take advantage of smart contracts and network consensus	For improved reliability and better fault tolerance	To ensure data integrity, data authentication, and non-repudiation during firmware verification	For distributed trust and a common platform for controlled access to IoT devices and their data	For decentralized control	To achieve a transparent self-managed and self-regulating system based-on smart contracts	Due to its unforgeability	For transparency	Due to its decentralized control and ability to execute smart contracts	For decentralized control
What blockchain platform is used?	Ethereum	Not mentioned	Proprietary blockchain platform with PoW consensus	Proprietary	Ethereum	Ethereum	IBM blockchain platform	Not mentioned	Ethereum	Proprietary
How is TX validation done?	As in Ethereum	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned
What conventional issues are resolved?	Trust in a centralized authority/entity, single point of failure, user and data privacy issues, errors induced through human interactions.	Difficulty in sharing data received from heterogeneous devices	Mitigating the effects of cyber-attacks, avoids network congestion issues	It provides controlled access to IoT data and also ensures data confidentiality, integrity, and availability along with protection against DDoS attacks	Centralized control and privacy issues	Centralized control and issues in transparent data sharing/services	Issues of centralized database	Food traceability in supply chain	Centralized control and human intervention for access control and manual handing over of the products	Data privacy during sharing and distributed computation
What blockchain issues are resolved?	Data privacy, user privacy, ID management, user-defined access control for data, and scalability	None	Scalability (related to blockchain size)	Computational intensiveness, latency in TX confirmation and energy consumption by forgoing the use of PoW in block mining	Not mentioned	Not mentioned	Not mentioned	Blockchain scalability, energy & computational intensiveness	Scalability, by reducing the number of TXs to be mined in a block	Scalability, by storing actual data on the off-chain DHT

4.6.8 Blockchain-driven IoT for Food Traceability with an Integrated Consensus Mechanism

Key Features. Authors in [340] have proposed a Blockchain-IoT based food traceability system (BIFTS) to provide reliable traceability of perishable food. The proposed framework also enables customized food shelf life and quality decay performance. BIFTS focuses on the traceability of food at various stages of the shipment starting from the farm to the food processing facility, and further to the distributor and the retailer, and finally delivery to the customer. The researchers use blockchain technology for data integrity. However, the data collected from the IoT are stored in cloud storage. While the events and data payload IDs generated from IoT interactions are stored in the blockchain. The proposed model also introduces a customized blockchain consensus protocol named "Proof of Supply Chain Share (PoSCS)," which is an imitation of the PoS consensus protocol. In PoSCS, the blocks are minted/forged by the validator nodes. Whereas, the validator nodes are the stakeholders in the food supply chain. The factors for the selection of a validator node include transit time, stakeholder analysis, and the shipment volume. The authors claim that the proposed model addresses the issues of blockchain scalability and energy/computation intensiveness.

Voids. There is no discussion about data security and privacy. The authors did not plan for any contingency in which a validator node misbehaves and forges blocks with invalid IoT data. Moreover, it is inferred that a stakeholder with the largest stake and shipping volume may have the monopoly of minting maximum blocks. Hence, block validators can be predictable. Moreover, secure input of IoT data to the blockchain/BIFTS, and ensuring device integrity have been identified as open challenges.

4.6.9 Managing Things' Services through Smart Contracts

Key Features. To exploit blockchain's ability to run smart contracts, "Slock.it" was developed as a commercial product [345]. It is a smart lock called Slock, which is controlled through smart contracts on Ethereum blockchain. In practice, the slock can be any smart device available for rent, such as a bike, car, computer, etc. An app controls conventional smart devices for a pre-defined purpose. However, using smart devices through the blockchain gives the users unlimited options and use cases such as renting out rooms, cars, bikes, electronic appliances, and parking facilities. In [341], the founder of the "Slock.it" demonstrates the complete process of renting a slock. The perceived working of Slock.it is shown in Figure-4.11. Firstly, the owner registers its slock/item for rent, on the app provided by the blockchain service provider. As soon as the owner registers his device, the device gets a private/public key pair in the smart contract. The owner then sets the deposit amount (same as security) and the cost per minute/hour/day for a particular slock/item.

On the other side, when the client wants to rent a service/slock, the client just selects the desired item/slock and then clicks the rent-it button to sign the contract. The client can also see the amount required to be deposited and the cost per minute/hour for the said service. As soon as the customer clicks rent-it, a TX is initiated on the blockchain. The TX confirmation can take some

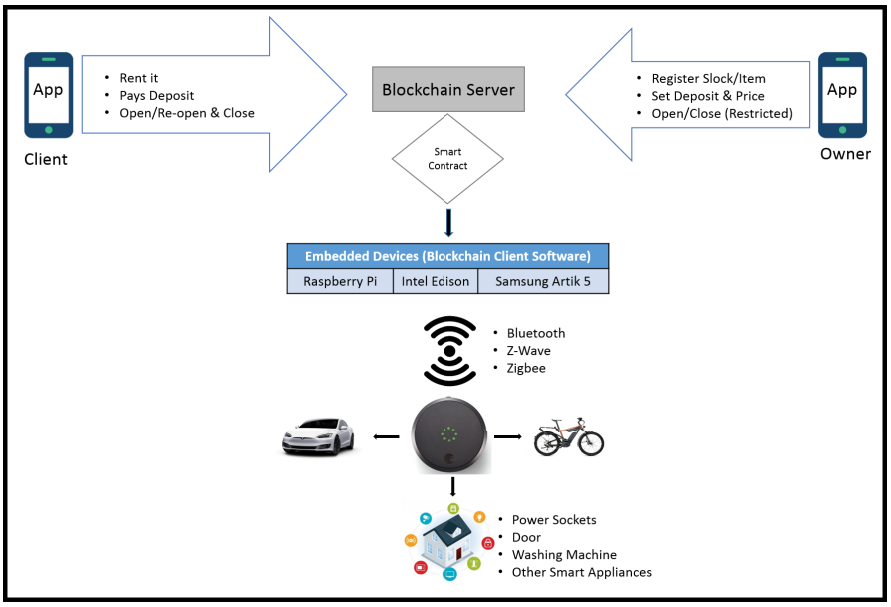

Figure 4.11: Managing the IoT device services using smart contracts

time equivalent to one or two blocks generation period depending upon the settings of the service provider. Once the TX is confirmed, the client can click the open option and access the service. When the customer has used the service, he can terminate the service by clicking the close button on the app. As soon as the service is closed, a TX is initiated on the blockchain, and the client gets his balance money (Balance = Deposit - Cost of service) through the smart contract.

The slocks/smart-devices are integrated with a blockchain-based smart contract hosted on a single or distributed blockchain servers, through embedded devices running a blockchain client software. The embedded device can be a Rpi, an Intel Edison, Samsung Artik-5, or any other System on Chip (SoC) solution. The blockchain client communicates with smart devices/slocks through Bluetooth, Z-Wave, ZigBee, or any other communication protocol supported by the service provider. Considering the scalability factor, only initial open and last close TXs are recorded in the blockchain. The rest of the open and close TXs during usage of the rented service/slock are termed as whisper messages and are not stored in the blockchain. However, these messages are verified through the private/public key of the client. The scalability issue can be managed differently depending upon the system architecture and the type of devices being used.

Voids. Apart from inherent Ethereum blockchain benefits, Slack.it mostly focuses on the functionality of the product. It is not mentioned what security measures are taken to ensure device security.

4.6.10 Security and Privacy of Data

Considerable work has been done to ensure the privacy of user data on the blockchain-based networks. A data management system for decentralized networks has been proposed in [346]. It protects against issues related to data transparency and auditability, data ownership, and access

control. Moreover, Viral Communications, MIT Media Lab has developed Ethos, a Bitcoin-like network for secure sharing of personal data [347]. However, the suitability of Ethos for its application in an IoT system still requires deep assessment. In addition to this, a privacy-preserving decentralized computation platform named Enigma [212] has been proposed. It ensures the confidentiality of data by implementing secure multi-party computation guaranteed by verifiable secret sharing scheme. Enigma restricts access to complete data by all the nodes, i.e., every node has a secret share of data, and it performs computations on that particular share without leaking information to the other nodes. Such an arrangement decreases memory requirement for embedded devices, and the distributed storage enables the performance of more intense computations on data.

Voids. Although, the idea of decentralized computation in Enigma seems feasible, yet the computation and communication overhead is required to be analyzed for its efficient implementation in an IoT system. Since most of the IoT end devices like sensor nodes communicate using wireless media. Any current or future solution for secure data sharing and distributed computing must comply with the regulations of wireless communications as per respective country's law. The distributed computation schemes like multi-party secret sharing schemes [212] seem very efficient, but their efficacy regarding bandwidth/channel utilization needs to be assessed. E.g., In Europe for the LoRaWAN protocol that operates on the 868 MHz frequency band, the allowable duty cycle is 1% for each user/device [325]. Hence, any blockchain-based secure data sharing platform for IoT systems should cater to such limitations.

4.7 Gap Analysis

In-spite of inherent benefits of the blockchain, i.e., TX integrity, TX authentication, non-repudiation, and auditable log of events, etc., there are numerous challenges (highlighted in Section-4.5), that need due consideration for secure adoption of blockchain in IoT. Further elaborating on these issues, firstly, the existing consensus protocols such as PoW, PoS, PoET, IOTA, and PoA are designed for public blockchains (PoS and PoET also support permissioned blockchains) in which the miner is selected based on some lottery scheme. Thus, a block is mined by the lottery winner without network consensus. The previous block is confirmed only, once the next miner and the subsequent other miners extend the chain. Hence, these protocols lack instant consensus finality and are prone to blockchain forks. As far as BFT-based consensus protocols are concerned, although they do provide consensus finality and avoid forks along with low latency in TX confirmation, yet they are prone to DoS attacks. Moreover, with an increase in the number of replicating/validator nodes, the communication complexity also increases. On the other hand, IOTA provides low latency in initial TX approval. However, it is currently not determined that after how much time and indirect approvals, the TX stands confirmed. This is an important aspect of near-real-time IoT service management, such as toll payment by the smart car, payment for gas, parking fees, etc. Hence, IoT-centric consensus protocol is required to be designed and developed duly considering factors such as IoT centric TX/block validation rules, resistance to DoS attacks (exploiting timing assumptions), increased fault tolerance ($> 1/3$ faulty nodes), consensus finality, and low communication complexity.

Table 4.8: Gap analysis

Challenges	**Applications**									
	ADEPT	Smart Cities	Firmware Update	Smart Home	VANETS	IoT eBusi-ness	SCM	BIFTS	Slock.it	Enigma
IoT centric consensus protocol	X	X	X	X	X	X	X	Supply Chain Centric	X	X
IoT focused TX validation rules	X	X	X	X	X	X	X	X	X	X
Scalability	Yes	X	Yes (By not storing firmware files on the blockchain)	Yes (By storing device data on cloud storage)	X	X	X	Yes	Yes (By limiting the number of TXs to be mined in a block)	X
Secure device integration	X	X	X	X	X	X	X	X	X	X
Protection against device compromise	X	X	X	Yes (By limiting outward data flow from the devices)	X	X	X	X	X	X
Secure firmware update	X	X	Yes	X	X	X	X	X	X	X
Data Security	Yes	X	Yes	Yes	X	X	X	Not discussed	Not mentioned	Yes
Privacy-preserving computation	X	X	X	X	X	X	X	X	X	Yes

If we look at the blockchain-based IoT applications, discussed in Section-4.6, Table-4.8 shows a synthesis matrix, that pitches the challenges identified (Section-4.5) against the blockchain-based IoT applications. It is evident that most of the challenges are not tackled by any of the blockchain applications. In this regard, the foremost issues are lack of IoT-focused consensus protocol and TX validation rules followed by secure device integration and secure firmware update. Only three applications, i.e., firmware update, smart home, and BIFTS mention consensus protocol. In that, the firmware update application only comments that it uses PoW consensus for firmware verification. However, no further details are given as to how it manages PoW's computation and energy costs and latency in TX processing. It also does not comment on any distinction between the miner and normal nodes. On the other side, the smart home application uses a proprietary blockchain platform and does not use PoW consensus protocol because of its high computation and energy costs and latency in TX confirmation. However, the proposed scheme does not mention how it selects miners for subsequent block mining. Currently, it seems that only the smart home miner mines the block for all the devices in a particular house, which is against the trust-free and decentralized architecture of the blockchain. Similarly, BIFTS proposes a supply chain centric consensus protocol to avoid energy and computational overheads of the PoW consensus. However, the block validators are selected based on transit time, stakeholder analysis, and shipment volume. Nonetheless, BIFTS does not plan for any eventuality in which a validator may act maliciously and forges invalid blocks. It is also inferred that a rich stakeholder with the largest stake, and shipping volume may propose maximum blocks. Besides, rest of the applications do not discuss any issue related to consensus protocols.

Third hitch is regarding the scalability of the blockchain. Five applications, i.e., ADEPT, secure firmware update, smart home, BIFTS, and Slock.it address this issue. Generally, scalability can be interpreted in terms of the size of the blockchain and latency in TX confirmation concerning network expansion. A typical IoT system, e.g., a smart city environment monitoring system may comprise thousands of embedded devices with limited memory and power resources. The constraint resources cannot store the ever-increasing size of the blockchain, which is required to maintain a full node. Hence, this aspect limits the number of full nodes in the network. However, if there are fewer full nodes with mining capabilities, then it means the workload of mining TXs will be on limited mining nodes, which may create a bottleneck and result in high latency in TX confirmation. Similarly, as proposed in BIFTS, blockchain only stores the event or data payload IDS. Whereas, the realtime IoT interactions are managed in a cloud database. This implies that there has to be a compromise between transparency and size of data stored on the blockchain. Therefore, due diligence is required in resolving the issue of scalability, as this limitation has a significant impact on the design of blockchain-based IoT systems.

The fourth issue is of secure IoT device integration with the blockchain. None of the applications brace this problem. Therefore, there is a need to design and develop a method to securely interface IoT devices with the blockchain such that the data from heterogeneous IoT devices can be directly sent to the blockchain. It is also essential to ensure the integrity of IoT

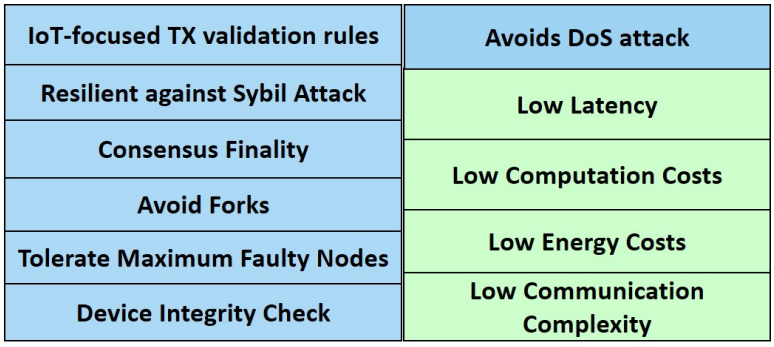

Figure 4.12: Considerations for the IoT-centric consensus protocol

devices for correct operation in a trustless environment, without the use of any additional hardware, e.g., trusted platform modules. The factor of secure hardware is specifically mentioned here, as in practice, manufacturers reduce the cost of IoT devices such as CCTV cameras, embedded sensor modules, smartwatches, smart TV, etc., by cutting investment on security hardware/features and just focusing on the application features.

Protection against malware attacks and runtime firmware/software upgrades is another lacking area. Although, authors in [334] propose a blockchain-based firmware update procedure. However, the proposed scheme does not protect against node compromise attacks in which node hardware configuration is changed to allow for back-door access later. Hence, an attacker can install malicious code in the memory of a node to launch further attacks on the network like espionage and DoS by initiating unnecessary network traffic to target legitimate users/applications.

Although most of the applications do not consider or need data security in the form of data encryption. However, it is no more an un-addressed issue as the blockchain-platforms such as Hyperledger Fabric, and IBM ADEPT, provide data confidentiality and data privacy. Another important predicament is related to the privacy of sensitive data. In a blockchain-based distributed system, preserving the privacy of sensitive user data such as financial information, health data, personal/house security data during distributed processing is still a big challenge. The distributed computation scheme Enigma [212] seems very efficient, but its efficacy regarding bandwidth/channel usage needs to be assessed. Hence, any future solution should also cater for computation/transmission overheads and bandwidth utilization.

4.8 A Way Forward

4.8.1 IoT-Centric Consensus Protocol and TX Validation Rules

The design and development of an ideal consensus protocol for an IoT environment demand that the requirements of a consensus protocol for a blockchain-based IoT system be distinguished

from existing general-purpose and cryptocurrency oriented consensus protocols. Some of these requirements are shown in Figure-4.12. The points mentioned in blue color are concerning security/consistency and the points shown in the green color pertains to the performance requirements. The foremost requirement for IoT systems is that the TXs should be validated based on IoT-centric TX validation rules. It is an essential requirement since every new TX in IoT is mostly independent of the previous TX and an incident or change in environmental conditions can influence the change in the sensor readings. Therefore, IoT TX validation rules should be carefully drafted and they must incorporate environmental context, e.g., in a smart home, the fireplace is ignited, only if the camera or any other sensor also detects the presence of a human in that room. It means a sensor reading is validated based on the environmental context and not in isolation. The consensus protocol should also be robust against Sybil attack and must have consensus finality to avoid forks. Other than avoiding forks, consensus finality is equally vital for achieving minimum latency in TX confirmation and the ultimate high TX throughput.

Moreover, IoT systems are also vulnerable to physical or cyber-attacks. Recently, a cyber-attack named "Mirai" [54] infected a large number of IoT devices including DVR and CCTV cameras and turned these devices into bots. The compromised devices were then used to launch a DDoS attack on a DNS service provider "DYN" by directing huge data traffic in the form of millions of DNS lookup requests. Whereas, if we look at the BFT-based protocols, most of them can only tolerate less than $1/3$ faulty nodes. Therefore, an IoT-centric consensus protocol must have the capability to sustain maximum possible faulty/dishonest nodes. An important consideration to lessen the effect of faulty nodes is to carry out a random integrity check of the validator/mining nodes so that no dishonest node participates in the consensus process [235]. In addition to the security requirements, there are some performance considerations as well. These include low computation overhead, low energy consumption, and less communication complexity.

4.8.2 Managing Blockchain Size

To address the issue of scalability concerning the management of ever-increasing blockchain size on light/embedded IoT devices, various blockchain architectures are being proposed, such as sidechains and treechains. An example of a sidechain is a decentralized P2P network designed for multi-party privacy-preserving data storage and processing [212, 346]. The proposed model implicitly improves the issue of blockchain scalability by storing user data on an off-chain network of private nodes in the form of DHT [348]. The blockchain only contains the pointers/references to data, and not all the nodes replicate all TXs.

IBM [333] also addresses the issue of blockchain size by introducing a concept of universal and regional blockchains. It is achieved by categorizing the network nodes into light peers, standard peers, and peer exchanges depending upon their processing, storage, networking, and power capabilities. The light peers consist of embedded devices, such as Arduino and Rpi-based sensor nodes. These nodes only store their blockchain address and balance and rely on other trusted peers to obtain TXs relevant to them. Whereas the standard peers have more processing power and storage capacity than light peers. They can store some of the recent TXs of their own and the

light peers in their neighborhood. Finally, the peer exchanges have high storage and computing capabilities, and they can replicate complete blockchain data with an additional feature of data analytics services. Also, as per NIST [349], resource-constraint devices may maintain a compressed ledger containing only their TXs.

Authors in [204] and [211] also propose a scalable two-layer blockchain architecture to log distributed database TXs. The first layer represents a permissioned blockchain comprising a miner each from respective federation members. The miners in layer one are selected randomly based on a fast consensus protocol. The hash of the layer one blockchain is periodically stored on the second layer using PoW, to ensure the integrity of the hashes. Hence, if a malicious node alters the log in the first blockchain, the hash of the data would be different as in the second layer. Hence, forgery can easily be detected. To achieve scalability in the proposed scheme, especially at layer one, the authors proposed the technique of data sharding, in which every miner maintains a DHT-based ledger based on keyspace partitioning and only handles TXs for specific subsets of keys. Thus tuning TX load on miners and making the system more scalable.

Another solution proposed for the scalability of the Ethereum blockchain is called "Plasma" [350]. It uses a series of smart contracts to create hierarchical trees of sidechains, which can be thought of as "subchains." The subchains live within a parent blockchain and periodically communicate with the root-chain (Ethereum). The subchains are off-line; hence, theoretically, there can be as many subchains as desired [351]. Similarly, BigchainDB [352] introduces a blockchain database that utilizes the benefits of both the blockchain and the big data distributed database. It integrates the immutability and decentralization of the blockchain with the high throughput and fast TX settlement time of big data distributed database.

4.8.3 Improving Upon TX Confirmation Time

TX confirmation time can also be associated with the problem of blockchain scalability. In current public blockchains such as Bitcoin and Ethereum, the miner nodes are required to store the complete blockchain and validate every TX in an order. This arrangement does help in ensuring the security of the system but can also be prone to bottlenecks in case of high TX volume. Since the blockchain cannot process more TXs than a single node can. One of the methods being researched to reduce TX confirmation time is "Sharding" [353]. It means a subset of miner nodes process a subset of TXs (as shown in Figure-4.13). The subset of miner nodes should be populated in a way that the system is still secure, and at the same time, several TXs can be processed simultaneously [351, 353]. In its purest form, each shard has its own TX history, and it is affected only by the TXs it contains. E.g., In a multi-asset blockchain, there are n shards, and each shard is associated with one particular asset. In more advanced forms of sharding, TXs on one shard can also trigger events on some other shard. This is usually termed as cross-shard communication. Correspondingly, in addition to reducing TX latency, Sharding also improves system scalability. However, currently being in a novice state, there are numerous challenges that should be resolved before sharding is adopted publicly. Some of these challenges include; cross-shard communication, fraud detection, single-shard manipulation, and data availability attacks [353].

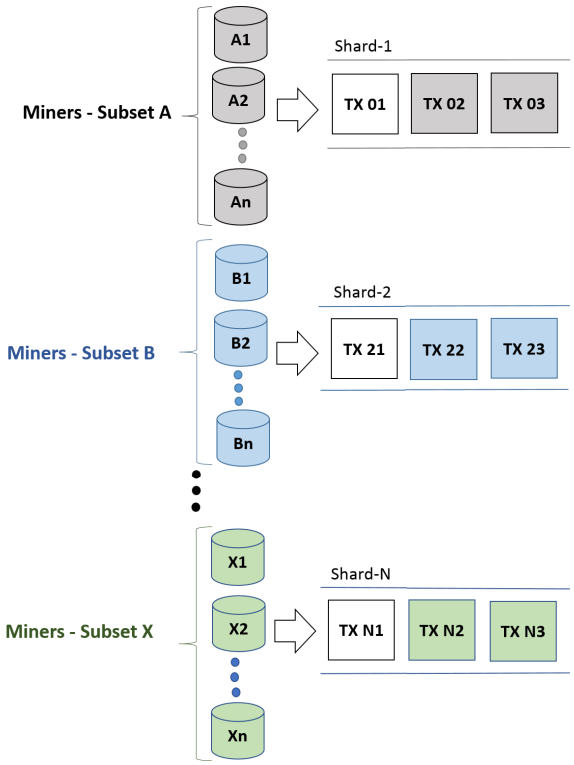

Figure 4.13: Sharding

Another approach to reduce TX processing time is "Raiden." It proposes the use of state channel technology to scale the Ethereum network off-chain and to facilitate micro-TXs between IoT devices [351]. The off-chain TXs will allow a set of nodes to establish payment channels between each other without directly transacting with the Ethereum blockchain. Hence, off-chain TXs would be faster and cheaper than on-chain TXs because they can be recorded immediately, and there is no need to wait for block confirmations. However, it is believed that channel-based strategies can scale TX capacity only but cannot scale state-storage. Moreover, they are also vulnerable to DoS attacks [353].

In another development, to address Bitcoin blockchain's problems of scalability, high TX fee, and requirement of substantial hardware resources, a blockless architecture named "IOTA" have been introduced [258]. IOTA is a distributed architecture based on DAG called "Tangle" [297], instead of a conventional blockchain. It aims to promote the machine economy, in which smart devices can interact with each other by making the smallest possible, nano-payments. To ensure fast TXs, IOTA does not require a TX fee. Moreover, the consensus (TX validation) and normal TX process are also inter-knitted, i.e., before making a new TX, each user randomly approves/validates the previous two TXs. IOTA achieves high throughput by parallelizing the TX validation process. Hence, an increase in the number of new TXs on the Tangle is inversely proportional to the TX settlement time [354]. Therefore, an expanding network contributes well to the overall security and fast TX settlement. The two TXs to be approved by every new TX are randomly

selected based on the MCMC method. A TX getting more and more direct/indirect approvals is considered to be more accepted by the network. Hence, it would be difficult for anyone to double-spend that particular TX. The difference between IOTA and a typical blockchain architecture is shown in Figure-4.14 [354].

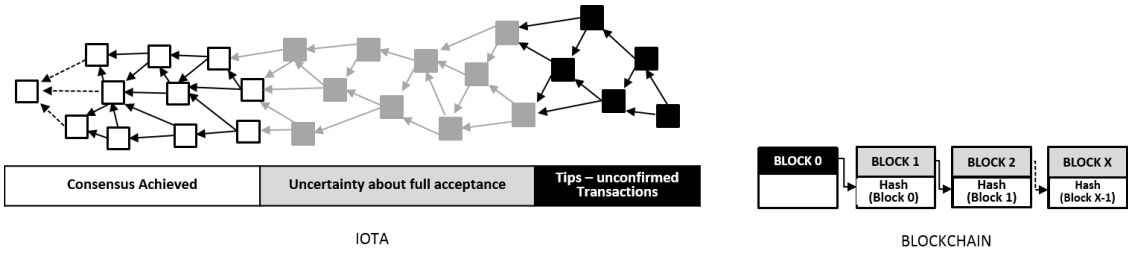

Figure 4.14: IOTA vs. Blockchain

4.8.4 Secure IoT Device Integration with the Blockchain

In addition to securing the web UI and mobile App, IoT device integration with the blockchain can be augmented by device enrolment, in which only approved devices be allowed to communicate with the blockchain and call smart contract methods. Correspondingly, smart contracts can restrict access to selected methods to a specific node only. Concerning the physical security of IoT devices, all the unnecessary ports such as JTAG and UART should be blocked. Since any open port can be used by an adversary to access the device and make malicious changes. Moreover, most of the commercially available IoT devices, such as sensing devices, do not have a secure execution environment due to cost effects. Therefore, the device integrity check should frequently be performed to ensure its legitimacy [235].

As of today, most of the IoT systems depend on a certain cloud platform due to computational and storage scarcity, and because of the same, resource-constrained IoT devices cannot be used as a miner or full nodes in a blockchain network. Hence, to ensure a smooth transition from cloud to blockchain-based network, the IoT systems can leverage fog computing components that already follow some degree of distribution and are more resourceful than IoT devices. The fog nodes can function as blockchain miners and can facilitate direct interaction between IoT devices and the blockchain. E.g., As shown in Figure-4.15, the fog nodes can incorporate blockchain miner nodes to collect and mine the TXs received from the IoT devices in a block. The IoT devices have enough resources to be the full nodes. Hence, they can store the blockchain and also route and validate the TXs. In this way, most of the TXs from the IoT devices would be propagated to both the fog nodes. Hence, IoT can leverage existing fog computing infrastructure to adopt blockchain technology, until IoT devices are manufactured with embedded blockchain mining functionality to gain on the maximum benefits of blockchain's distributed architecture.

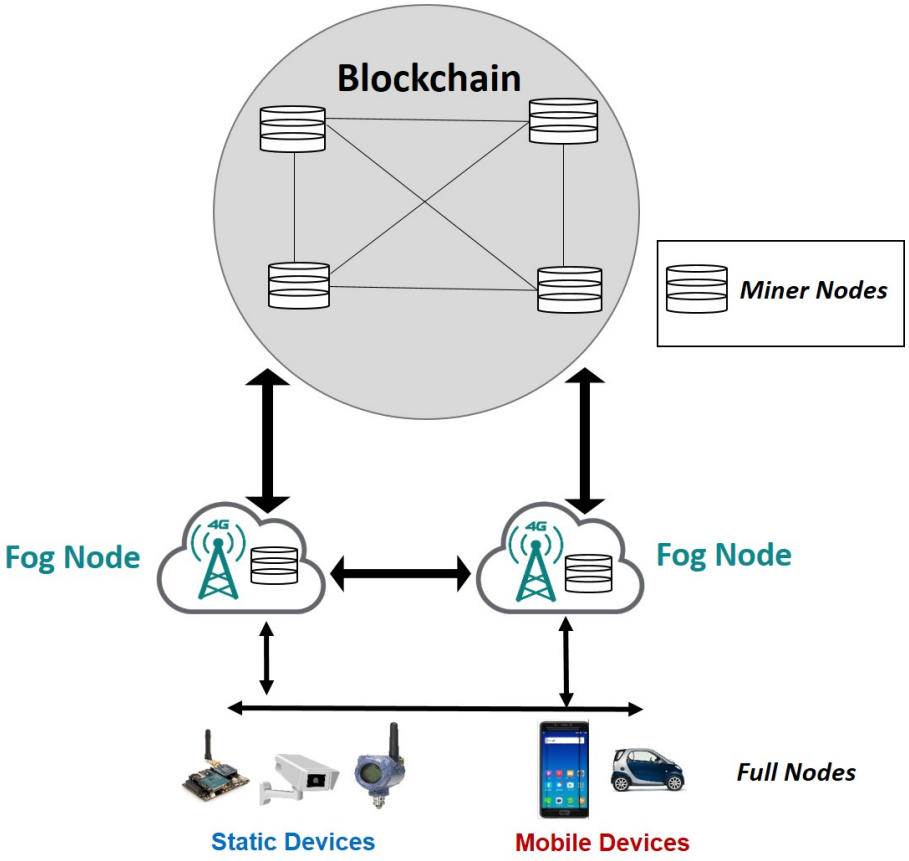

Figure 4.15: Blockchain and the IoT integration using fog nodes

4.8.5 Integration of IoT Communication Protocols with the Blockchain

To integrate blockchain protocols with the communication layer of IoT, [333], and [223] recommend the use of TeleHash as the messaging protocol, which is based on Kademlia DHT [346]. It is a lightweight and secure P2P network protocol that uses encryption for secure mesh communication across multiple platforms [359]. TX records can be converted into blocks and further broadcast into the blockchain network.

4.8.6 Resolution of Bitcoin Blockchain's Limitations

Till now, we have analyzed every aspect of the blockchain, from its basic concepts to the advancements in blockchain platforms, related challenges and latest trends in blockchain-based IoT applications. However, it is vital to present a consolidated gist of the evolution in blockchain technology that aims to mitigate Bitcoin blockchain's limitations. This summary will help blockchain and IoT researchers to understand related technologies and find their way forward to resolve blockchain-based IoT issues. Hence, Table-4.9 pitches Bitcoin blockchain's limitations and vulnerabilities against requisite blockchain technologies and applications that promise to abate respective limitations.

Table 4.9: Resolution of Bitcoin Blockchain limitations

Bitcoin Blockchain Limitations	Advancement in Blockchain Platforms/Applications/Technologies
Energy and computation intensive PoW consensus	PoS [275], PoET [286], PoB [287], PoA [283, 284], BFT-based consensus protocols [207–210, 293]
Lack of consensus finality and forks	BFT-based consensus protocols [207–210, 293]
Latency in TX confirmation	Ethereum (GHOST, Casper) [245], Hyperledger Fabric (PBFT, SIEVE) [261], Bitcoin-NG [328], and BFT-based blockchains [210]
Low Throughput	BFT-based blockchains (Multichain [263], Hyperledger Fabric [261])
De-anonymization (Linking attacks) [242]	Monero [330], Zerocash [331],
Scalability (Size of blockchain)	Universal and regional blockchains (IBM) [333], Sidechains [346, 350], Data compression (NIST) [349], Scalable blockchain architecture [204, 211], BigchainDB [352]
51% attack [236], Double-spending [24, 355]	BFT-based consensus protocols [207–210, 293]
No runtime firmware/software update	Secure firmware upgrade [334], Gitar [356], RemoWare [357]
Data privacy	Multichain [263], Quorum [264], Hyperledger Fabric [261], Hawk [332], DHT [348]
Privacy-preserving computation	Enigma [212], Homomorphic encryption [172]
Limited scripting	Smart contracts supported by Ethereum [245], Hyperledger Fabric [261]
Legal issues in smart contracts	Alastria [358] (Idea of a national regulated blockchain)
Public/Permissionless blockchain	Private/Permissioned blockchains Ethereum [245], Multichain [263], Quorum [264], Hyperledger Fabric [261]

4.9 Summary and Future Work

In this chapter, we initially introduced IoT security and performance requirements and important blockchain concepts. Then, we analyzed the impact of blockchain technology on IoT, followed by identification of challenges to blockchain's integration with IoT. Later, we reviewed various blockchain-based IoT applications to highlight the trends in IoT applications and the blockchain issues resolved by these applications. In the end, we carried out the gap analysis and recommended a way forward to resolve some of the significant challenges that hinder the adoption of blockchain in the IoT environment.

Based on the discussion in this chapter, it is concluded that no doubt blockchain can satisfy most of the security and performance requirements of IoT. However, there is a need for devising a mechanism that not only offers security guarantees for a typical IoT system but is also scalable and has low resource requirements. Therefore, future research should endeavor to resolve issues including privacy-preserving computation on sensitive user data, secure IoT device integration with the blockchain, device authentication and integrity check, software integrity check and run-time/synchronized software update, IoT-centric consensus protocol and IoT-focused TX validation rules. As far as IoT performance requirements are concerned, some of these requirements are addressed by Hyperledger Fabric. However, low communication complexity and scalability should also be kept in view while designing an ideal IoT-oriented consensus protocol.

”Privacy is an inherent human right, and a requirement for maintaining the human condition with dignity and respect. It is about choice, and having the power to control how you present yourself to the world.”

- Bruce Schneier

5

PrivySharing: A Framework for Privacy-Preserving and Secure Data Sharing

This chapter introduces “PrivySharing,” a blockchain-based innovative framework for privacy-preserving and secure IoT data sharing in a smart city environment. The proposed scheme is distinct from existing strategies in many aspects. The data privacy is preserved by dividing the blockchain network into various channels (Chs), where every Ch comprises a finite number of authorized organizations and processes a specific type of data such as health, smart car, smart energy or financial details. Moreover, access to users' data within a Ch is controlled by embedding access control rules in the smart contracts. In addition, data assets within a Ch are further isolated and secured by using private data collection and encryption techniques, respectively. Likewise, the REST API that enables clients to interact with the blockchain network has dual security in the form of an API Key and Open Authorization standard “OAuth 2.0.” The proposed solution conforms to some of the significant requirements outlined in the EU GDPR. PrivySharing also has a system of reward in the form of a digital token named “PrivyCoin” for users sharing their data with stakeholders/third parties. Lastly, the experimental outcomes advocate that a multi-Ch blockchain scales well as compared to a single-Ch blockchain system. This work has been published in the *Journal of Computers and Security*, Imran et al. [360], and an initial version was also presented at the 16^{th} International Conference on Security and Cryptography (SECRYPT) [361].

5.1 Background

The ubiquitous use of IoT ranges from ICS to e-Health, e-commerce, smart cities, agriculture, SCM, smart cars, CPS, and a lot more. However, the data collected and processed by IoT systems,

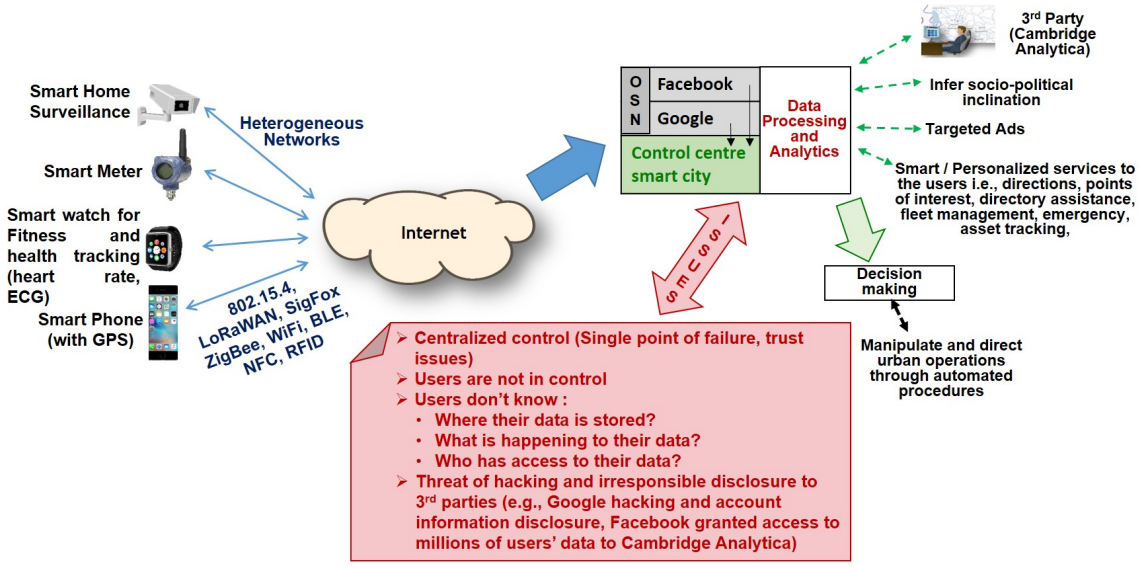

Figure 5.1: Issues in the smart city environment

especially the ones with centralized control, are vulnerable to availability, integrity, and privacy threats. Similarly, a smart city network also suffers from numerous security and privacy issues [362, 363], such as threats to privacy, integrity, and availability of user data, false data injection [364], vulnerability to Sybil attack [365], and single point of failure due to centralized control. If we look at Figure-5.1, the user data collected by numerous sensors is stored and processed by OSN, smart city control center or various other smart city components such as ITS, health emergency response, fire and rescue, etc., These components (with mostly centralized control) process user data for the provision of various services to the data owners and third parties. Although such a centralized control may look effective from the outside, yet it has some significant security concerns.

Centralized control is subject to a single point of failure in case of a cyber-attack or other technical malfunctions [22]. Moreover, it also has trust issues, as the users have to put their trust in the entity that is handling their data. Hence, users have no control over their data assets. Further concerns for data owners include: Users do not know where their data is stored and what is happening to it. Who has access to it, and is there any unauthorized disclosure to the third parties. The above-mentioned, users concerns are very much real as the disclosure of personal data leakage concerning millions of users by Facebook Inc. [27, 28] and a bug in Google Plus [29] that resulted in the exposure of personal information of approx 500,000 users is a candid example of one of cloud/OSN vulnerabilities.

Moreover, any smart city application is believed to store, process, and analyze users' data. Hence, every security solution developed for a smart city environment must comply with the undermentioned key requirements of EU GDPR [171] while handling users' data:

a. Personal data should be processed only with the consent of the data owner.

b. Any technology dependent on user data must preserve user privacy by design.
c. The gathering, processing, or use of personal data should be in accordance with the instructions based on a mutual contract between the user and the third parties.
d. The owner of data has the right to access the information concerning the processing of his data, i.e., which third parties have access to what data and how they use it.
e. It is the right of the data owners that their data be erased immediately once it is no longer needed.
f. The system should be transparent such that individuals know about the collection and use of their data.

As far as IoT security is concerned, researchers and security analysts are trying to leverage cryptographic security benefits of blockchain to resolve security and privacy issues of IoT. Hence, we believe that a carefully selected blockchain technology with an insightful business network design can resolve most of the data integrity and privacy issues of IoT applications such as a smart city.

5.1.1 Related Work

Security researchers around the world are developing and investigating ingenious ways to implement blockchain in the IoT environment. These use cases aim to take advantage of the inherent benefits of the blockchain, such as decentralized control, immutability, cryptographic security, fault tolerance, and capability to run smart contracts. Recently, researchers [366] presented a blockchain-based data-sharing framework for a smart city environment. The framework called "SpeedyChain" focuses on reducing the TX settlement time for real-time applications such as smart cars and also aims to ensure user privacy. Moreover, it ensures data integrity, tamper-resistance, and non-repudiation that are some of the intrinsic benefits of the blockchain. In another work, Pradip Kumar and Jong Hyuk proposed an SDN and blockchain-based hybrid network architecture for a smart city [367]. The proposed architecture addresses usual smart city issues including high TX latency, security and privacy, bandwidth bottlenecks, and requirement of high computational resources. In the proposed model, the smart city network is divided into a distributed core network comprising resourceful miner nodes and the centralized edge network constituting inept devices. The edge nodes store access policies for locally registered nodes. Authors claim that in addition to reducing TX latency, and reduced network bandwidth, the proposed model limits the effects of a node compromise to the local area.

Additionally, authors in [368] proposed a smart contract based sharing economy services in a smart city. The proposed model uses Artificial Intelligence (AI) for data analytics and also uses blockchain to store the results. Similarly, Biswas and Muthukkumarasamy [223] presented an overview of a blockchain-based security framework for secure communications between smart city entities. Authors claim that the integration of the blockchain with devices in the smart city will provide a shared platform where all the devices would be able to communicate securely. However, the researchers did not disclose some necessary details about the type of blockchain platform, consensus protocol, and TX/block validation techniques adopted in the smart city application.

In another endeavor [369, 370], security researchers have proposed solutions to address various user privacy issues in ITS. Nonetheless, they do not cater to the challenges of smart cities, such as trustless data sharing among multiple organizations. Similarly, Ali Dorri and Raja Jurdak proposed a secure, private and lightweight architecture of a blockchain-based smart home application [240, 335]. It aims to solve certain blockchain issues such as computational intensiveness, latency in TX confirmation and energy consumption. To reduce computational overhead and energy consumption, each block is mined without any PoW. Moreover, the latency in TX confirmation is reduced by considering a TX true, whether it is mined in a block or not. Also, the proposed scheme utilizes cloud storage to ease up the memory requirements for smart home devices. However, there are many security concerns that need further explanation with logical reasoning [234]. Likewise, another team of researchers proposed an Ethereum Blockchain [245] based mechanism to manage IoT devices [241]. However, Ethereum Blockchain does not provide data privacy.

In another work, to avoid issues concerning the single point of failure in a centralized system, researchers proposed an Ethereum Blockchain based decentralized, self-managing VANET with a challenge-response based authentication [336]. However, the proposed scheme does not explain the procedure of consensus and block mining. There is also no discussion about the type of information to be published on the blockchain and the latency in TX confirmation. Above all, Ethereum Blockchain does not provide data privacy and confidentiality.

Correspondingly, Yu Zhang, and Jiangtao Wen proposed an Ethereum Blockchain based decentralized electronic business model for the IoT [337]. However, the proposed solution mostly focused on the working of the e-business model, so there is a lack of discussion on technical aspects such as block mining mechanism, modalities of implementing blockchain on IoT devices, and the methodology of achieving data confidentiality and privacy. Similarly, in another work [371] authors introduced a blockchain-based security framework for IoT implementations. Nonetheless, the proposed solution focuses on data authentication and secure communication between the sensor devices and the controllers. The researchers make use of the received signal strength (RSSI) of the message sent by a sensor device as a parameter to ensure the randomness of data to avoid replay and data forging attacks by a MITM attacker. Few other researchers have also proposed a blockchain-based approach of exchanging data in the smart city between nontrusted organizations [372]. In this regard, if a third party queries some data, e.g., a credit report concerning a user, then the executor node gets the input in the form of private data from the respective organization through a local private API. The data is encrypted with an organization's private key and is decrypted once in the executor sandbox using the organization's public key. Hence, the querying party receives only the processed data and does not see the original data itself.

Since the GDPR legislation came in to effect on 25^{th} May 2018, researchers have been working on various aspects of data protection to develop GDPR compliant data protection/processing frameworks. In this endeavor, [373] proposed a blockchain-based design concept for developing GDPR compliant data management platforms. The solicited framework shares and revokes the sharing of user data only with the consent of the data owner. Moreover, the blockchain-based

framework can also endorse the service providers for being correctly following the GDPR policies or not. As per the devised concept, only data owners and data controllers can create, update, and withdraw consent, and only authorized entities can process user data. The proposed mechanism uses blockchain to handle authentication, authorization, and data access control token validation. Whereas, the data is stored in a centralized resource server that is assumed to be a trusted party. Apart from the resource server being a trusted party, the proposed solution does provide some security guarantees; however, it seems to have high communications complexity. As authors also claim that due to increased message overhead, the proposed scheme does not support high performance and scalability since the TX latency significantly increases and throughput decreases with the increase in the number of nodes. Similarly, [374] recommended a conceptual architecture of a human-centric and GDPR compliant Blockchain-based Personal Data and Identity Management System (BPDIMS). The authors focus on designing a framework, which is transparent and provides data owners with full control over the usage of their data. The researchers address specific issues concerning data usage, i.e., user consent, transparency of data processing, purging of user consent, reward mechanism for users, data integrity, and confidentiality. However, this work is still at conceptual stages and does not present any technical details or performance evaluation.

Similarly, Ricardo et al. proffered a blockchain-based scheme to facilitate data accountability and provenance tracking [375]. Data provenance tracking is achieved by maintaining a list of references to the data provided to the controller. The list is updated whenever some data is sent to the data controller/service provider. Whereas, data accountability is accomplished by specifying restrictions on data usage in smart contracts. The restrictions are defined under the domain of a preventive mechanism, using a security policy language recommended by Model-based Security Toolkit (SecKit). The preventive mechanism denies actions such as allow, deny, modify, or delay the operations concerning data usage to the data controllers. The authors primarily discuss various design choices for the data usage contract models while considering the provision of maximum data provenance information to the data owners in a trusted and privacy friendly-manner. The sample contract models are evaluated based on gas consumption in Ethereum virtual machine.

Correspondingly, authors in [376] introduced a consent management platform named ADvoCATE for IoT data processing. ADvoCATE uses Ethereum Blockchain to preserve the integrity of users' consents and related updated versions. The ADvoCATE may be interpreted as a cloud service platform with various components such as blockchain, intelligent policy analyzer, consent notary, and storage. The consents notary ensures that the created consents are up to date and are also protected against unauthorized modification. Whereas, the intelligence component makes use of the Fuzzy Cognitive Maps (FCM) methodology to identify any rules/policies that contradict with GDPR requirements concerning the handling of users' data. Moreover, whenever an IoT device is installed, the user gives his consent to the data controller/service provider through a smart contract to access IoT device data. The digital consents duly signed by the data controller and the device owner are stored on the ADvoCATE platform, whereas, the blockchain stores only the hashes of these consents for integrity. However, the proposed platform is still in the development phase and has not been extensively tested or evaluated. The authors only highlight the gas (ether)

consumption of smart contracts, and there is no analysis on TX latency, TX throughput, scalability, or communications overhead.

In a similar endeavor [377], Nesrine Kaaniche and Maryline Laurent presented a blockchain-based data usage auditing architecture that provides the data controllers with unforgeable evidence of users' consent. The researchers claim to provide user anonymity by letting the data owners (which are delegated Public Key Generators (PKG)) create a distinctive public-private key pair for each smart contract they initiate to share data with a service provider or a data processor. Moreover, the authors used hierarchical ID-based encryption to prevent unauthorized disclosure. The data is stored on off-blockchain storage, whereas blockchain smart contracts are used to store the hash of data and data usage policy. Also, there is a specific smart contract between the data owner and every other service provider or data processor. However, the architecture is not supported by any performance evaluation, e.g., TX settlement time, block commit time, or latency. In another work, authors evaluated the potential use of blockchain technology to facilitate the transformation of institution-centric exchange of data to patient-centric, and patient-driven data sharing [378]. The researchers recommend that the blockchain can be used to provide transparency over the state of shared data, and related TXs among different stakeholders. In that permissioned blockchains can be more productive in terms of delivering strict access control concerning read-write permissions over users' health data. Authors also believe that the blockchain provides a lower cost of TX verification and data integrity as compared to the traditional systems. It is also accredited that the blockchain can also ensure the availability, swift access, and immutability of health data. Moreover, it can also provide unique identities to all patients. However, authors foresee inevitable glitches in the use of blockchain such as high TX volume of health records, massive storage requirements, and security and privacy issues concerning user data.

Though the research work discussed above has undoubtedly made some significant contributions towards the blockchain and IoT domain. Nevertheless, there are many open issues such as preserving data privacy in a smart city environment, user-defined fine-grained access control, fast TX settlement, users' right to forget (concerning data deletion), an incentive for users to share their data, and distributed storage of user data without centralized control. Therefore, to fill the respective research gaps, we propose "PrivySharing," a blockchain-based secure and privacy-preserving data-sharing framework. The proposed solution aims to protect a smart city environment against most of the data integrity and privacy threats. The experimental results prove that a carefully designed blockchain solution can ensure user data privacy and integrity in various network settings as per the wishes of the data owner. It also effectively prevents false data injection and Sybil attacks. Moreover, PrivySharing complies with some of the significant data security and privacy requirements of the EU GDPR. The significant contributions of this chapter are:

a. Provides protection against most of the external as well as insider attacks threatening user data integrity and privacy in a smart city setting.

b. Compliance with some of the essential requirements of EU GDPR.

c. A blockchain-based solution providing the "right to forget" concerning user data.

d. A scalable (concerning blockchain size), secure, and efficient (in terms of energy consumption and computational requirements) data-sharing framework.

e. User-defined fine-grained access control to data.

f. Providing a transparent and auditable network operation and simultaneously controlling the exposure of users' private data.

g. Secure client access to the blockchain network through a REST API.

h. A reward system for the users for sharing their data with the stakeholders/third parties.

5.1.2 Basic Terminologies

Before getting involved with the detailed architecture of PrivySharing, it is imperative to understand some terminologies specific to Hyperledger Fabric:

a. **Smart Contract (SC).** A SC is a sort of a digital contract based on certain rules between different organizations in the form of an executable code [379]. Blockchain network uses smart contracts not only to encapsulate information but also to automate certain aspects of business TXs. Applications invoke a smart contract to generate TXs that are further recorded on the ledger.

b. **Chaincode.** The difference between smart contracts and chaincode is that a smart contract defines the TX logic that updates the state of a business object contained in the world state. Whereas, a chaincode can be termed as a technical container that may contain multiple related SCs for installation and instantiation. When a chaincode is deployed, all smart contracts within it are made available to the applications [379].

c. **Committing Peers.** Every peer node in the Hyperledger Fabric blockchain is a committing peer. However, a Committing Peer does not have a smart contract installed. It just validates and commits a new block of TXs sent by the ODS to its copy of the ledger [380].

d. **Endorsing Peers.** These are special, committing peers with the capability to run the smart contracts. They prepare, sign and endorse the responses to the TX proposals sent by the clients, in line with the endorsement policy of the respective Ch [380].

e. **Ordering Service (ODS).** It is a collection of some peer nodes that arrange the new TXs in a block and then broadcast that block to all the peers of the concerned Ch [380].

f. **Membership Service Provider (MSP).** While CAs issue X.509 certificates to the network entities, an MSP states that which CAs are accepted by the blockchain network and also determine that which peer nodes are members of which organization. Different MSPs can be used to represent various organizations or multiple groups within an organization. Usually, the MSPs are defined at the network, Ch, and local/peer level.

5.1.3 Organization of the Chapter

The rest of the chapter is organized into five sections. Section-5.2 presents the detailed architecture, operation, and reward mechanism of "PrivySharing." Whereas, security analysis of the proposed framework is performed in Section-5.3. Experimental results, a limitation of the proposed solution, and a way forward to address the limitation are illustrated in Section-5.4. Finally, the chapter is summarized in Section-5.5.

5.2 PrivySharing: Blockchain-based Secure Data Sharing

By leveraging data integrity and smart contract features of the blockchain, various operations in a smart city environment can be securely and autonomously performed. Moreover, blockchain also protects against the adverse effects of server hacking and falsification/modification of permissions [365]. No doubt, people in a smart city environment feel safe while sharing their personal information only when they have the assurance that their personal and sensitive data collected by various devices are fully protected, and they have control over it [381]. Such assurance can only be provided by none other than a prudently selected and assiduously designed blockchain technology.

Table 5.1: List of assets

Data Types	Assets
Health Data	- Health Alert (Heart rate, blood sugar, blood alcohol, etc.) - Full Health History - Insurance Cover - Health Payment Claims - Type of Disease - Current Disease History
Smart Car Data	- GPS Data - Accident Alert - Damage Assessment - Servicing and Auto Payments
Smart Meter Data	- Line Status - Units Consumed and Bill - Consumption Pattern
Surveillance Data	- Equipment Status and Servicing - Security Breach Alert - CCTV Recording
Financial TXs	- Income - Expenses - Tax

5.2.1 Smart City Scenario

We assume that Alice is living in a smart city where every aspect of her life is being monitored and controlled through numerous sensors and smart devices. The critical aspects include monitoring of key health parameters, smart car (operation and service management), smart living (operation and service management) including smart meters (generating data concerning energy

consumption), surveillance cameras, and intrusion detection equipment (generating security-related data), and financial TXs to keep the services running. For better understanding, we have formulated a list of numerous assets (associated with a specific type of data) that Alice owns (as shown in Table-5.1). Based on these assets, Alice can easily decide about the permissions (shown in Table-5.2) to be granted to the stakeholders/third-parties concerning her data assets. Such a distinction among the stakeholders/third-parties further assists Alice to plan and control the access to her data. It is also assumed that all the registered users of the smart city network, whether offline or online, interact with each other through the PrivySharing (blockchain) APIs.

Table 5.2: Assets, stakeholders, and access rights

Assets	Stakeholders	Access Rights
Health Data		
Health Alert (blood alcohol, blood sugar, heart rate, etc.)	- Alice - Primary (Pri) Medical Center - Police	- Read - Read and write - Read
Health History	- Alice - Pri Medical Center - Alice and Pri Medical Center	- Read - Read - Modify (Requires consent of both Alice and the medical center)
Insurance Cover	- Alice - Pri Medical Center - Health Insurer - Alice and Health Insurer	- Read - Read - Read - Modify (Requires consent of both, Alice and the insurer)
Health Payment Claims	- Alice - Pri Medical Center - Health Insurer	- Read and write - Read - Read
Type of Disease	- Alice - Pri Medical Center - Health Insurer - Ministry of Health	- Read - Read and write - Read - Read
Current Disease History	- Alice - Pri Medical Center - 2nd Medical Center	- Read - Read - Read
Smart Car Data		
GPS Data	- Alice - Car Service provider	- Read - Read

Continued on next page

Table 5.2 – *Continued from the previous page*

Assets	Stakeholders	Access Rights
	- Roads/Transportation Authority (ITS)	- Read
Accident Alert	- Alice - Police - Car Insurer	- Read - Read and write - Read
Damage Assessment	- Alice - Car Insurer - Workshop	- Read - Read - Read and write
Servicing and Auto Payments	- Alice - Smart Parking - Security Service Provider - RTA	- Read - Read and write - Read and write - Read and Write
Smart Meter Data		
Line Status	- Alice - Lineman	- Read - Read
Units Consumed and Bill	- Alice - Finance Manager of the Service Provider	- Read - Read and write
Consumption Pattern	- Alice - Operations Manager of the Service Provider	- Read - Read
Total Energy Consumption	- Ministry of Power	- Read
Surveillance Data		
Equipment Status and Servicing	- Alice - OEM/Service Provider	- Read - Read
Security Breach Alert	- Alice - Police	- Read - Read
CCTV Recording	- Alice - Police	- Read - Read
Total Incidents of Security	- Ministry of Interior	- Read
Financial TXs		
Income	- Alice - Bank - Revenue	- Read and write - Read - Read
Expenses	- Alice - Bank	- Read and write - Read
Tax	- Alice - Bank - Revenue	- Read - Read - Read and write

5.2.2 Selection of a Suitable Blockchain Platform

To implement the above mentioned smart city use case, Hyperledger Fabric is selected as the underlying blockchain platform due to its effective data security and privacy-preserving capabilities as compared to other blockchain platforms [234, 382]. Correspondingly, a comparison of various blockchain technologies has been presented in Chapter-4. Hyperledger Fabric is a private and a permissioned blockchain that restricts participation in the network to the authorized parties. The key feature that distinguishes Hyperledger Fabric from other blockchain technologies is that in Hyperledger the blockchain ledger consists of two distinct but related parts, i.e., a blockchain to log the TXs and a world state (a database such as CouchDB [383], and LevelDB [384]) to keep track of the ledger states. Moreover, it is also important to distinguish Hyperledger Fabric from another prominent DLT named "Corda" [385, 386].

According to [387], Hyperledger Fabric has a modular and extendable architecture that can be employed in various industries ranging from healthcare to banking and supply chain. Whereas Corda is more focused on the financial services industry. Moreover, Hyperledger Fabric has the option to develop digital currency/token, which can be used within the blockchain network. On the contrary, Corda does not provide such a feature. Concerning data privacy, Fabric broadcasts a transaction to all members of a channel while Corda does this on a peer by peer basis. Therefore, Corda ledger architecture is more likely to face a higher management overhead as the number of P2P relationships grows as it has to be configured on a case by case basis. Whereas, Fabric's management of ledger visibility is done at the channel configuration level. The only time P2P relationships have to be managed is in the case of its private data feature. Also, Hyperledger Fabric has shown a promising throughput of over 3000 TPS, whereas, currently, it is difficult to characterize that how quickly a Corda network can send payments [385]. Finally, the most critical requirement is the ability to configure user-driven fine-grained access control to data. For which Hyperledger Fabric is the most suitable.

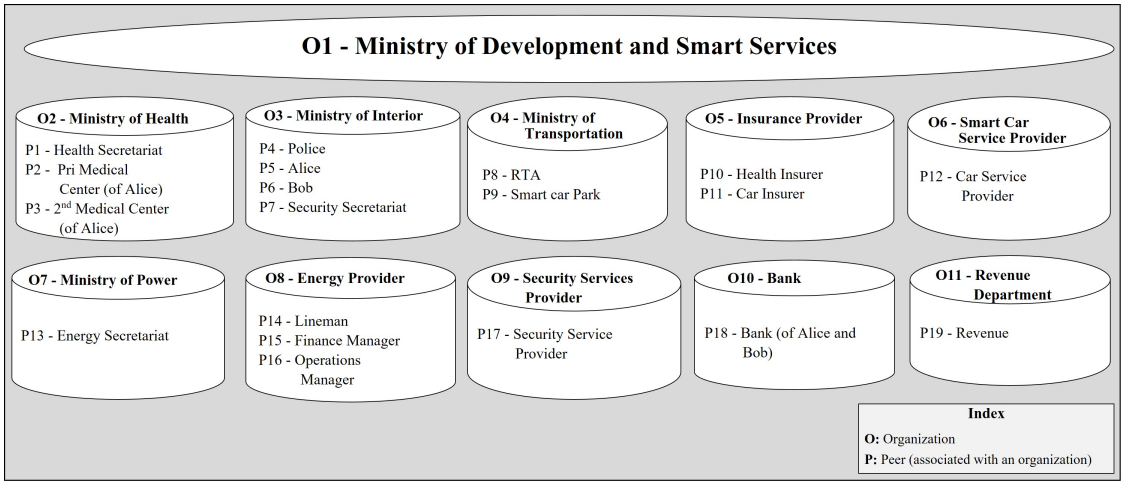

Figure 5.2: Network participants

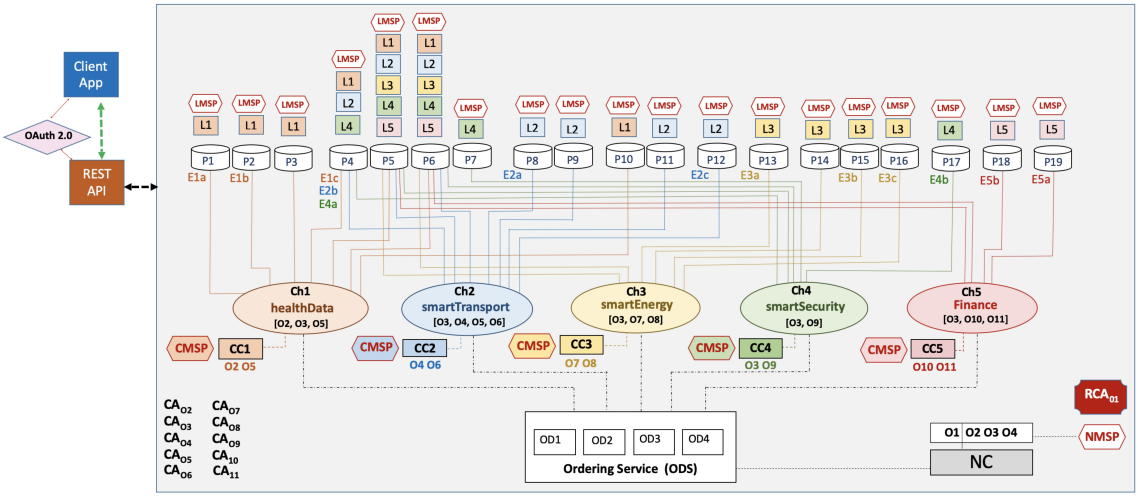

Figure 5.3: Smart city blockchain-network architecture

5.2.3 Network Architecture

As shown in Figure-5.2, the smart city blockchain network comprises eleven organizations (O) and their associated peer nodes. Keeping in view the sharing of different categories of users' data with different stakeholders (shown in Table-5.2) and the requirement to ensure user data privacy and security, the blockchain network shown in Figure-5.3 comprises five different data Chs. Where Ch1 is used for the sharing of users' health data and O2, O3, and O5 are its members. Similarly, Ch2 is for smart transportation data, and it comprises O3, O4, O5, and O6. Whereas Ch3 is for smart energy, Ch4 for smart security and Ch5 handles financial data (e.g., income, expenses and taxes). A Ch provides a completely separate communication mechanism between a set of Os. Moreover, every Ch is independent of the other Chs. Hence, these Chs serve to preserve the privacy of user data by securely sharing a particular type of data with authorized entities only. The network is initiated by O1, i.e., the Ministry of Development and Smart Services and is governed by the policy rules specified in the network configuration (NC). NC also controls access to the smart city network. Later, O1 updates NC and gives administrative (admin) rights to O2, O3, and O4 as well. These Os can now create consortia and Chs to add more network members. Similarly, every Ch is regulated by the policy rules specified in the respective Channel Configuration (CC). In this setting, Ch1 is under the control of O2 and O5 and is governed by CC1. Correspondingly, Ch2 is regulated by CC2, and so on.

The CC is essential for Ch security, e.g., if the client application (clientApp) wants to access a SC on P1, then P1 consults its copy of CC1 to determine the operations that clientApp can perform. Moreover, there is a separate ledger for every Ch, and all the peer nodes have to maintain a copy of the ledger concerning every Ch, in which they are participating. Therefore, if a peer, say P4, is a member of three different Chs, then it has to maintain three ledgers. Data in a Ch is isolated from the rest of the network including other Chs. Another important aspect of the smart city blockchain network is the ODS, which is common to all the Chs. In this setup, the

ODS has four ordering nodes, one each from O1, O2, O3, and O4. Each node in the ODS keeps a record of every Ch created through NC. Regarding CAs, every organization in the network can have its own CA. But there is one Root CA (RCA) in the network to establish the root of trust. As a PoC for PrivySharing, we are using Hyperledger Fabric RCA to issue X.509 certificates to all the network entities. These certificates serve to authenticate the network entities and to digitally sign the client application TX proposals and smart contract TX responses. A user accesses the network through a clientApp with a specific X.509 ID, using a SC. It is imperative to mention that only the endorsing peers can see the SC logic as they have to run the users' TX proposals to prepare the responses.

To ensure the privacy of critical user data within a Ch, i.e., keeping part of user data private from some organizations within a Ch, we adopted a methodology of "Private Data Collection," in which the critical private data is sent directly to the authorized organizations/stakeholders only. This data is stored in a private database (a.k.a sideDB) on the authorized nodes. While private information is stored on the authorized nodes, only the hash of this data is processed, i.e., endorsed, ordered, and written to the ledgers of every peer on the Ch. The hash of the data serves as evidence of the TX, and it also helps in the validation of the world state. A vital data security feature here is that the ordering nodes do not see the private data. However, to further increase the level of data privacy/confidentiality, the user has the option to encrypt his private data such that not even the peers/nodes authorized to view data stored in the private data collection can see the original contents. The data is encrypted using AES-256 bit symmetric encryption key and then stored in the private data collection. Later on, only the authorized users who have access to the decryption key can query the user's private data. Supplementary to the data encryption, there is an additional feature of signed encryption of private data for an increased level of user authentication and data security.

Another important feature of our proposed network architecture is the use of MSP at various levels, such as network, Ch, and local/peer. The network MSP (NMSP) defines, who all are the members of the network and who out of them have the admin rights. Additionally, an NMSP also defines that which RCAs/CAs are trusted. On the other hand, the Ch MSPs (CMSP) outline admin and participatory rights at the Ch level. All the peers and the ODS share a common CMSP to correctly authenticate and verify the authorizations of the Ch members. A use case for the CMSP is that, e.g., an admin of an organization wants to instantiate a SC on Ch1, then by looking at the CMSP, the other Ch members can verify that whether that admin is a part of a specific organization or not and whether he is authorized to instantiate the SC on Ch1 or not.

Similarly, a local MSP (LMSP) is defined for every client-node/peer. The LMSP associates a peer with its organization. It also defines the permissions for that peer and allows it to authenticate itself in its TXs on the Ch. Here a question may arise that, what is the difference between CC and a CMSP? A CC contains the policies that govern that Ch, i.e., which organizations can regulate the Ch and add new members. Whereas, a CMSP establishes the linkage between the nodes and their respective organizations, and what roles a node can play within a Ch, i.e., can it instantiate a SC on a Ch? Concerning decentralization aspect; the use of a

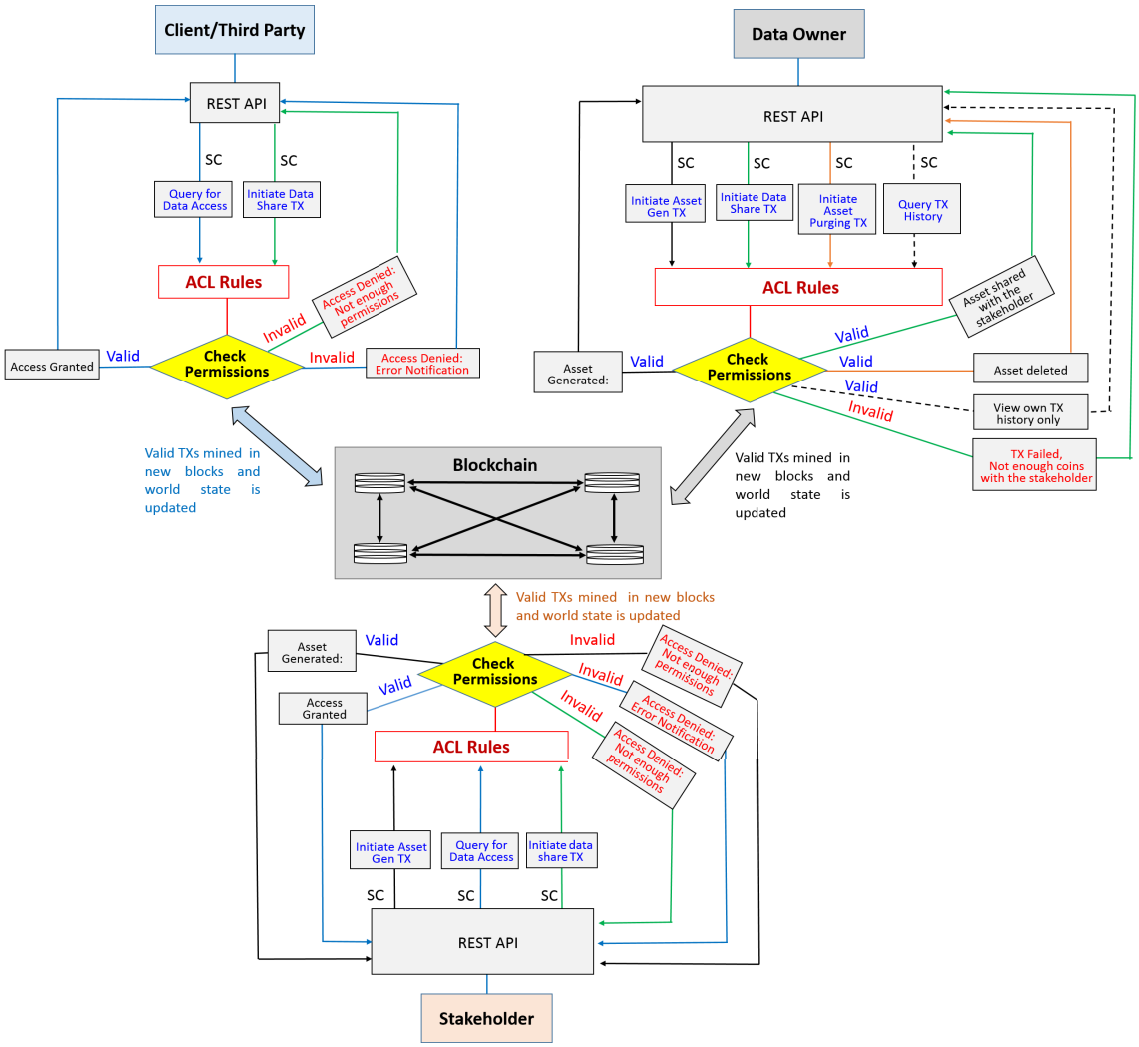

Figure 5.4: Smart Contract TXs

dedicated trusted CA, a blockchain admin, and a business network admin by every organization in the blockchain network provides some degree of decentralization as compared to all the admin rights resting with a single organization.

Another question may arise that what advantages do we get by using multiple Chs for different data types as compared to a single Ch blockchain network to share all the types of data. There are two aspects to this selection; one is scalability, and the second is the increased privacy of user data. From the scalability point of view, if there is only one Ch for all types of data, then it means that the users will have to store the ledger comprising all those TXs that are not even related to them. Hence, the ledger size will increase rapidly, thus putting more strain on the storage resources of all the users/peers. Whereas, in the case of "PrivySharing," the users will maintain a ledger that stores only that data which concerns all the users of that particular Ch. Moreover, the experimental results (Section-5.4) have validated that the multi-Ch blockchain network scales well as compared to a single-Ch blockchain. As far as the privacy of user data is

concerned, a data specific Ch shared only by some of the stakeholders provides more privacy than a single Ch comprising all the stakeholders sharing multiple data types. Although, use of multiple data specific Chs seems scalable as compared to a single Ch, yet the requirement for users to maintain a ledger each for every Ch, in which they participate, may still crave for ample storage resources.

PrivySharing framework has been designed, developed, and tested based on the agile blockchain application development guidelines proposed by [388]. The said guidelines helped in a systematic design, development, and testing of PrivySharing network architecture, SC functionality, and efficacy of ACL rules. Moreover, influenced by these guidelines, Figure-5.4 highlights different TXs initiated by various actors operating in the smart city network. Every TX and its associated decision/response based on ACL rules are depicted by the same colored line. E.g., a client/third party can only query for some user data asset. If it is authorized to access the data, the query will be successful. Otherwise, there will be an access denied error message. Both the query and respective responses are shown by blue lines. Similarly, the data share TX is sketched in green color. As per the PrivySharing business model, the client/third party should not be allowed to submit a data sharing TX; hence, if a client still initiates a TX to share data assets of some user, then he gets a "access denied: not enough permissions" error message. TXs concerning data owners and stakeholders have also been projected accordingly.

5.2.4 Smart City Blockchain - Plain TX Flow

There are two types of TXs; one is plain TX that can be viewed by all the Ch members, and the other one is private data TX that is to be shared only with some selected peers in a Ch. In this regard, e.g., a plain TX that is required to update Alice's car's current location state on Ch2 is initiated by the ClientAppA installed in Alice's smart car. This TX (as shown in Figure-5.5a) is processed in the following steps:

a. **Step-1**. ClientAppA invokes the SC_A and sends a TX proposal containing the current location of Alice's car to the pre-defined endorsers as per SC_A endorsement policy on Ch2. In this case, the endorsers are E2a (RTA), E2b (Police), and E2c (Car Service Provider). A TX will be approved if it is endorsed by a minimum two out of the three prescribed endorsers.
b. **Step-2**
   2.1. Three endorsers E2a, E2b, and E2c, invoke SC_A with the proposal.
   2.2. SC_A generates a query or update proposal response. The endorsers, E2a and E2b, endorse the proposal for correctness.
c. **Step-3**. E2a and E2b both send a signed (endorsed) TX proposal response along with the Read-Write (RW) set back to the ClientAppA. At this stage, the endorsing peers do not apply the proposed update to their copy of the ledger.
d. **Step-4**. ClientAppA verifies that the response received from at least two endorsers is the same, i.e., deterministic. However, there is a possibility that the results were generated at different times on different peers with ledgers at different states. Hence, the peers can return different TX

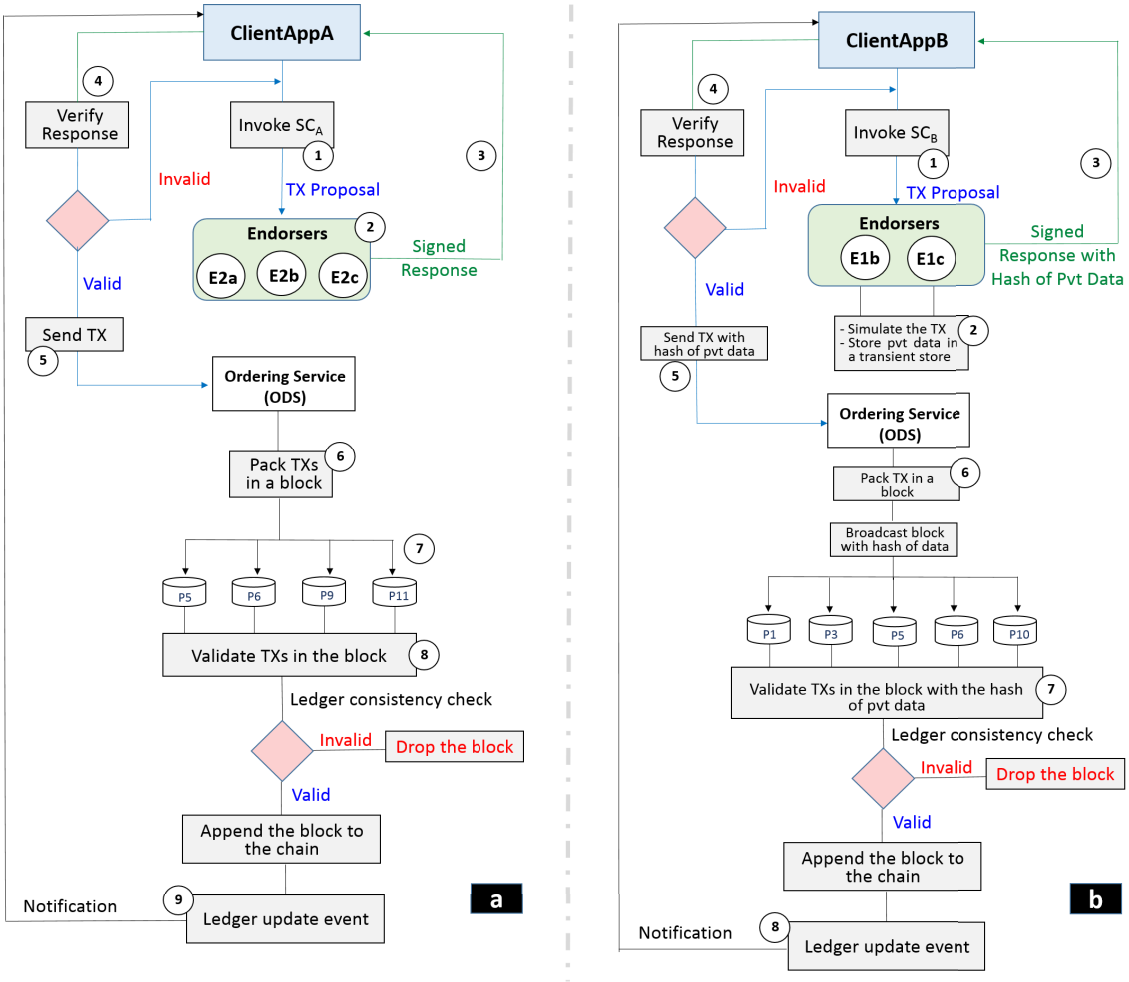

Figure 5.5: a) Plain TX flow, and b) Private data TX flow

responses for the same TX proposal. In this case, an application can request a more up-to-date proposal response. Another less likely possibility is that the SC might be non-deterministic, e.g., while getting forex (foreign exchange rates) data from some websites, the TX responses can be different, as forex rates may differ at different times. Therefore, inconsistent results cannot be accepted by the application and applied to the ledger.

e. **Step-5**. Once the ClientAppA verifies the endorsers' responses, it sends the TX to the ODS.

f. **Step-6**. ODS then groups the received TXs in a block. The sequence of TXs in a block is not necessarily the same as the order of arrival of the TXs at the ODS. However, the generated blocks are final, and there are no forks. Moreover, the orderers do not host the ledger and the SCs, and they are also not concerned about the value of the TX; rather, they just package the TXs into the blocks.

g. **Step-7**. ODS broadcasts the next proposed block to all the peers on the Ch2.

h. **Step-8**. All the committing peers validate every TX in a block (in the same sequence as they appear in the block) to ensure that it is correctly endorsed by all the required endorsers before it is applied to the ledger. Once a TX is verified correctly, the peers perform a ledger consistency

check to establish that the current state of the ledger is compatible with the state of the ledger when the proposed update was generated. World state is updated based on the validated TXs. It is to be noted that the failed TXs are not applied to the ledger, but they are retained for audit purposes. Moreover, TX validation in Step-8 does not require the running of SCs. This is done only by the endorsers. Hence, SCs are installed only on the endorsers. This keeps the logic of the SCs confidential to the endorsing organizations only. Moreover, peers also mark each TX in each block as valid or invalid. Finally, a new block is appended to the hash chain stored in the ledger L2, maintained by all the peers in their file system.

i. **Step-9**. Ledger update event is generated, and the ClientAppA is notified.

It is important to note that before appending a block, a version check is performed to ensure that the states being updated are the same that were read during SC execution. It protects against double-spending and other data integrity threats. The above mentioned TX workflow mediated by the orderers is called "Consensus," as all the peers reach on an agreement about the content and the order of the TXs.

5.2.5 Smart City Blockchain - Private Data TX Flow

As per smart city network settings shown in Figure-5.3, if a wearable blood alcohol monitoring device on Alice generates an alert to be seen only by her Pri Medical Center and the local police for immediate response. In such a case, it is required to keep such a TX private which should not be seen by other members on Ch1 except P2, P4, and P5. Such a private data TX (as shown in Figure-5.5b) is processed in the following steps:

a. **Step-1**. The clientAppB submits a proposal request to invoke a SC function (RW private data) to the endorsing peers E1b (Pri Medical Center) and E1c (Police), which are part of the authorized organizations of the collection (defined by the private data dissemination policy on health alert). The private data concerning health alert on blood alcohol level is sent in a transient field of the proposal.
b. **Step-2**. E1b and E1c simulate the TX and store the private data in a transient data store (temporary storage local to them). The endorsing nodes also distribute the private data based on the collection policy to authorized peers via gossip. But in this case, we only have three peers, i.e., P2(E1b), P4(E1c), and P5.
c. **Step-3**. E1b and E1c send the proposal response back to the clientAppB with public data, including a hash of the private data key and value (Blood alcohol level). No private data is sent back to the clientAppB in plaintext.
d. **Step-4**. The clientAppB verifies that the RW sets received from E1b and E1c are the same.
e. **Step-5**. The clientAppB submits the TX with a hash of the private data to the ODS.
f. **Step-6**. The ODS packs the TX in the latest block. The block with the hashed value is distributed to all the peers on Ch1.
g. **Step-7**. All the peers on the channel validate TX with the hash of the private data in a consistent way, without knowing the actual private data.

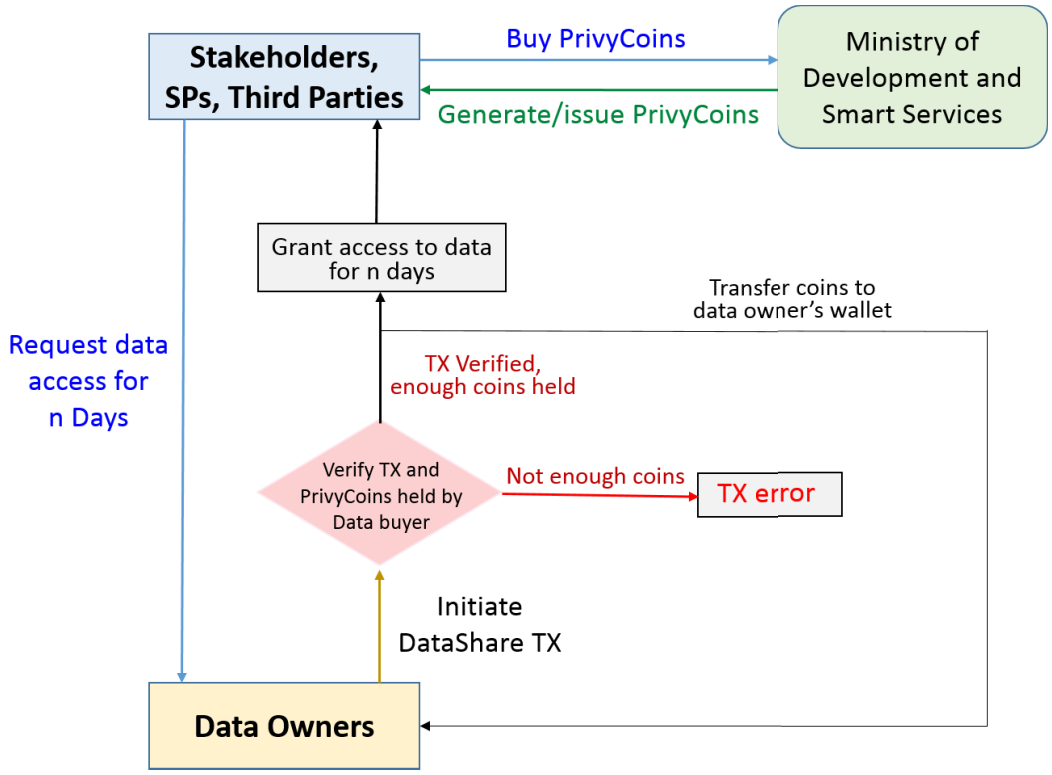

Figure 5.6: Reward mechanism based on PrivyCoins

h. **Step-8**. Ledger update event is generated, and the clientAppB is notified.

5.2.6 Reward Mechanism

PrivySharing incentivizes the users to share their data with other users, stakeholders, service providers (SPs), or third parties by rewarding them with a local digital token named "PrivyCoin," as exhibited in Figure-5.6. PrivyCoin is just like an asset in the smart city network that is issued only by the network admin (Ministry of Development and Smart Services) against the payment in terms of fiat currency. The secure execution of such a TX is not covered in this work. However, it is envisaged that the stakeholders can pay the ministry through any secure payment app and then receive the coins in their wallet, just like any other cryptocurrency/token. PrivyCoin is primarily used for trading or getting access to the data assets. After acquiring PrivyCoins, the stakeholder forwards the request for data access along with asset ID and the duration of access (in terms of days). Currently, in PrivySharing, the third parties/stakeholders pay one PrivyCoin to a user to get access to a data asset for one day (24 hours). Hence, if a stakeholder wants to get access to two data assets of a user for five days, he has to pay ten PrivyCoins to the user. Upon receiving the request to share data, it is only the prerogative of the data owner to initiate the data sharing TX. The data owner gets the incentive as soon as the data sharing TX is committed. In this context, if a stakeholder does not have requisite coins in his account, the TX will fail (shown in Figure-5.7). The pseudocode for the reward-based data sharing TX is illustrated in Algorithm-1.

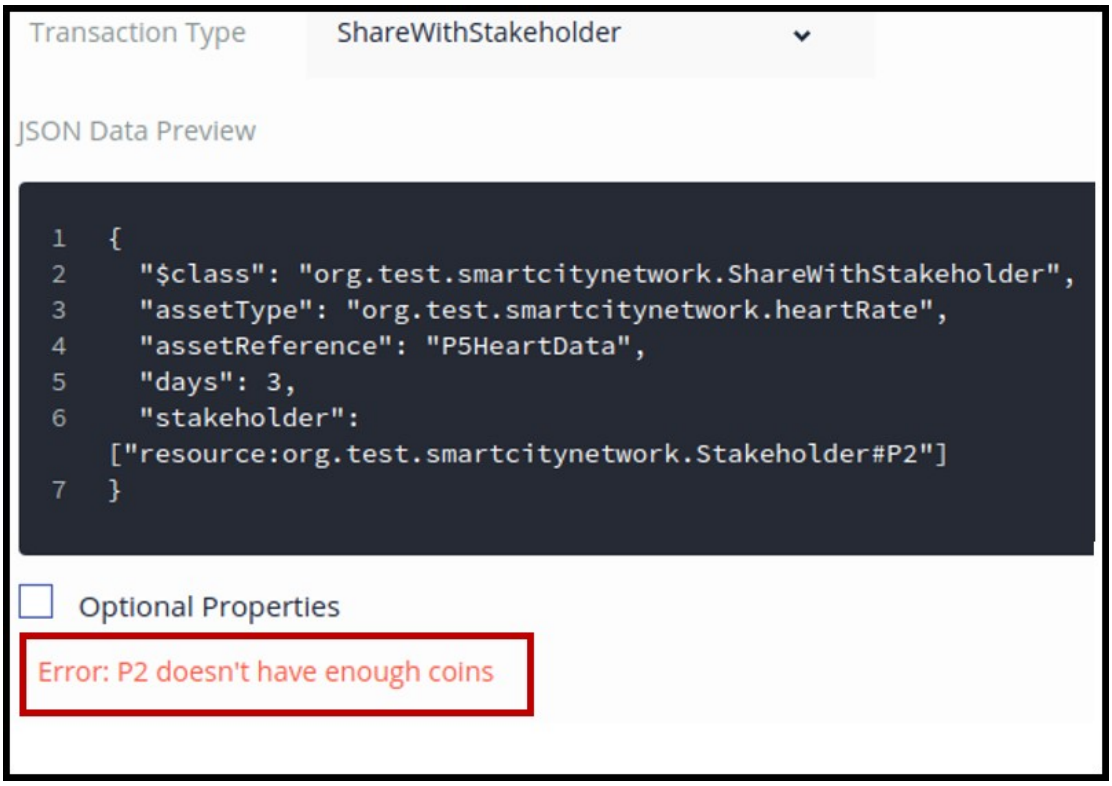

Figure 5.7: Error for not having enough coins

This algorithm can be summarized into four steps. Firstly, the data asset to be shared is obtained from the asset registry. Whereas, the input data structure of the data sharing *TX* contains the asset type (e.g., Heart Rate, Blood Sugar, etc.), the asset reference (ID of the asset), the time duration of sharing (e.g., three days) and a list of stakeholders (e.g., P2, P4). Then, the algorithm checks whether the asset has already been shared with the stakeholders or not. After that, stakeholders pay PrivyCoins to the data owner. Finally, the asset status is updated, and an event is emitted to notify the related parties, i.e., the user and the stakeholders.

5.3 Security Analysis

The security, being the core objective of this work, has been assessed at every level of the network operation. The key aspects shown in Figure-5.8 are illustrated as under.

When the blockchain network is first created, all the peers and orderer organizations are issued with certificates from respective RCA, or other trusted CAs. Then, a connection profile is created for all the network entities, including Chs, ODS, organizations, peers, and CAs. The connection profile defines the complete blockchain network setup. E.g., for a Ch, it defines the Ch name, its associated ODS and peers. It also defines which peers are the endorsing peers for that particular Ch. For an organization, it defines the namespace, MSP ID, member peers, and the respective CA. The peers' profile includes the namespace, URL including the port number, and the TLS certificate for its principal organization. The key point here is that no other peer (with the intention of

Algorithm 1 Reward-based Data Sharing with the Stakeholders

Input: ShareWithStakeholder(tx)
  $asset \leftarrow assetRegistry.get(tx.assetReference)$ {**STEP-1:** Retrieving the asset from asset registry}
  {**STEP-2:** Check, whether an asset is already shared with the stakeholder or not}
  for all $stakeholder$ In $tx.stakeholders$ **do**
    if $asset.stakeholdersWithAccess$ is not Empty **then**
      $stakeholderId \leftarrow stakeholder.operatorId$
      if $stakeholderId$ exists in $asset.stakeholdersWithAccess$ **then**
        MESSAGE: Data already shared.
        Jump to the next $stakeholder$
      else
        push $stakeholderId$ into $asset.stakeholdersWithAccess$
      end if
    else
      $asset.stakeholdersWithAccess \leftarrow [stakeholderId]$
    end if
    {**STEP-3:** Stakeholders pay coins to the asset owner}
    $coins \leftarrow Coinsbelongtostakeholder$
    if $coins.length < tx.days$ **then**
      return ERROR: $stakeholder$ does not have enough coins
    else
      for $j = 0$ to $tx.days - 1$ **do**
        $coins[j].owner \leftarrow asset.owner$
        Update coin status
      end for
    end if
    {**STEP-4:** Event generation}
    Emit event of sharing
  end for
  Update $asset$ status
  return Sharing Success

endorsing the TXs on a Ch) can join the network if it is not defined in the connection profile. It is clarified that by peers, we mean committing, endorsing or ODS peer nodes that maintain the blockchain network. Whereas, the users/clients access the blockchain network through REST API or clientApps. The smart city blockchain network entities including ODS, peers, CAs, ledgers, and SCs, run in separate docker containers (symbolize by blue boxes numbered from D1 to D16 in Figure-5.8). This separation minimizes the effects of a container compromise, i.e., if one container's security is breached the other containers remain unaffected.

To deploy the business network model (PrivySharing in this case) that comprises asset definitions, TX and event logic, and ACL rules on the blockchain, the admin of responsible organization (O1 in this scenario) requires a Business Network Card (BNC). The BNC is created using the connection profile of the organization and the valid public and private key for that admin issued by the authorized CA, as defined in the connection profile. The TXs initiated by the clientApps on a specific Ch are endorsed as per the endorsement policy defined for the respective Ch before the start of the business network. The endorsement policy may include, e.g., what all peers (with endorsing ability) are required to endorse a TX on a Ch concerning health data. Similarly, a TX is considered valid only if the response of all the required endorsing peers is the same. Hence, only a valid TX will update the world state. Another vital security feature of PrivySharing is that before the start of the business network on the blockchain, business network admins have to be defined and issued with the certificates (Public and Private key pairs) by the respective CAs. These certificates are later used to create the BNCs for the said admins to access the business network. Without a valid BNC, no one can add participants (clients/peers) for an organization. Moreover,

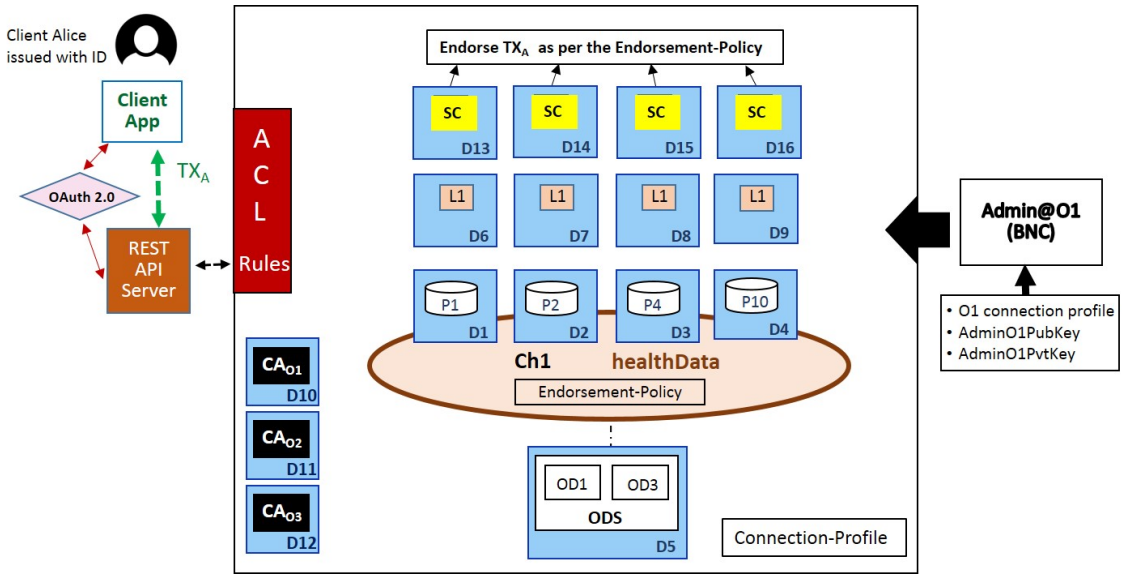

Figure 5.8: Elements of PrivySharing network security

every new client/peer added under an organization is also issued with an ID by the respective CA with the approval of the business network admin. These IDs are further used to control access to the users' profile and assets as per the ACL rules defined for the specific Ch.

As far as privacy of user data is concerned, the use of data specific Chs, private data collection, and data encryption does provide some degree of data privacy. However, even if a user's IoT device data is encrypted, still a passive network attacker can infer a pattern of user's activities. The same has been demonstrated by the researchers in [389]. The authors exhibited that an adversary capable of monitoring the network traffic between a smart home gateway device and the internet can determine the type of IoT devices being used inside a smart home, based on DNS queries. Also, the attacker can analyze the metadata of the network traffic and observe variations in the IoT data send/receive rates. Hence, based on these abrupt changes in data rate/packet size, the adversary can deduce vital information about user's behavior and daily routine. Although, the conventional IoT classification methods do not apply to the blockchain, as the TXs in blockchain contain public keys instead of IP addresses, and are broadcast to the network. Nevertheless, to avert the effects of malicious network traffic monitoring measures such as the incorporation of VPN tunneling or obfuscating and shaping all smart home network traffic can be taken to mask variations that encode real-world behavior of the device owner.

Correspondingly, in blockchain-based IoT systems, the combination of device classification and user deanonymization can infer private information about a user to an adversary. Although, in PrivySharing, the IDs of all the members of the network are known, and there is also a provision that each user can be issued with multiple cryptographic IDs (Public-Private key pairs) [390]. Hence, users can use a different ID to communicate with every stakeholder. Such an arrangement seems robust against linking attacks [391]. However, blockchain researchers in [392] established the possibility of IoT devices classification by analyzing IoT device data stored on the blockchain

by applying Machine Learning (ML) algorithms. Unlike in [389], an adversary is assumed to have access only to the data stored on the blockchain rather than the network traffic [392]. The attack methodology identifies the IoT devices based on different patterns of timestamp differences in successive TXs of each type of device. However, researchers also proposed combinations of various methods of timestamp obfuscation to avoid device classification. These techniques include: introducing a random delay in the TXs of a device, combining multiple data packets of a specific device into a single TX, and lastly, merging ledgers of numerous devices.

5.3.1 ACL Rules

PrivySharing has embedded user-defined ACL rules in the data sharing chaincodes to protect user data. The graphical illustration of the access control process based on some of the ACL rules is shown in Figure-5.9. These rules enforce that the data asset owners have access to their assets only, i.e., no user can see data assets of any other user, and only the data owners can initiate a TX to share their data assets with other users/stakeholders. Similarly, a data owner has the right to revoke the sharing of his assets, and he can also delete his assets when no longer required without affecting the TX history stored on the blockchain. Moreover, as all the TXs are recorded on the blockchain, hence, to increase privacy, a data owner can see the TX history concerning his assets only. Additionally, valid users can read and update their profiles only, and other users/stakeholders cannot see each other's profile. Users can also delegate the stakeholders to create assets on their behalf. E.g., Alice (P5) delegates her Pri Medical Center (P2) to create a health data asset for her. Accordingly, the stakeholders can only see the data assets that are shared with them or created by them. Lastly, all the users/stakeholders can view their coins only. The pseudocode of the data asset unsharing and asset deletion is accordingly shown as Algorithm-2, and Algorithm-3, respectively.

Algorithm 2 Unsharing Data Assets with the Stakeholders

Input: UnshareWithStakeholder(tx)
  $asset \leftarrow assetRegistry.get(tx.assetReference)$ {COMMENT: Retrieving the asset from asset registry}
  {COMMENT: Removing the stakeholders}
  for all $stakeholder$ In $tx.stakeholders$ **do**
    if $asset.stakeholdersWithAccess$ is not Empty **then**
      $stakeholderId \leftarrow stakeholder.operatorId$
      if $stakeholderId$ exists in $asset.stakeholdersWithAccess$ **then**
        Remove $stakeholder$ from $asset.stakeholdersWithAccess$
      else
        MESSAGE: $Asset$ is not shared with the $stakeholder$
      end if
    else
      MESSAGE: $Stakeholder$ has no access to any record.
    end if
    {COMMENT: Emitting an event of unsharing asset}
    Emit event of unsharing
  end for
  Update $asset$ status
  return Unsharing Success

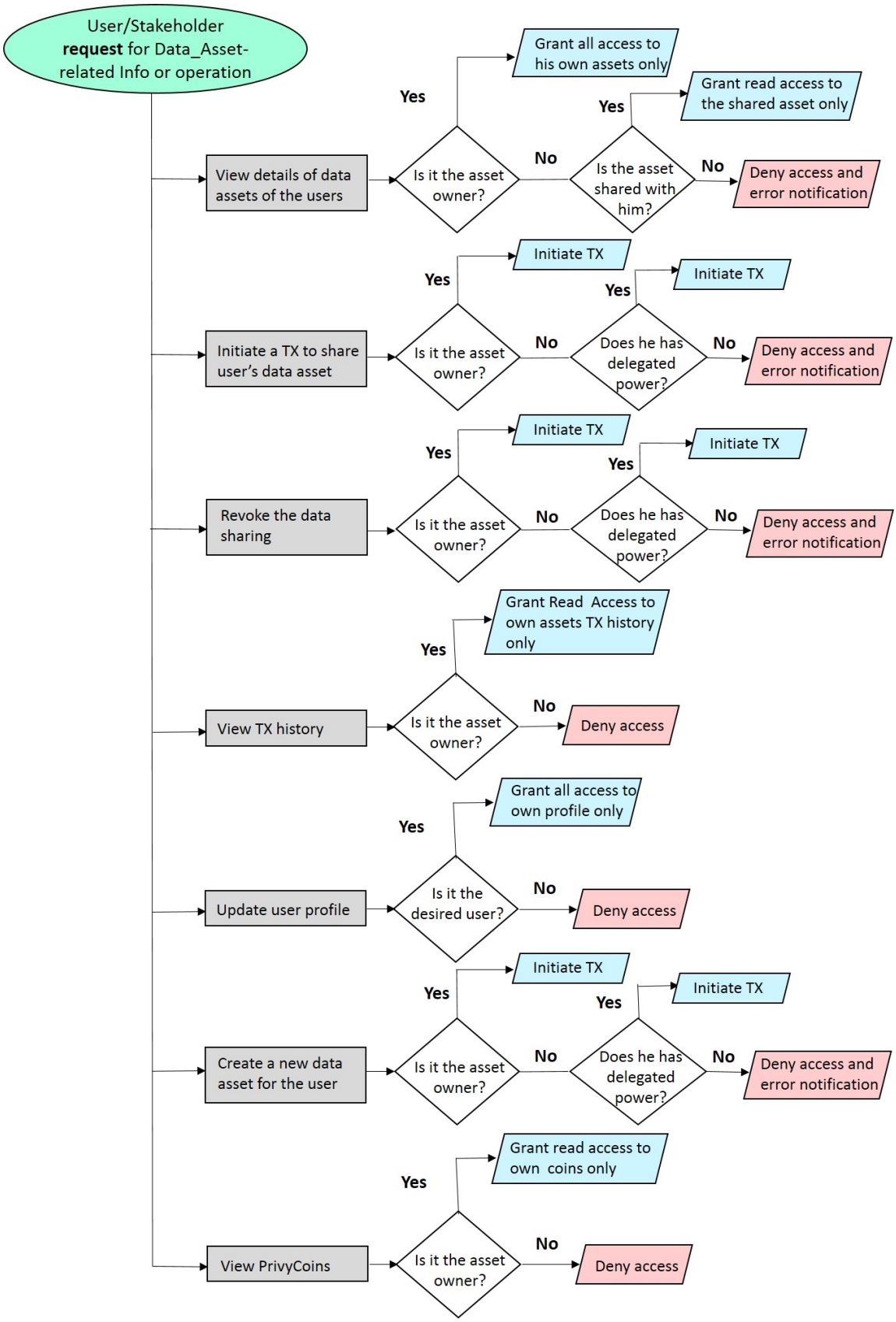

Figure 5.9: ACL rules

Algorithm 3 Deleting a Data Asset

Input: DeleteAsset(tx)
$asset \leftarrow assetRegistry.get(tx.assetReference)$ {COMMENT: Retrieving the asset from asset registry}
{COMMENT: Removing the asset from asset registry}
Delete $asset$
{COMMENT: Emitting an event of Deleting}
Emit event of asset deletion
return Deleting Success

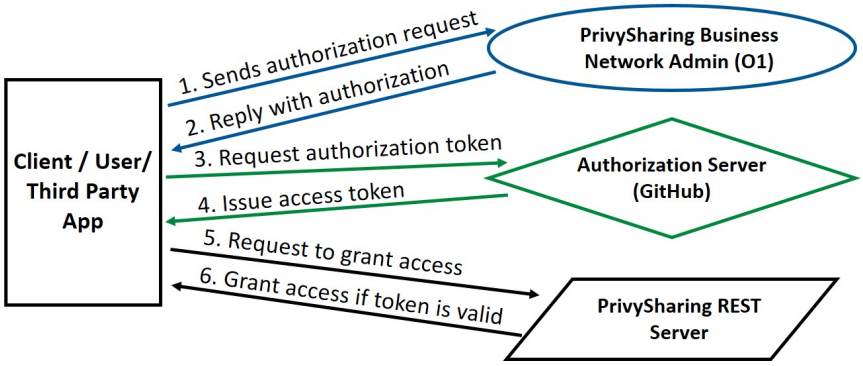

Figure 5.10: PrivySharing REST server OAuth protocol

5.3.2 Security of REST API and DApp

Access to the REST API is secured using the API key, which is required to launch the REST API. In addition to the API Key, the OAuth-2.0 authorization protocol [393] is also employed to authorize access to the PrivySharing REST server instance, and allow the end-users/clients to interact with the PrivySharing business network deployed on the blockchain. The mechanism of the OAuth-based REST API security protocol is shown in Figure-5.10. In Step-1, the client/user/third-party App sends an authorization request to the PrivySharing business network admin from O1 that also acts as the resource owner. The resource owner then replies with the authorization grant in Step-2. In Step-3, the client sends an authorization token request containing the authorization grant received from the resource owner in Step-2 to the authorization server. After validating the authorization grant, the authorization server issues an access token to the client in Step-4. The client then requests the PrivySharing REST Server in Step-5 to grant access by presenting the access token. Finally, in Step-6, if the token is valid, the client is granted access to call the PrivySharing REST API operations. Currently, there are more than three hundred options for the client REST Server authentication strategies including SAML, LDAP, GitHub and a blend of OSN such as Facebook and Google. For this PoC, we have used the Passport-GitHub strategy to authenticate the users. The detailed procedure of enabling OAuth for PrivySharing REST Server is depicted in Figure-5.11.

Furthermore, due to the distributed nature of the SCs, the integrity of any business network deployed on the blockchain is guaranteed. Similarly, it also protects against hacking of servers, where the attackers can change the policy rules, escalate access rights, etc. Correspondingly, protection against application and web vulnerabilities can also be guaranteed with high probability, as

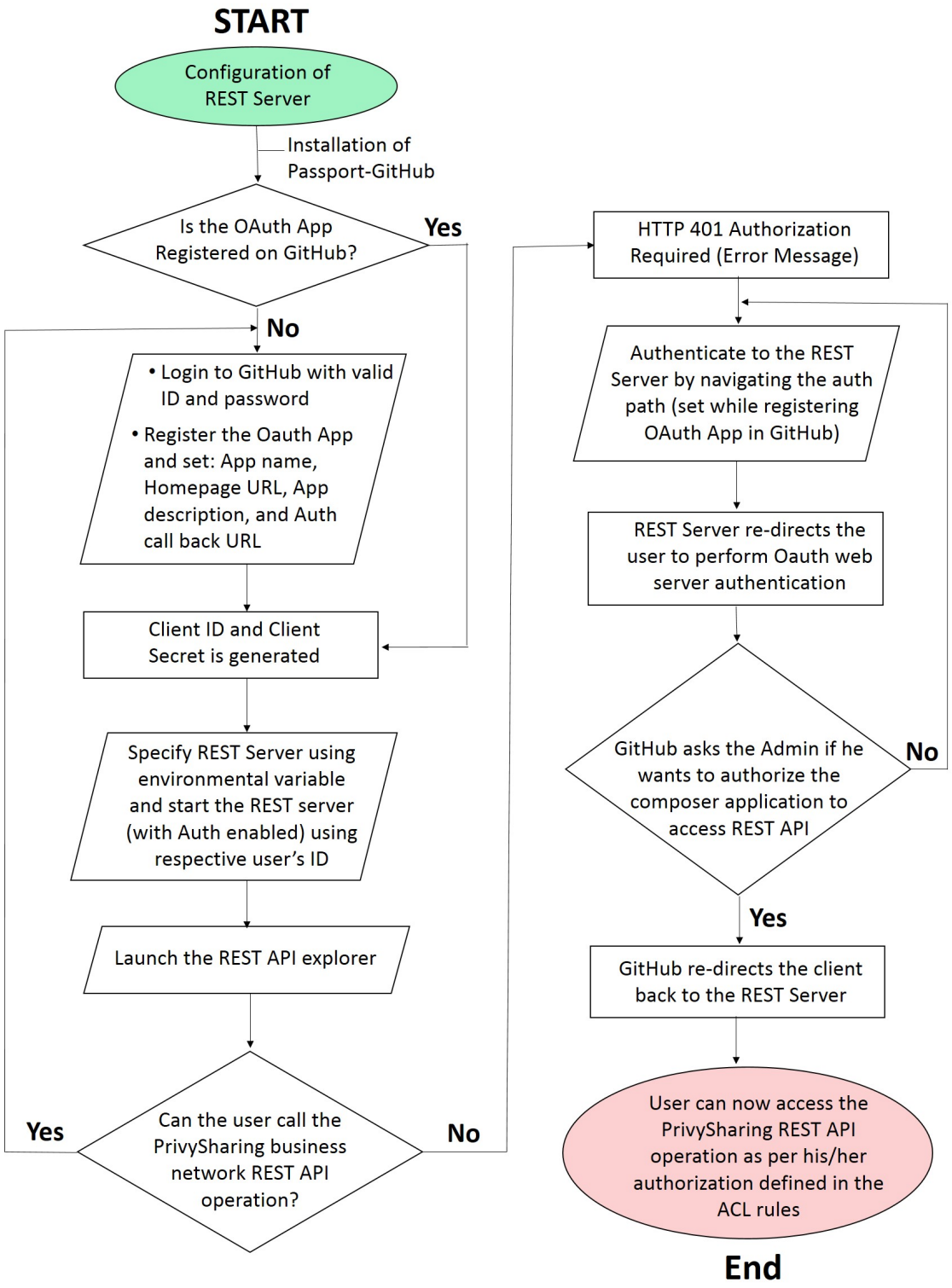

Figure 5.11: PrivySharing REST server OAuth flowchart

```
root@7efa652b877c:/opt/gopath/src/github.com/hyperledger/fabric/peer#  peer chaincode query -C healthdata -n sacc -c '{"Args":["get","HeartRate"]}'
Error: error endorsing query: rpc error: code = Unknown desc = access denied: channel [healthdata] creator org [Org7MSP] - proposal response: <nil>
```

Figure 5.12: Access denied for out-of-Ch data query

any change in the smart contract requires installing and instantiating a new version of the contract on all the endorsing peers. However, it cannot be done discretely. Additionally, due to a distinction between blockchain and the world state, an auditable log of TXs and events is maintained without compromising the privacy of the users' data.

5.3.3 Restricted Access to User Data Assets via Multiple Chs

In addition to restricting access to users' data assets through ACL rules within a Ch, the use of data specific Chs is also helpful in preserving users' data privacy. Through our PoC, we have validated that every Ch in PrivySharing smart city network is independent of other Chs with associated Ch members. As shown in Figure-5.12, when P13 from O7 (not a member of Ch1), tries to query a user's heartRate data, he gets an access denied error because he is not authorized to access any data asset propagated on Ch1. As PrivySharing is a permissioned consortium blockchain, all the network members are duly registered and authenticated before joining the network. However, even if an unauthorized node gets added to the system through a corrupt network admin, the ACL rules prohibit the intruder from unauthorized access to users' data assets.

Moreover, Table-5.3 shows the methodology we adopted to achieve the security objectives derived from the smart city threat environment and EU GDPR requirements. However, one of these objectives, i.e., IoT device integrity check, has not been addressed in this research.

Table 5.3: Methodology to achieve PrivySharing objectives

Ser	Factors Deriving the Objectives	Objectives	Methodology
Threats to User/Data Security in a Smart City Environment			
1.	User privacy (ID disclosure)	Reduce the possibility of users' real-world ID disclosure	PKI (X.509 Certificates) based multiple IDs for users
2.	User data privacy	Data confidentiality at rest and in transit, prevent over data collection, controlled access to data as defined by the data owner	Data encryption, use of SSL/TLS for data security in transit, user-defined ACL rules, use of multiple Chs and private data collection within a Ch
3.	Single point of failure (from physical as well as trust point of view)	Distributed data storage and decentralized control	Using Hyperledger Fabric Blockchain

Continued on next page

Table 5.3 – *Continued from the previous page*

Ser	Factors Deriving the Objectives	Objectives	Methodology
4.	False injection of data	Prevent data injection by unauthorized users	ID management, authentication, and participation of only authorized nodes in the network. Moreover, TX initiation rights given to data owners or the parties given delegated powers by the data owners
5.	Vulnerability to Sybil Attack	Prevent Sybil Attack	User ID management and TX initiation by authorized entities only as per ACL rules
6.	Lack of common security framework for heterogeneous IoT devices with different communication protocols and diverse hardware parameters	Provide a common platform to store data transmitted/received from the heterogeneous sensors, irrespective of their diverse hardware and communication technologies	Use of Hyperledger Fabric Blockchain
7.	Threats to data integrity (data forgery and manipulation)	Preserve user data integrity	User authentication and restricted privileges to update user data, and blockchain's inherent data integrity protection
8.	Threats to smart city applications	Protect applications against the escalation of privileges and alteration attacks	As any change in the code of a smart contract or ACL rules, requires the deployment of a new version of the smart contract on the blockchain with network consensus. Hence, there cannot be any malicious alteration in the smart contract based DApps without detection.
9.	Scalability	Contain the size of the blockchain	Use of blockchain to store TX logs only, whereas a world state is used to store updated state of user data
10.	TX Latency and Throughput	More TX throughput with less latency	Use of multi-Ch blockchain as compared to a single-Ch blockchain
Essential GDPR Requirements for User Data Security			
1.	Personal data to be processed only with data owner's consent	The data owner is in complete control of his data, transparency of the complete process, visibility of all security and data access control changes	Chaincode-based user data access control rules, maintaining, and disseminating TX log on the need to know basis (Only a data owner or an authorized entity can see the TX log of a specific asset) and data sharing TX can only be initiated by the data owner
2.	Privacy by design	By default user data should be inaccessible to all, except those who are specifically allowed by the data owner	Access control rules deny everyone to see other's profile and assets unless explicitly shared by the data owner
3.	Commissioned data processing (i.e., data collection and processing as per the contract between the data owner and other parties)	A contract-based user data sharing that should conform to the contractual obligations	Business logic is transformed into Smart Contracts for secure and efficient data sharing as per contractual obligations

Continued on next page

Table 5.3 – *Continued from the previous page*

Ser	Factors Deriving the Objectives	Objectives	Methodology
4.	Data owner should have access to all the information concerning his data (i.e., where is it stored, who has access to it, and for how long)	A transparent system, where data owner has complete visibility of the process and should be able to see and control the access to his data	User-defined data access control, and TX log management
5.	Right to forget, i.e., user data to be erased when no longer required	The system should allow user data deletion after a specific time, when the contract between the user and a third party expires, or when data is outdated or no longer required. Hence, there should be some distinction between TX log maintenance and user data storage. Such that even if user data is deleted, we are still able to verify the integrity of the past data	The world state is distinct from the blockchain. Hence, data/asset owner can delete user data from the world state without affecting TX log history
6.	Transparency	The system should be transparent, i.e., log all the activities concerning users' data (when and who modified the access control policies for data and updated the data itself)	TX log management and event notification

5.4 Experimental Results

To validate the security effectiveness and measure the performance efficiency of the proposed solution, we designed, developed, and set up a three-Ch smart city data sharing scenario for the sharing of health, smart energy, and financial data. The experimental setting, as shown in Figure-5.13, comprises six organizations and twelve peers. However, for a production environment, the minimum nodes required to establish a blockchain network primarily depends upon the type of consensus protocol being used for ordering service. Moreover, other contributing factors may include the type of blockchain application and the degree of decentralization required. Hence, there may be multiple Chs, more than two organizations with their peers and CAs, and numerous stakeholders participating in the ODS. Currently, Kafka is the recommended consensus protocol for the production environment. Moreover, Kafka-based ordering service is a combination of a Kafka cluster and a Zookeeper ensemble. To establish a Kafka cluster and Zookeeper ensemble, there should be a set of a minimum of four Kafka and three Zookeeper nodes to achieve fault tolerance. As a PoC, we deployed the business network model of PrivySharing on Hyperledger Fabric ver 1.4 and validated various security and performance attributes. It is also verified that access to users' data assets is effectively regulated by numerous ACL rules. To measure key performance indicators of PrivySharing, we used Hyperledger Caliper, a blockchain benchmark tool. The experiments were performed on a machine with Intel Core i7 2.9 GHz CPU, 8 GB RAM, and Ubuntu 18.04 operating system.

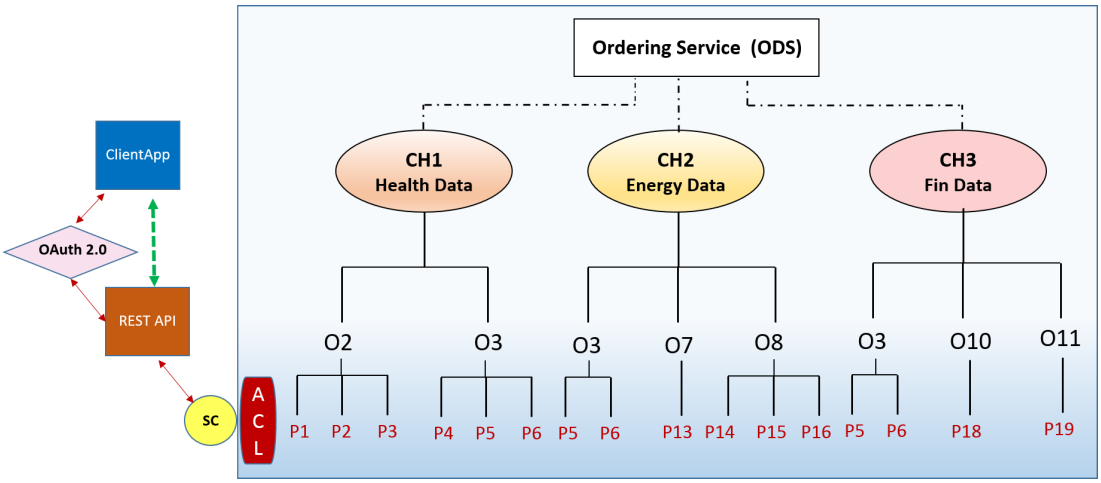

Figure 5.13: Experimental settings phase-1

Figure 5.14: Validation of assets access control

5.4.1 Validation of ACL Rules

The validity of the ACL rules was checked on both, the Hyperledger Composer-Playground and the REST API. E.g., As shown in Figure-5.14a, and 5.14b, to compare the access rights, we have created a user with admin rights that can view assets (blood alcohol level) of all the users, i.e., P5 and P6 in this case. Whereas, the user P5 with ID Pid5 can only see his assets. Moreover, Figure-5.14c, and 5.14d show that initially, a user P4 with id Pid4 cannot see any asset, as no asset is currently shared with him. However, once user P5 shares his blood alcohol level with P4, he can then see P5's blood-alcohol data. Similarly, only P5 can initiate a TX to share its assets. Whereas, if P4 tries to share the asset of P5 with any other entity, then he will get an error (as shown in Figure-5.15) as he currently does not have the right to initiate a data-sharing TX. As far

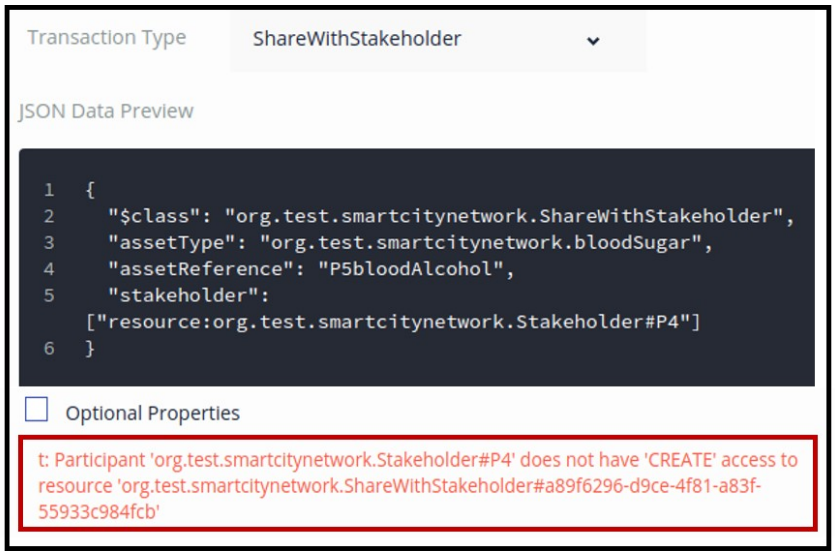

Figure 5.15: Validation of TX initiation rights

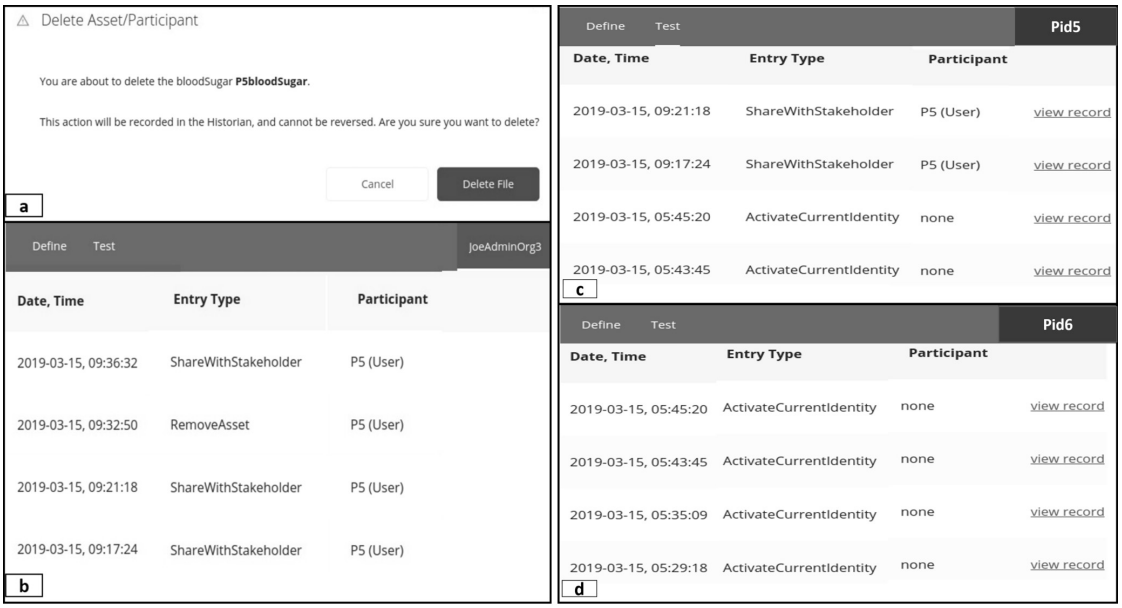

Figure 5.16: Historical record of purged data asset and visibility of TX history

as the purging of a data asset is concerned, as shown in Figure-5.16a, a data asset say P5's blood sugar can be deleted. However, Figure-5.16b manifests that the historical record (TX history) of a deleted asset remains immutable in the blockchain. Sequel to this, the TX history concerning the data assets can only be viewed by respective users only. As shown in Figure-5.16c, and 5.16d, only P5 (Alice) can view the record of her data sharing TXs. Whereas, any other user, say P6 (Bob), cannot see Alice's TX history. However, even if a blockchain admin is allowed to view the TX history of all the nodes for accountability, the admin still cannot see the value of the data asset being shared.

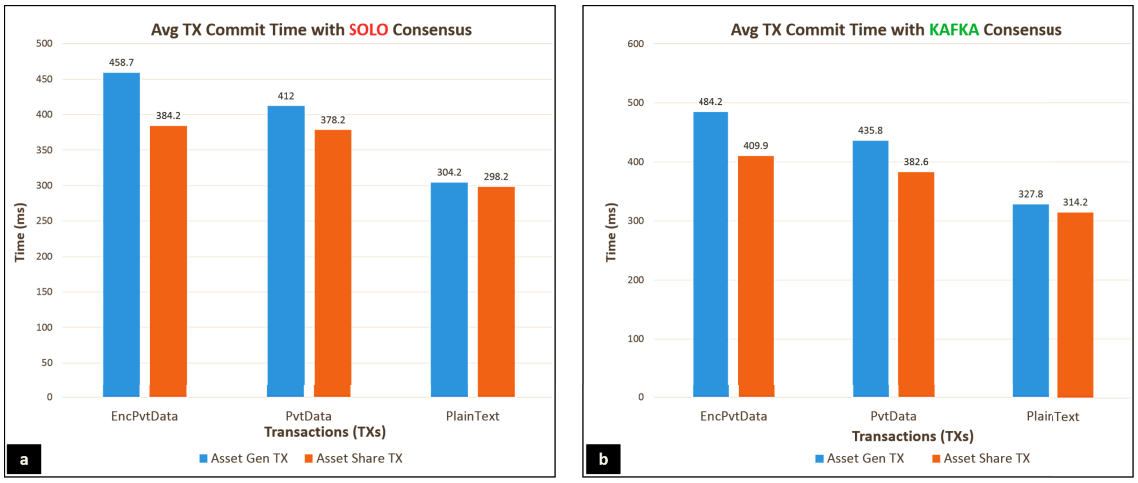

Figure 5.17: Avg TX commit time

5.4.2 Performance Efficiency

Though, a detailed comparison of performance efficiency of Hyperledger Fabric with some of its counterparts is already presented in [234] and [394]. However, as per the experimental settings for phase-1 (as shown in Figure-5.13), we measured the time taken to commit various types of TXs in the preview of PrivySharing. The avg commit time has been measured for three different TXs based on ten iterations each. The TXs include; plain text (PlainText) TX, private data (PvtData) TX, and encrypted private data (EncPvtData) TX. These TXs are analyzed in two different consensus environments, i.e., SOLO and Kafka.

It is evident from Figure-5.17 that all types of TXs irrespective of the employment methodology take less than 490 milliseconds (ms) to commit in a new block. However, there is a clear pattern that the EncPvtData TXs for both asset generation and sharing take more time to commit than the PvtData and PlainText TXs. Moreover, the time taken by an asset sharing TX is lower than the asset generation/creation TX in almost all three cases. Similarly, Figure-5.18 highlights the avg time taken for state validation, block commit, and state commit for asset generation and asset sharing TXs with SOLO and Kafka consensus both. It can be ascertained that the time taken for block commit (represented by rust strip) in all three cases, i.e., EncPvtData, PvtData, and PlainText TXs, does not show many variations. However, the state commit time (expressed in the grey strip) significantly reduces for the PlainText TXs with SOLO and Kafka consensus in both cases, i.e., asset generation and asset sharing TXs. Similarly, the overall TX commit time for a plain text TX is lower than the EncPvtData and PvtData TXs.

In the second phase of the experiment, we measured various performance indicators of PrivySharing using Hyperledger Caliper as per the settings shown in Table-5.4. For the initial test, we ran thirty rounds of the experiment for both one-Ch and three-Ch scenarios with Kafka ordering service (consensus). There were six peers and six clients operating in the one-Ch and two peers and two clients per Ch in the three-Ch scenario. A total of 300 TXs were input to the system

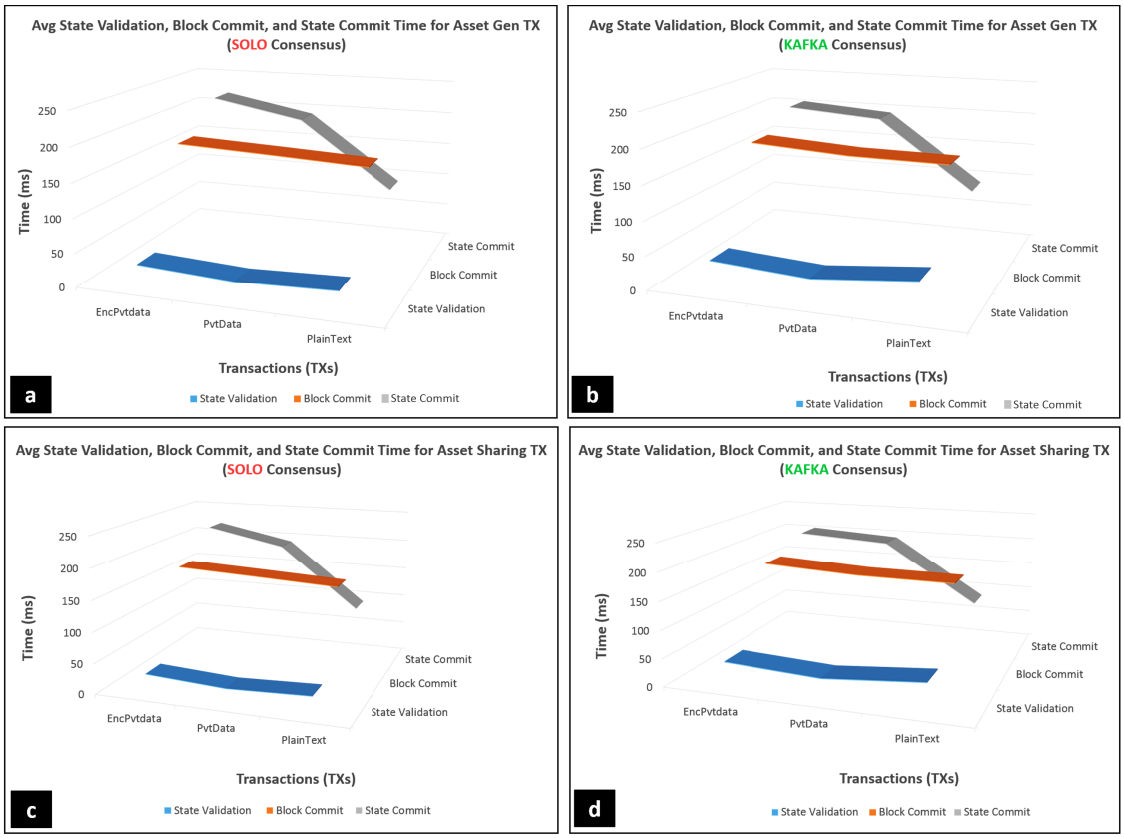

Figure 5.18: Comparison of state validation, block commit, and state commit avg time

Table 5.4: Experimental settings phase-2

Parameters	Settings for One CH Scenario	Settings for Three CHs Scenario
Number of Chs	1	3
Number of Input TXs	300	300
TX Send Rate	50 TPS	50 TPS
Number of Member Organizations	6	6
Peers Per Ch	6	2
Total Peers	6	6
Number of Orderer Nodes	4 Kafka Nodes 3 Zookeeper Nodes	4 Kafka Nodes 3 Zookeeper Nodes
Number of Clients	6	6
Number of Experiment Rounds	30	30

at the rate of 50 TPS in both scenarios. The highlight of this experiment as shown in Figure-5.19, is that the three-Ch scenario has demonstrated efficient performance as compared to the single-Ch scenario, with an avg throughput of 42.4 TPS and avg latency of 1.54 sec at the TX Send Rate of 50 TPS.

After this primitive comparison, we also determined the p-values [395, 396], for both the scenarios to substantiate our findings. In that, we first applied independent two-sample T-test on latency measurements to determine the p-value to accept or reject the null hypothesis, i.e., "The

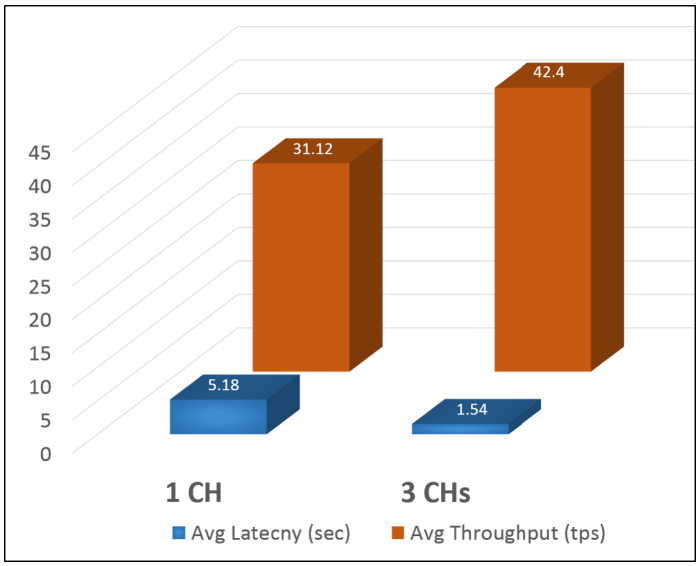

Figure 5.19: Comparison of avg latency and avg throughput in one-Ch and three-Ch scenario

avg latency of the one-Ch network is equal to the avg latency of the three-Ch network." Whereas, the alternative hypothesis is; "The avg latency of the one-Ch network is greater than the avg latency of the three-Ch network." The p-value resulted from the first test on system latency was 8.62 x 10^{-31}, which is less than 0.05. The result suggests the rejection of the null hypothesis in favor of the alternative hypothesis. Therefore, it is more probable that the avg latency of the one-Ch network is higher than the avg latency of the three-Ch system. Later, The second two-sample T-test was performed over throughput values. The null hypothesis in this case was; "The avg throughput of the one-Ch network is equal to the avg throughput of the three-Ch network." Whereas, the alternative hypothesis states that "The avg throughput of the one-Ch network is less than the avg throughput of the three-Ch network." The p-value emanated from this proceeding was 1.23 x 10^{-28}, which is smaller than 0.05. Hence, the result asserts the rejection of the null hypothesis in favor of the alternative hypothesis. Therefore, it is much likely that the avg throughput of the one-Ch network is smaller than the avg throughput of the three-Ch system. Hence, based on the p-values, it can be concluded that the one-Ch network has inferior performance in terms of high latency and low throughput as compared to the three-Ch network.

In the third phase of the performance testing, we mapped the correlation between different performance indicators for the three-Ch network. TX Send Rate was pitched against network latency and throughput, as per the test settings shown in Table-5.5. The experiment was run for ten rounds with varying TX Send Rate in each round. Although we had set specific TX Send Rate for the test case, however, the actual Send Rate that was executed by the system came out to be different. There were two peers, and two clients in each Ch to process and submit the TXs, respectively. Figure-5.20a, interprets the relationship between TX Send Rate and network latency. The avg latency increases uniformly until the TX Send Rate reaches around 106 TPS. After that, the latency starts fluctuating between 3 and 4 secs. Correspondingly, Figure-5.20b also highlights a similar

Table 5.5: Experimental settings phase-3

Parameter	Settings
Number of Chs	3
Number of Input TXs	300
TX Send Rate (configured) for Ten Rounds (TPS)	25, 50, 75, 100, 125, 150, 175, 200, 225, 250
TX Send Rate (actual) for Ten Rounds (TPS)	24.4, 47.4, 70.9, 89.6, 106.4, 116.3, 145.8, 154.3, 198.2, 199.1
Number of Member Organizations	6
Peers Per Ch	2
Total Peers	6
Number of Orderer Nodes	4 Kafka Nodes 3 Zookeeper Nodes
Number of Clients Per Ch	2
Total Clients	6
Number of Experiment Rounds	10

trend, in which the network throughput rises with the increase in the TX Send Rate. However, once TX Send Rate reaches 106, the throughput waffles between 50 and 56 TPS. We believe that such a result is induced by the small number of orderer nodes, which could not handle more than 200 TPS. Likewise, the latency in TX confirmation increases with the rise in TX Send Rate.

Later, we also studied the correlation between an increase in the number of peers and avg latency, and throughput respectively at varying TX send rates (as shown in Figure-5.21a, 5.21b, and 5.21c). For this test, there were six clients, and the number of peers varied from 6 to 24 in an increment of 6. It is observed that the throughput is mostly consistent with the send rate until the number of peers goes beyond 18. It can also be seen in Figure-5.21c that the throughput decreases notably as the number of peers reaches 24. Similarly, the latency also increases with the increase in the number of peers. Such a behavior can be attributed to the number of endorsing and orderer nodes in the network that has to endorse and pack the TXs in the blocks and broadcast new blocks, respectively. Moreover, it can also be accredited to the fact that for this experiment, all the peers were run on a single machine in a constrained environment. Hence, once in a distributed setting, each peer is expected to perform much better. It is also believed that the TX throughput can be scaled by load balancing TX endorsement across a pool of endorsers [397].

The experimental results uphold the idea of a multi-Ch blockchain network, as the same has demonstrated more throughput and less latency than the one-Ch system. The network latency and throughput in Hyperledger Fabric depend upon numerous factors, such as, application design, fabric network architecture, specifications of endorsement policies, complexities of ACL rules, application/chaincode language, number of endorsers and ordering nodes, the batch timeout, and the physical or the virtual network infrastructure [397]. Hence, a meticulously designed and laid out blockchain network and application can yield higher TX throughput with less latency. E.g., FabCoin built on top of Hyperledger Fabric can achieve a throughput of over 3560 TPS with Kafka ordering service [301].

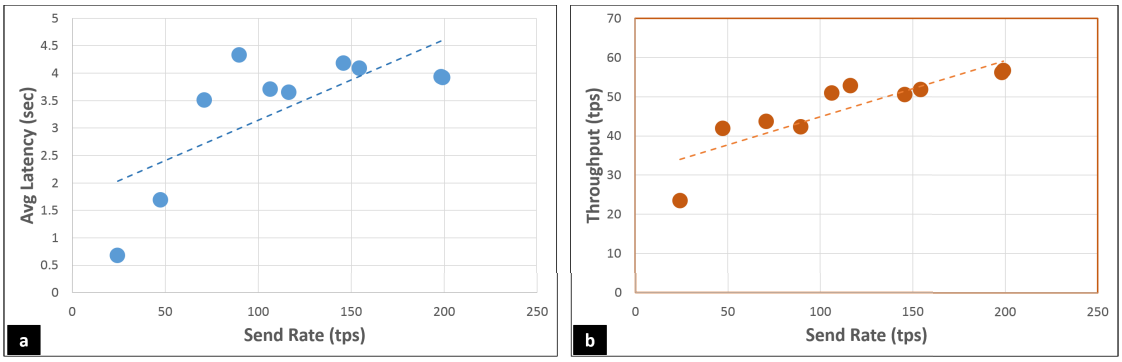

Figure 5.20: a) Correlation between TX send rate and latency. b) Relation between TX send rate and network throughput

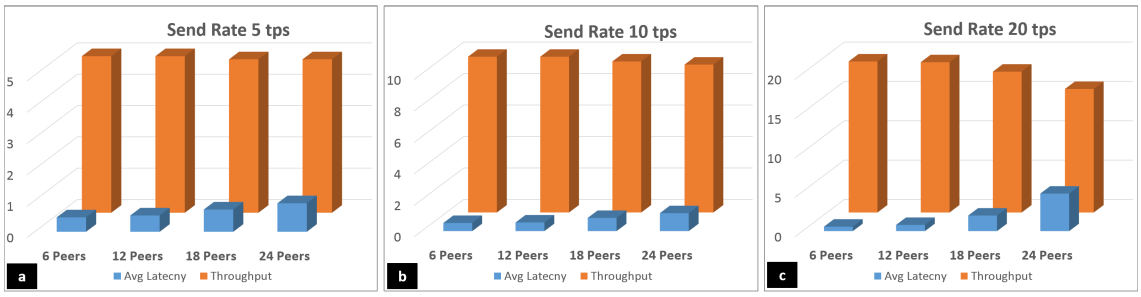

Figure 5.21: Correlation between the number of peers and network throughput at the send rate of (a) 5 TPS, (b) 10 TPS, and (c) 20 TPS

5.4.3 Limitation and A Way Forward

5.4.3.1 Storage of Multiple Ledgers by The Peers

The use of multiple data specific Chs is presumed to be scalable than a single Ch. However, since committing peers have to maintain numerous ledgers, there may be a massive resource requirement for such nodes in a vast smart city network.

5.4.3.2 A Way Forward

The concept of integrating edge computing into the mobile network architecture is not new [398]. Thereafter, researchers are exploring the idea of using Mobile Edge Computing (MEC) as a gateway for IoT devices to achieve low latency, data aggregation, processing, and real-time application response [399–401]. The deployment models of MEC range from Small Cell Cloud (SCC) [402, 403] to Mobile Micro Cloud (MMC) [404], MobiScud (Fast Moving Personal Cloud) [405], Follow Me Cloud (FMC) and etc. In all these MEC concepts, the first point of contact between the User Equipment (UE) and the mobile network is Small Cell evolve NodeB (SCeNB) or evolve NodeB (eNB). However, depending upon the MEC architecture the computational and storage resources are located (can be in hardware or virtual form) at SCeNB/eNB for SCC and MCC and

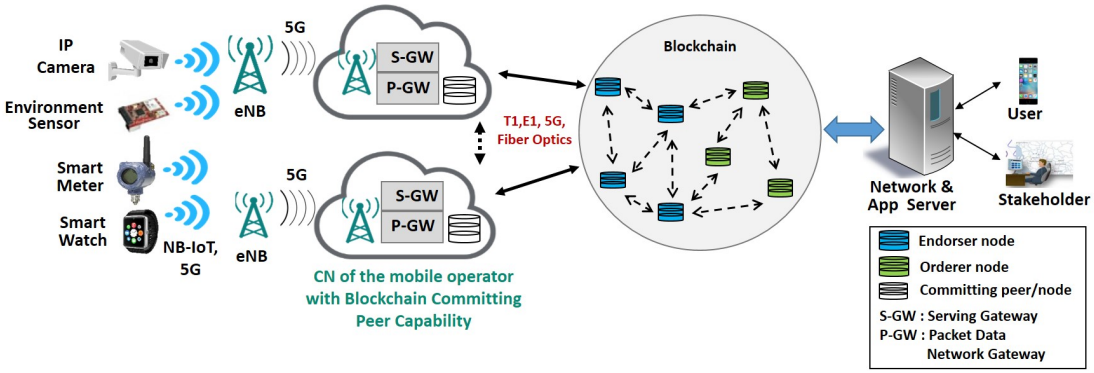

Figure 5.22: Integration of blockchain with MEC

at distributed Core Network (CN) in the case of FMC. However, FMC with decentralized control and distributed architecture is the preferred choice over SCC and MCC [406].

We believe that based on the edge computing concept, we can integrate blockchain with MEC to relieve end nodes from maintaining many ledgers. In this context, the SCeNB/eNB or CN nodes (in case of FMC architecture) can be harnessed with a suitable blockchain platform to facilitate fast TX settlement and provision of swift data processing and analytics services. Moreover, the end nodes can send queries for data (authorized to them) to the MEC nodes. To realize this concept, we propose a solution based on the FMC model, as shown in Figure-5.22. As of today, almost every inch of a populated area has cellular coverage, and most of the latest IoT devices also support NB-IoT technology. NB-IoT is a sub LTE frequency band, and soon, all the telecommunications companies (telcos) will be able to provide NB-IoT services. Moreover, the launch of 5G mobile network technology is also imminent. Hence, IoT devices can send sensor data to the MEC nodes via NB-IoT/5G. The MEC nodes being resourceful in terms of infrastructure, computational power, storage, and energy can also act as a blockchain committing peer. In this way, we can utilize the existing infrastructure of MEC/cellular networks without incurring high costs. The MEC node can then communicate with the endorsing nodes/peers using a backhaul network (5G, E1, T1, fiber optics, satellite, etc.) and existing infrastructure at any distance. The inherent communications security of fiber optics, NB-IoT [26], 4G, and 5G [407,408] technology will also add another layer of security over the blockchain P2P communication.

Turning a MEC node into a blockchain committing peer will be safe from the data security point of view, as the committing peers do not install and run the SCs. Hence, the SC TX logic will not be visible to them. Moreover, to incentivize the cellular companies for their services, they can be paid some TX fee as a reward in terms of the local digital token, e.g., PrivyCoin. Another advantage of integrating blockchain with the MEC model will be ease in mobility management (e.g., handover) of end nodes/user devices if they move throughout the network.

5.5 Summary

User data generated by today's smart devices ranging from smartwatches to smart cars, smart homes, auto-pay systems, ITS, etc., are vulnerable to privacy and security threats. Moreover, users also reserve the right to manage and control access to the data they own. Therefore, in this chapter, we introduced "PrivySharing," an innovative blockchain-based secure and privacy-preserving data sharing mechanism for smart cities. The proposed strategy ensures that personal/critical user data is kept confidential, securely processed and is exposed to the stakeholders on the need to know basis as per user-defined ACL rules embedded in smart contracts. Moreover, the data owners are rewarded for sharing their data with the stakeholders/third parties. PrivySharing also complies with some of the fundamental EU GDPR requirements, such as data asset sharing, accessibility and purging with data owner's consent. In addition, the experimental results verified that a multi-Ch blockchain solution scales better than a single Ch blockchain system.

PrivySharing not only meets most of the security requirements for IoT systems but also the performance requirements specified in Chapter-4, Section-4.2.1, and Section-4.2.2 respectively. E.g., the proposed framework provides a trustless environment with distributed storage and decentralized control. It also ensures data authentication, integrity, and availability with optional data confidentiality. Moreover, user security is augmented by the option of using multiple-pseudonymous IDs for interactions with different stakeholders/third-parties. Also, PrivySharing performs efficiently with high TX throughput as compared to Bitcoin, Ethereum, and IoTA. It also provides instant TX confirmation with less communication complexity.

Although PrivySharing provides a secure and privacy-preserving mechanism for users in a smart city environment. However, as identified in Chapter-4, Section-4.7, still some research is required to design an IoT-oriented consensus protocol. The protocol should be designed to increase the tolerance of maximum possible faulty/Byzantine nodes. It should also be scalable and must not degrade with the increase in the number of network validator/miner nodes. Moreover, besides providing instant TX confirmation, the consensus protocol should be safe against DoS attacks and should also have minimum communications and computational overheads. Most importantly, the consensus protocol should verify IoT TXs based on some IoT-centric TX validation rules.

”Our deeds determine us, as much as we determine our deeds.”

- George Eliot

6

Pledge: A PoH-based Consensus Protocol

This chapter proposes “Pledge,” a novel Proof-of-Honesty (PoH) based consensus protocol for IoT environment. It reduces the possibility of participation by faulty, malicious, and non-performing nodes in the blockchain consensus process. Pledge is believed to be a secure protocol with proven scalability and low communications complexity. Another contribution of this research is the introduction of IoT-oriented TX validation rules that prevent malfunctioned IoT devices to submit faulty sensor readings. It is also imperative to mention that the work in this chapter was presented in the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) [409].

6.1 Introduction

A comprehensive discussion on blockchain consensus protocols is already presented in Chapter-4, Section-4.5.1. Therefore, only some of the key points are reviewed here. The consensus mechanisms in blockchain technologies comprise two phases. First is the leader election to propose the next block, and in the second phase, the agreement is achieved by the majority of the network nodes on the order of the TXs in the block proposed by the leader. There are two main approaches to network consensus in blockchain-based applications: Nakamoto consensus and BFT. Nakamoto consensus [24] is a PoW-based protocol that elects a leader through a lottery scheme based on computation power and the agreement on the order of the TXs in a block is achieved through the longest chain rule. Due to the probabilistic nature of the Nakamoto consensus, temporary forks occur, which are likely to cause latency in TX confirmation, thus resulting in low TX throughput [278]. This delay in TX confirmation is not suitable for most of the real/near-real-time IoT systems requiring instant TX finality. In addition, [410] also highlighted numerous security

risks in PoW-based blockchains.

On the other hand, traditional BFT algorithms such as PBFT [289,292,342], and DBFT [210], select the next block proposer in a round-robin fashion and use multiple rounds of explicit voting by a limited number of chosen validators to achieve consensus. Moreover, BFT-based protocols are also susceptible to DoS attacks due to their dependence on weak timing assumptions for liveliness [207, 234]. Consequently, weak synchrony also adversely affects the throughput of such systems [207]. BFT consensus protocols provide consensus finality yet they have high communications complexity, which often involves as many as $O(n^2)$ messages per round [342]. BFT protocols also fail to operate correctly in the presence of more than $1/3$ faulty/malicious nodes. Hence, there is a strong fault-threshold assumption in BFT protocols that at least two-thirds of nodes are honest [289]. Moreover, to mask Byzantine faults, the computations are performed on numerous replicated nodes and the results that repeat more than a threshold number (mostly $2/3$ of the total nodes) of times are accepted. However, in this mechanism, the faulty nodes can only be detected at the end of the computation after wasting system resources. Additionally, this technique relies on a single trusted entity to decide which computations are correct [411].

Correspondingly, to avoid Sybil attack, Algorand [296], a cryptocurrency, employs a BFT-based protocol in which the committee members are randomly selected for the consensus process based on a weighted sum of wealth they own. Similarly, to resolve the issue of nothing at stake, Clique, a Proof-of-Authority based consensus protocol [412], was developed. Clique was conceived based on a statement of Warren Buffet [413], where he said that "It takes twenty years to build a reputation and five minutes to ruin it. If you think about that, you will do things differently." Hence, Clique puts the users' real-world IDs/reputation at stake, where all the participants value the stake equally. To establish the authenticity of user IDs, the public notaries being the trusted parties perform the on-chain ID verification [412]. However, it is believed that the integration of a public notary or such a government entity to a private/consortium blockchain will take some time to realize due to a lack of legislation/rules and policies on the subject. Moreover, the validation of identities by a trusted third party is against the decentralization spirit of the blockchain.

Concerning TX validation rules, this aspect has been amply highlighted in Chapter-4, Section-4.5.2. It is believed that TX validation rules of cryptocurrency may not be useful for IoT device TXs [234,382]. Hence, there is a requirement of an IoT-centric consensus protocol that must conform to IoT-oriented TX/block validation rules, prevent DoS attacks (exploiting timing assumptions), provide increased fault tolerance (greater than $1/3$ faulty nodes), and near instantaneous TX confirmation with low communications complexity. Thereby, we present "Pledge": A Proof-of-Honesty (PoH) based consensus protocol for blockchain supported IoT systems.

6.1.1 The Motivation

The key idea of selecting a block proposer based on honesty came from the Islamic concept of "Bayt and Shura," which forms the basis of the Islamic democratic political system. In the case of political succession, bayt is the act of nominating and accepting a potential ruler [414]. The

process of bayt consists of two stages. In the first stage, certain selected individuals from the community called "Ahl al-Hal wal Aqd" (meaning those who can enter into a contract or dissolve it) conduct extensive consultations and then nominate the potential Khalifa (Ruler). Whereas, in the second stage, the nomination is accepted by the general public as a formality. In another form, the same concept is used by shura for advising the Khalifa on governance issues. The literal meaning of shura is "The principle of consultation," and it is applied to the government. Whereas, "The cooperation of all," is the fundamental pillar of shura [415]. The people who form part of shura must have the following qualities: They should be honest, truthful, just, wise, and must have a good reputation. They should also possess the kind of knowledge that would enable them to make the best decisions. Moreover, any decision or advice by shura must not contradict the laws/teachings of the Quran and Sunnah [414].

Similarly, authors in [234] proposed that to lessen the effect of faulty nodes, an integrity check of the validator/mining nodes must be carried out. So that no dishonest node participates in the consensus process. Hence, by applying above mentioned concepts to the blockchain consensus protocol, instead of selecting a block proposer based on its material properties such as computation power or wealth (coins), it is ascertained that a block proposer should be chosen based on specific attributes that reflect upon the node's integrity and character. Moreover, the overall block proposal process should not contradict the consensus rules concerning IoT TX validation.

6.1.2 Related Work

Authors in [416] proposed a P2P reputation system based on blockchain for the users involved in file sharing. Once a user receives a file, it initiates a signed blockchain TX containing reputation score, time stamp, and the hash of the received data. The miner node checks the validity of the TX by asking all the users involved in the TX to send a signed proof of the file-sharing, i.e., the hash of the file, and a nonce sent by the miner. However, the users have to stay online for the miner verification. The authors claim that the scheme is resilient against unfair rating and collusion attacks. Moreover, the proposed methodology restricts the users from generating multiple IDs by linking the ID creation to the IP address of the user. Nonetheless, due to the additional TXs concerning reputation score propagation by the peers and the confirmation of the file-sharing TXs by the miners, there is an immense load on the network bandwidth and a rapid increase in the blockchain size. Hence, the increased resource requirements for the miners entail fewer miners, thus lower security of the network. Moreover, as the users rely on the miners to compute and send the reputation score concerning a particular file owner, the network latency and the processing time of the request add a considerable delay in the execution of file sharing TXs. This issue may pronounce if malicious users collude to overload the miners with a large number of TXs. Thus forcing miners to perform computationally expensive verification of TXs and causing requests of legitimate users to be queued.

Correspondingly, there are other proposed P2P reputation models as well [417,418]. However, they have either weak assumption that users are honest in their ratings of other peers or they do

not cater for the presence of malicious actors in the network. Moreover, there is also no ID management. Hence, a malicious attacker can create multiple IDs and unfairly increase his reputation. Similarly, authors in [419] developed a mechanism for the detection of fairness policy violations in public and private blockchain networks. In this context, a block proposer cannot: (1) Deny that it did not receive a particular TX. (2) Control that which TX to include in the proposed block. (3) Manipulate the order of TXs in a block. The proposed scheme introduces a method of One Way Accountable Channel (OWAC), that helps in detecting TX dropping and TX re-ordering by the miners or nodes relaying the TXs. In another work [420], researchers proposed a reputation model based on transient trust. Hence, the service providers make a decision based on this transitive reputation that whether to accept a particular attribute of a customer from a specific AP or not. The AP stores the user's credentials/attributes in a secure database and when a service provider (SP) wants to verify the user based on his attributes, the user provides a cryptographic token to the SP to be presented to the AP to get access to the attributes.

Similarly, authors in [421] proposed a Proof-of-Trust (PoT) based framework to decrease the difficulty of PoW. Hence, the more trusted a node is, the less work it performs. PoT implies that each peer in a blockchain network declares his trust towards other nodes, thereby constructing a trust graph that is later used to compute trust matrics, which is stored on the blockchain. The PoT assumes that the members of the Trusted Candidate Set (TCS) are honest. However, there is a challenge to control the extent to which an adversary can sabotage the trust graph to illegally increase his trust ratings. There is also a danger of few highly trusted nodes dominating the consensus process. Moreover, there is a question that do we trust all the peers to give a genuine opinion about others? [416]. In addition, recently researchers in [422] proposed a reputation-based consensus protocol. However, the proposed model measures the reputation based on the age of currency held by the node, its social interaction and consistent participation in the consensus. Moreover, a block including the TX sub-block and the reputation sub-block is validated based upon majority voting by atleast 2/3 high reputation nodes.

Therefore, the main contribution of this research is to introduce “Pledge,” a PoH-based consensus protocol with an IoT-oriented TX validation scheme. Pledge aims to reduce the participation of faulty, corrupt/malicious, and non-performing nodes in the consensus process, thereby increasing the fault tolerance to the maximum. Pledge is a PoH-based consensus protocol in which the block proposers are selected based upon the cumulative score of their honesty attributes. Whereas, the attributes are collected internally from the blockchain. Hence, we take advantage of the inherent benefits of blockchain, i.e., data immutability, ability to operate in a trustless environment, and protection against data forgery. Consequently, no trusted IDP (Identity Provider), AP, Notary Public, or a third party is required to validate the attributes. Therefore, it is nearly impossible to forge or emulate fake honesty attributes. Pledge also restricts the block proposal responsibility to a couple of nodes randomly selected out of honest nodes in the network. By denying dishonest nodes' participation in the consensus process, Pledge reduces the probability of malicious behavior by a validator node during consensus. The proposed protocol protects against Sybil attack, which was one of the significant factors that formed the basis of the PoW consensus algorithm [207,272].

We believe that Pledge will prove to be a governing factor for IoT systems considering to adopt blockchain technology.

6.1.3 Organization

The rest of the chapter is organized as follows: Section-6.2 unfolds the properties of an ideal consensus protocol, Pledge methodology, and context-aware TX validation rules. In Section-6.3 comprehensive security and performance analysis of Pledge is presented. Finally, the chapter is concluded in Section-6.4.

6.2 The Pledge Protocol

Before getting into the details of the Pledge protocol, it is important to first sift through the properties of an ideal consensus protocol for blockchain-based IoT systems.

6.2.1 Properties of an Ideal IoT-Centric Consensus Protocol

It is envisaged that an optimal consensus protocol for blockchain-based IoT systems should satisfy the following properties:

a. **Fairness.** All nodes should have an equal chance of being selected as the block proposer.

b. **Investment.** The cost of the block proposer selection process should be proportional to the value gained from it.

c. **Verification.** It should be relatively simple to verify that the block proposer was legitimately selected [423].

d. **Honesty.** Nodes participating in the consensus process should have a high probability of being honest.

e. **Termination.** All honest nodes finally decide on a block.

f. **Agreement.** All honest nodes agree on the same block.

g. **Validity.** The block that is being agreed upon should be from a legitimate node [424].

h. **Consensus Finality.** A block, once agreed upon and appended to the digital ledger, is not removed any time later [272].

i. **BFT.** The consensus protocol should be able to propose a valid block even in the presence of a large number of faulty, corrupt, or malicious nodes.

j. **Unforgeability.** The block proposer selection process should be unforgeable, and no node should be able to emulate fake attributes.

k. **Security.** The system should be resilient against common attacks on reputation systems, and also does not subvert the fundamental security guarantees of the blockchain.

l. **Decentralization.** The consensus protocol should not be quasi-centralized by abandoning the decentralization property of the blockchain.

m. **Scalability.** The consensus algorithm should scale well with the increase in the number of network nodes without increasing the communication complexity.

6.2.2 Pledge Methodology

The design of the Pledge is based on certain assumptions.

a. **Assumptions.** There is a likelihood of Byzantine failures in the blockchain network. Correspondingly, the Byzantine nodes are not expected to follow the protocol. Moreover, an adversary may control and manipulate the behavior of the nodes resulting in deteriorated performance. The adversary may also disrupt the communications and split the network. Nonetheless, being a consortium blockchain with identity management (IDM), it is very difficult for the malicious nodes to impersonate other honest nodes. Lastly, it is assumed that a typical private/consortium blockchain-based IoT system comprises a large number of resource constrained end-nodes (IoT devices). Besides, it has a limited number of miner-nodes (potential block proposers) that can generate new blocks and maintain a copy of the blockchain. Hence, the term "node" in this chapter refers to a miner-node/potential block proposer.

b. **Pledge Protocol.** When a new block is published, or the blocks proposed by the proposers of the last round are rejected, the consensus process to select the next pair of block proposers starts. As shown in Figure-6.1, an honesty metric (H_{Mat}), is computed and maintained for all the registered miner-nodes (potential block proposers) on the blockchain. Hence, whenever a block is successfully appended to the blockchain, an event [425] is triggered that starts the process of updating the H_{Mat} for every node based on the predefined attributes extracted/computed through the blockchain. The value of each attribute is obtained and weighted to compute the H_{Mat} for every node. Subsequently, a cumulative H_{Mat} score is calculated for each node, i.e., $H_{MAT_1}CumScore$, $H_{MAT_2}CumScore$, and $H_{MAT_3}CumScore$ respectively for Node 1, Node 2, Node 3, and so on. Next, a priority list of K honest nodes is formed based on $Honesty_{MAT}$, which comprises the individual $H_{Mat}CumScore$ of all the nodes.

   The nodes with $H_{MAT}CumScore >= H_{MAT}Threshold$ form part of the K honest nodes list. It is followed by a random selection of "Primary" and "Secondary" block proposers for the next block, from the K honest nodes. Finally, the primary proposes a new block followed by the validation of its $H_{MAT}CumScore$ and TXs in the proposed block by the rest of the K honest nodes before that block is committed. If there is any violation of the TX validation rules or the $H_{MAT}CumScore$ of the primary was not computed correctly, the proposed block is rejected, the primary is blacklisted, its owner organization is reprimanded, and a new block is introduced by the secondary proposer. The same checks are performed on the blocks proposed by the secondary proposer as well, and if a block is valid, then it is accepted and appended to the chain by all the nodes. Otherwise, secondary is also blacklisted, and new primary and secondary block proposers are selected for the current round. Finally, when the block is accepted and appended to the blockchain, the next round of $Honesty_{MAT}$ computation, and selection of a new primary and a secondary block proposer starts.

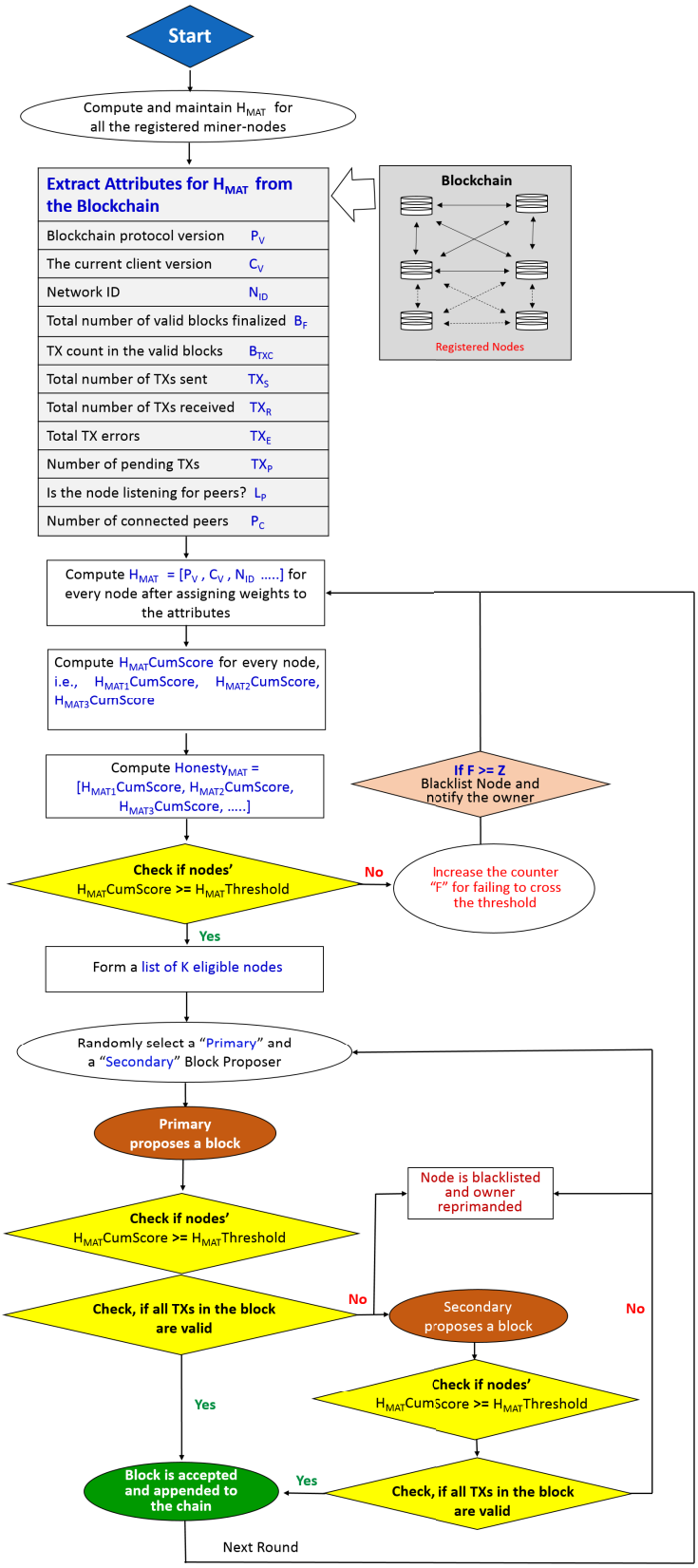

Figure 6.1: Pledge methodology

Table 6.1: Attributes' scoring criteria

Attributes	Weight Criteria	Attribute Score	
		Min	Max
Blockchain protocol ver (P_V)	Ver up-to-date or not	-1 for old ver	1 for latest ver
Client ver C_V	Ver up-to-date or not	-1 for old ver	1 for latest ver
Network ID (N_{ID})	Correct or not	-10 for incorrect ID	1 for correct ID
Number of valid blocks proposed (B_F)	1 mk for each block	0	f
TX count in the previous valid blocks (B_{TXC})	1 mk for every TX	0	c
Ratio of TX Errors vs. TXs sent (TX_E / TX_S)	For $TX_E >= 1$, multiply the ratio by -1	-1	0
Number of TXs received (TX_R)	1 mk for every TX	0	r
Number of pending TXs (TX_P)	-1 mk for every pending TX	-p	0
Is the node listening for peers? (L_G)	-10 mks for not listening	-10	1
Number of connected peers (P_N)	2 mks for each connection	0	n

The biggest challenge in this process is the selection of the attributes that optimally describe the honest behavior of the nodes and further help in identifying faulty, malicious, and Byzantine nodes. These attributes may be different for every blockchain technology such as Bitcoin, Ethereum, Hyperledger Fabric, etc. However, we have determined some traits common to every blockchain platform. As shown in Fig. 6.1, the first three attributes, including blockchain protocol version running on the node, the client application version, and the network ID, may contribute to the faulty or impaired behavior by a node. Whereas the rest of the attributes such as the number of valid blocks proposed, total number of TXs included in the valid blocks, TX errors, number of TXs sent and received, number of pending TXs at a particular moment, number of connected peers and whether the node is listening for peers or not, reflect the conscientious performance of the node. If a node is honest, its performance would be exceptional as it will mine more blocks with the maximum possible number of TXs in a block. An honest node is also expected to be connected to most of its neighboring nodes and process a high number of TXs. Depending upon the type of blockchain platform, some other attributes can also be included, such as for Bitcoin or any other fintech blockchain, the total reward earned by a node, number of confirmations for the TXs, and number of blocks relayed can be considered.

Additionally, inactivity can also be an attribute such that any period of inactivity higher than time Δt will earn a negative score for each period of non-activity exceeding Δt. Similarly, for PoS-based blockchains, current balance can be one of the attributes. In addition to the specific aspects, certain facets resonating misbehavior of the nodes, and anomalies in their performance, can be detected by employing a layer of deep learning over the blockchain network.

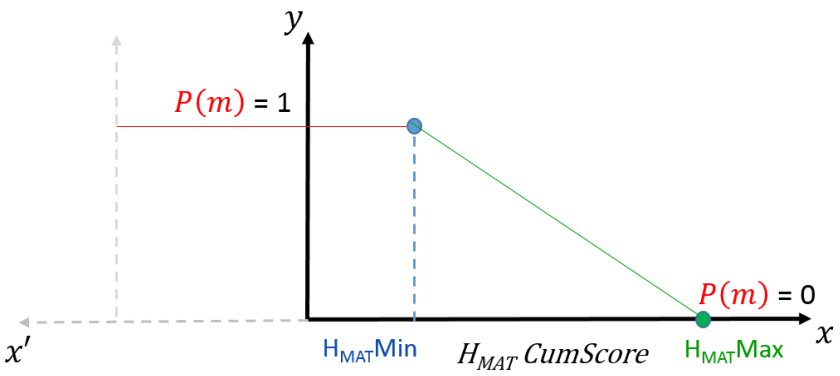

Figure 6.2: Probability of being malicious

6.2.3 Computing H_{MAT}CumScore

This section illustrates the approach adopted to model $H_{MAT}CumScore$ in respect of a node based on the weighted sum of its character traits/attributes. This model is not a hard and fast rule; instead, it may vary from system to system based upon the sensitivity/criticality of the application. Table-6.1 shows the attributes that are evaluated, the weight/scoring criteria, and the range of minimum (min) and maximum (max) values for each attribute. The min $H_{MAT}CumScore$ a node can secure is defined by (6.1), and the max $H_{MAT}CumScore$ score that can be achieved by a node is represented by (6.2).

$$H_{MAT}MinCumScore = -23 - pTX_P \tag{6.1}$$

$$H_{MAT}MaxCumScore = 4 + fB_F + cB_{TXC} + rTX_R + nP_N \tag{6.2}$$

Taking into account the $H_{MAT}MinCumScore$, and $H_{MAT}MaxCumScore$, ideally the probability of a node being malicious $P(m)$ (as shown in Figure-6.2) is close to one, if the node's $H_{MAT}CumScore$ is equal to $H_{MAT}MinCumScore$. However, practically the probability of having a malicious node can be calculated as

$$P(m) = \begin{cases} 1, & \text{Malicious node} \\ ax + b & \\ 0, & \text{Honest node} \end{cases}$$

where x is a random variable that represents $H_{MAT}CumScore$. Accordingly, the slope of the line $(ax + b)$ can be defined as:

$$\frac{x2 - x1}{y2 - y1} = \frac{H_{MAT}Max - H_{MAT}Min}{0 - 1} \tag{6.3}$$

Correspondingly the point-slope form can be represented as (6.4):

$$P(m) = \left(\frac{-1}{H_{MAT}Max - H_{MAT}Min} \times x\right) + \left(\frac{H_{MAT}Max}{H_{MAT}Max - H_{MAT}Min} + 1\right) \quad (6.4)$$

which can be further simplified to:

$$P(m) = \frac{-x + H_{MAT}Max}{H_{MAT}Max - H_{MAT}Min} \quad (6.5)$$

Now, by substituting (6.1) and (6.2) into (6.5), we can calculate the probability of a node being malicious, i.e,. $P(m)$, while $H_{Min} \leq x \leq H_{Max}$:

$$P(m) = \frac{-x + 4 + fB_F + cB_{TXC} + rTX_R + nP_N}{27 + fB_F + cB_{TXC} + rTX_R + nP_N + pTX_P} \quad (6.6)$$

Another important aspect is modelling the $H_{MAT}Threshold$, such that at a particular moment all the nodes that have $H_{MAT}CumScore >= H_{MAT}Threshold$, will be included in the list of K eligible block proposers. Consequently, the probability of a node being malicious will be less than 0.5, if the node's $H_{MAT}CumScore > H_{MAT}Threshold$. Hence, we define $H_{MAT}Threshold$ to be the avg value of $H_{MAT}Max$ and $H_{MAT}Min$, which can be represented by (6.7):

$$x = \frac{-19 + fB_F + cB_{TXC} + rTX_R + nP_N - pTX_P}{2} \quad (6.7)$$

$H_{MAT}Threshold$ being the avg of the $H_{MAT}Max$, and $H_{MAT}Min$, is dynamic and will rise with the increase in $H_{MAT}Max$, as the honest nodes continue to perform better with the passage of time.

6.2.4 IoT-Oriented TX Validation

In Chapter-4, we identified the need for IoT-oriented TX validation rules, and proposed a way forward. The foremost requirement for IoT systems is that the TXs should be validated based on context-aware TX validation rules. It is essential since every new TX in IoT is mostly independent of the previous TX, and a hardware malfunction, software bug, or a change in environmental conditions can induce variations in the sensor readings. The context-aware TX validation rules not only protect against malfunctioned sensors but also against malicious block proposers. Therefore, IoT TX validation rules should be carefully drafted, and they must incorporate environmental context based on the deployment scenario. This methodology can be described clearly with the help of a smart home and supply chain management system case study shown in Figure-6.3.

a. **Smart Home.** In a smart home scenario, during winters, if the temperature sensor installed in a room initiates a TX showing the temperature below threshold, e.g., 2^oC, to ignite the fireplace. This TX will only be considered valid if, during time Δt (Δt can be any value depending upon IoT application, in which co-located sensors can observe the same event and report upon it),

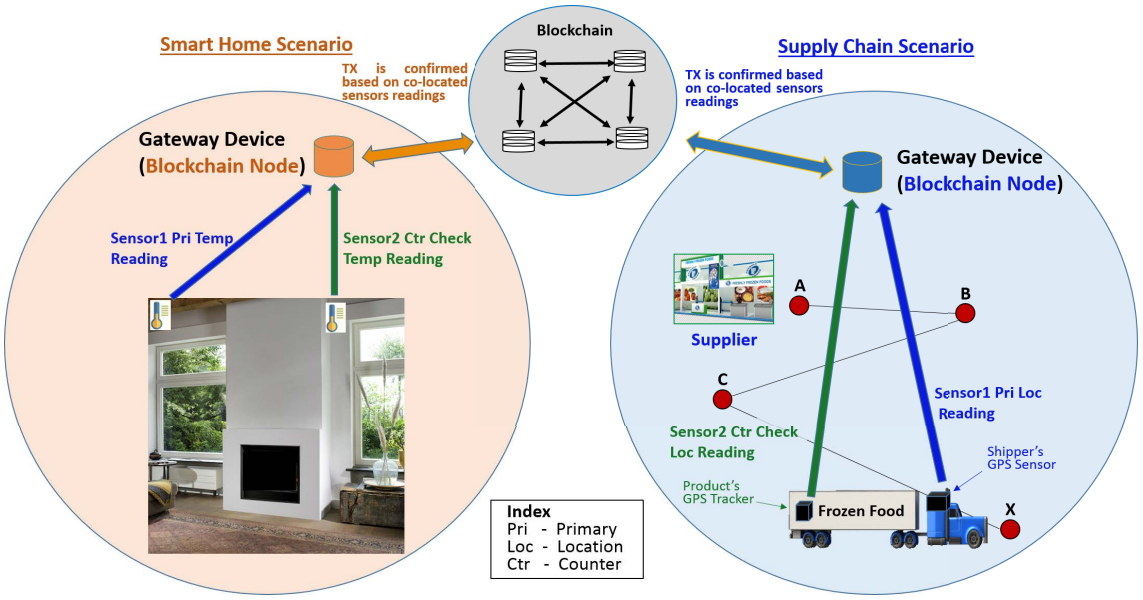

Figure 6.3: IoT TX validation rules

another sensor installed in the same room also initiates a TX indicating the occurrence of the same event, i.e., falling of temperature below the defined threshold. Such confirmation will not only protect against random faults in the sensors but also ensure validation of the TXs based on multiple sensors readings. Depending on the sensitivity of the location/application, multiple cross-checks can be included as rules to verify different types of TXs initiated by IoT sensors.

b. **Supply Chain Management (SCM).** Let us suppose that a shipment of frozen food is being monitored for swift movement on a pre-defined route from point A to point X (as shown in Figure-6.3). Therefore, when the shipper initiates a TX confirming that the shipment has reached the desired customer at location X, this TX will only be considered valid if, during time Δt (Δt can be any value depending upon IoT application, in which co-located sensors can observe the same event and report upon it), some of the GPS sensors attached to the frozen food package also initiate TXs indicating the exact location of the package. The package's GPS sensors can be easily programmed to initiate a TX, once the consignment reaches location X. These cross-checks will not only protect against any TX initiated with malicious intent by the shipper but also detect a malfunctioned IoT sensor.

6.3 Security Guarantees and Performance Analysis

Pledge offers numerous security guarantees with a scalable performance by satisfying most of the requirements of an optimal consensus protocol discussed in Section 6.2.1.

a. **Fairness.** Every node has an equal chance of being elected as a primary or a secondary block proposer if it satisfies the $H_{MAT}Threshold$ requirement.

b. **Investment.** The leader selection process in Pledge is neither computationally expensive like PoW nor does it require specialized hardware, as in the case of PoET [286]. Hence,

Table 6.2: Storage requirements for the attributes

Attribute	**Storage Requirement (Bytes)**
Blockchain Protocol Ver P_V	One
Client Ver C_V	One
Network ID N_{ID}	One
Valid Blocks Finalized B_F	Four
Valid TX Count in the Valid Blocks B_{TXC}	Four
TXs Sent TX_S	Four
TXs Received TX_R	Four
TXs Errors TX_E	Four
TXs Pending TX_P	Four
Is the Node Listening L_P	One
Number of Connected Peers P_N	Two

the computation, energy, and storage costs of selecting a block proposer are very economical. E.g., Depending upon the number of attributes to be evaluated for the computation of $H_{MAT}CumScore$ (eleven attributes in our case), there are eleven get operations to read the state of desired attributes from the blockchain. The storage requirement for these attributes, as shown in Table-6.2, sums up to be at the most thirty bytes. Moreover, the computation of $H_{MAT}CumScore$ for a particular node requires one add operation.

Correspondingly, to measure the cost of computing the $Honesty_{Mat}$, and selection of the two block proposers, a simulation of Pledge protocol was run on Ethereum blockchain. The experimentation was performed using the Remix-Ethereum IDE compiler ver 0.5.19 [426], and Geth (Go Ethereum) ver 1.8.27 deployed on a machine configured with an Intel Core $i5$, 6^{th} generation processor, and $8GB$ RAM. As shown in Figure-6.4, avg TX cost (in terms of gas) was computed by running thirty iterations of the Pledge protocol for each set of nodes varying from ten to hundred (total 300 iterations). It can be seen that the TX cost increases linearly with the number of network nodes. The avg increase in the TX cost is $6,46,071$, with the addition of every set of ten new nodes. Nonetheless, considering the ethereum block gas limit of $8,000,000$ (8 million) gas for a block [427], and gas consumption of under 6500000 (6.5 million) for hundred nodes, it can be concluded that the proposed PoH-based block proposer selection process is relatively economical in terms of computational costs.

c. **Verification.** All the nodes in the network continuously try to improve their performance so that they increase their probability of being selected into the list of K honest nodes and finally get elected as block proposers. However, the verification of such a selection is straightforward. When a primary and the secondary block proposers generate a new block, the rest of the nodes in the list of K nodes run the get operation to retrieve the latest state of the attributes in respect of the block proposer and compute the $H_{MAT}CumScore$ for verification by running just one

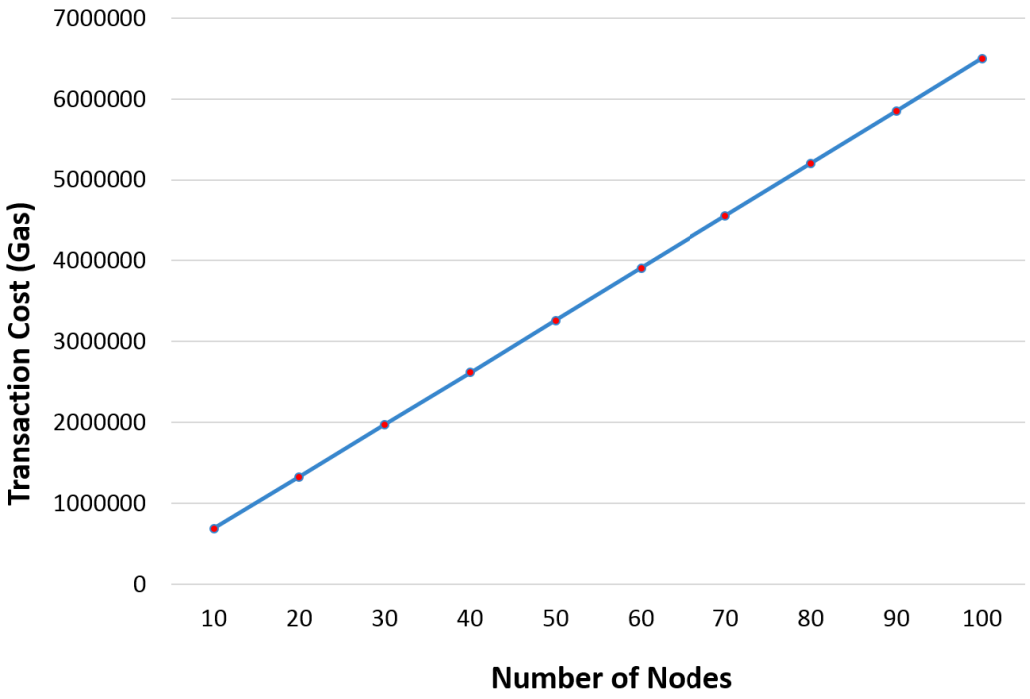

Figure 6.4: Transaction cost vs. Number of nodes

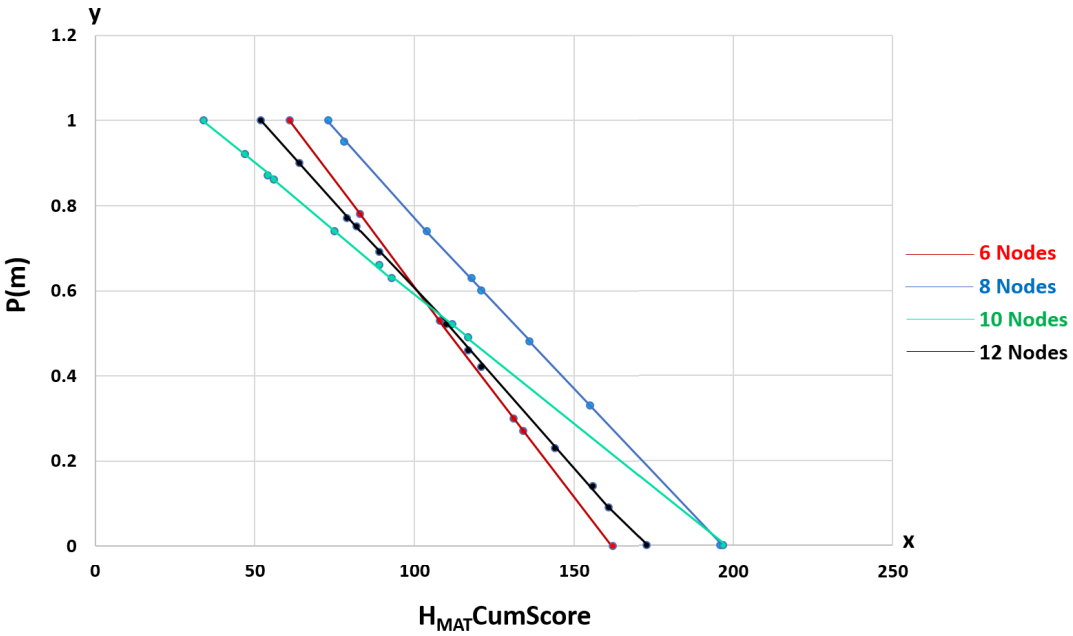

Figure 6.5: Probability of a node being malicious

addition operation and a logical match operation.

d. **Honesty.** As per (6.7), the probability of a node being malicious is less than 0.5 if its $H_{MAT}CumScore > H_{MAT}Threshold$. Moreover, as shown in Figure-6.5, the experimental results show that the probability of a node being malicious (as discussed in Section 6.2.3) is

linear with respect to its $H_{MAT}CumScore$. Also, this linearity is independent of the number of nodes in the network. Correspondingly, the lower the $H_{MAT}CumScore$ of a node is, the higher is the probability of the node being malicious. Similarly, to raise the criteria of a node being honest and to be included in the list of the K honest nodes, the threshold can be raised so that only the nodes with low probability of being malicious, e.g., 0.4, or 0.3, are eligible to be selected as the primary or secondary block proposers.

Moreover, to motivate the nodes to continuously perform honestly and achieve maximum $H_{MAT}CumScore$, the block reward/TX fee is distributed proportionally among all the K honest nodes as per their ranking in the list, i.e., the node with highest $H_{MAT}CumScore$ gets the maximum share, and the node with the lowest $H_{MAT}CumScore$ gets the smallest share. Hence, the nodes strive to achieve maximum $H_{MAT}CumScore$ to get the maximum share of the block reward.

e. **Termination and Agreement.** As shown in Figure-6.6a, Pledge protocol assures that under normal circumstances, when the primary (N_1), and secondary (N_2) block proposers propose the block following the protocol/TX validation rules and the block is agreed upon by all the other nodes, the consensus process terminates. This property holds even if N_1 fails to propose a valid block at the first instance. Besides, considering the adversary's power to disrupt the communications and split the network, Pledge can still perform with consistency. In this context (as shown in Figure-6.6b), to continue the consensus process, the network half comprising those honest nodes that have collectively proposed more blocks than the other half in the last R (eleven in this case) consensus rounds, continues to submit new blocks. Whereas, the other network half waits and synchronize its chain once the network topology is restored. It is imperative to mention that even if the network splits, still both the network halves can get the information about block proposers of the last eleven rounds (before split) from their copy of the blockchain. Moreover, in another scenario, a node may get delayed blocks due to network latency. In such a case, to avoid forks and to protect against the false invalidation of legitimate blocks, the network nodes always wait for the block that points to the block with the highest index in their local chain. Therefore, even if a node receives some blocks in random order, it will append the blocks to its local copy of the chain based on their index number in ascending order. It is also essential to mention that a single node or even a few nodes receiving delayed blocks due to poor network conditions do not affect the operation of the blockchain network irrespective of the duration of the network delays.

f. **BFT.** It is expected that the faulty/malicious nodes may not follow the protocol specifications and behave erratically. Pledge reduces the possibility of Byzantine behavior by a node during the consensus process by putting the node's integrity at stake. Hence, if a node proposes a block with invalid TXs, it is banished, and removed from the list of K honest nodes. Moreover, requisite clarification and corrective action is sought from the owner organization. Also, as a deterrence to others, the responsible organization is banned from participating in the consensus process for seventy-two hours, thus losing valued share of the TX fees. Depending upon the nature of the blockchain network, the organization may also be issued with a finan-

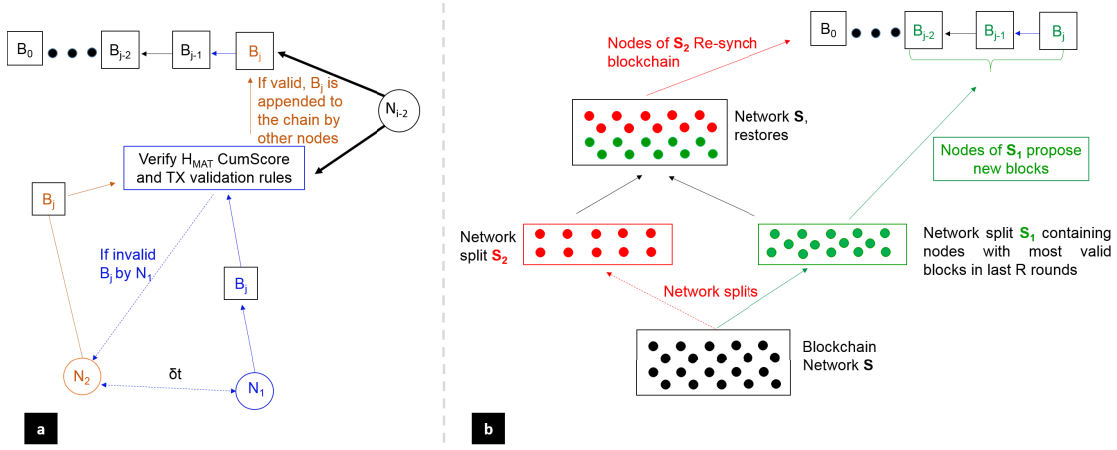

Figure 6.6: Consensus termination and block agreement, a) Normal scenario. b) Split network

cial penalty (mechanism of issuing a financial penalty is not covered in this work). Moreover, the non-performing node's software is re-installed, and it is also reconfigured to get rid of any software bug or malicious payload. Besides, the context-aware TX validation rules introduced in Section-6.2.4 let the nodes easily detect any malicious change in the value of a particular sensor during TX validation before committing a new block.

Moreover, as the block proposer is selected from a list of K honest nodes, the likelihood of participation of malicious/non-performing nodes in the block proposal and consensus process, i.e., validation of proposed blocks, is reduced to a great extent. Hence, it is presumed that till the time there are at least two honest nodes in the list of K honest nodes, the consensus process is safe from most of the faults.

g. **Unforgeability.** The $Honesty_{Mat}$ is computed based on attributes obtained from the blockchain. Hence, due to the distributed and immutable nature of blockchain, peers/nodes cannot emulate attributes or forge any change in respective $H_{Mat}CumScore$. Similarly, Pledge is resilient to forged trust where malicious users may create fake IDs to create a spam farm to boost their trust ratings.

h. **Sybil Attack.** The participation of only registered nodes in the consensus process based on $H_{MAT}CumScore$ reduces the risk of a Sybil attack.

i. **Targeted Attacks.** The selection of K honest nodes based on their bona fide performance and further randomization to select the block proposers avoids targeted attacks by the adversaries against the next deterministic block proposer.

j. **No Trust Issues.** Pledge does not use any P2P reputation or trust model to generate the Honesty Metrics to avoid unfair rating and collusion attacks. Also, Pledge does not rely on a third party, such as an AP or a trusted IDP, for the provision of node attributes. Instead, the attributes for the computation of $Honesty_{Mat}$ are directly obtained from the blockchain. The idea of generating, storing, and extracting attributes from blockchain has the potential to avoid most of the trust issues concerning acquisition of attributes [420, 428–431]. Similarly, Pledge also avoids some of the significant attacks against reputation systems, including discrimination [432], traitors

[433], and slandering attacks [420].

k. **Whitewashing Attack.** It is very likely that Pledge contains the effects of a Whitewashing attack [420], i.e. when the honesty score of a node becomes very low, he leaves the network and then joins later with a new pseudonym. Although acquiring a new pseudonym requires approval in a consortium blockchain, however, still to prevent an insider attack, Pledge provides a disincentive to the nodes for rejoining the network with a new ID. Therefore, when a new node joins the network, his honesty score is below $H_{MAT}Threshold$, due to lack of performance in the system. Thus, a malicious node stands no chance of being included in the list of K eligible nodes. Hence, the nodes with low honesty scores have no option other than to improve their performance and keep their attributes as per the required standard/threshold. However, there is a possibility that with the help of an inside attacker, a malicious node is successful in getting into the list of the K eligible nodes, and randomly gets elected for the block proposal. To counter such eventualities, whenever a primary block proposer broadcasts a new block, all the other honest nodes (in the list of K nodes) verify that whether the block proposer's $H_{MAT}CumScore$ was computed legitimately or not. In case the primary proposer is found to be malicious, the block proposed by the primary is rejected, and the secondary proposer's block is validated and accepted in the same way.

l. **Protection against Non-Performing Nodes.** Another vital aspect is the accountability of the nodes that violate the consensus rules or fail to surpass $H_{MAT}Threshold$. It is envisaged that the nodes that fail to get into the list of K eligible nodes for time δt equivalent to the duration of a number Z of consecutive published blocks, they are blacklisted. Where Z depends upon the sensitivity/criticality of the system. Hence, if the system failure has serious security or safety implications, then Z can be set as the lowest as possible. E.g., if a node fails to get into the list of K eligible block proposers for eleven consecutive blocks, it will be blacklisted. Similarly, if a node's conduct is erratic and it performs below the threshold in between the episodes of making into the group of eligible block proposers, such a node's behavior is measured by analyzing node's last eleven $H_{MAT}CumScores$. If it has secured below threshold score for six or more times (this can change depending upon the criticality/sensitivity of the IoT system), it is blacklisted.

m. **Replay Attacks.** To protect against replay/double-spending attacks, every TX initiated by a particular node/client application has a sequence number in addition to the timestamp. Hence, a particular node cannot generate another TX with a higher timestamp but a lower or same sequence number as the previous one.

n. **Decentralization.** The random selection of a primary and a secondary block proposer protects the system from quasi-centralization. Otherwise, few most honest nodes may have the monopoly to mine every new block, and they may try to play foul with the system. Instead, the system gets more decentralized as the network expands. It is because the list of K eligible nodes is likely to extend with more number of nodes satisfying the threshold $H_{MAT}CumScore$. Hence, the probability of a node to be selected as a primary or a secondary block proposer decreases with the increase in the number of nodes in the list of K nodes. However, preventing

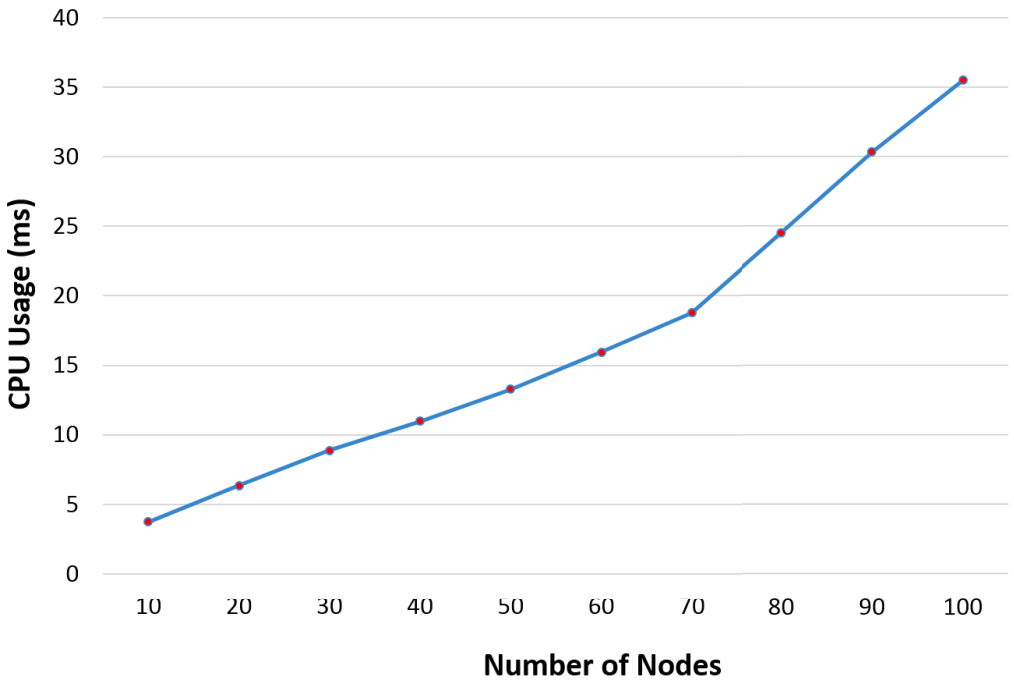

Figure 6.7: Avg CPU time to execute Pledge protocol vs Number of nodes

quasi-centralization has an associated risk, i.e., the random selection of two block proposers from a list of K eligible nodes based on the threshold score entails selection of those nodes that have the probability of being malicious equal to 0.5. However, depending upon the sensitivity of the IoT application, the threshold can be raised to decrease the probability of selecting a possibly malicious node. Similarly, for systems that are not concerned about quasi-centralization, the primary and the secondary block proposers can always be selected from the top x% of the nodes with the highest $H_{MAT}CumScore$.

Table 6.3: Avg difference in avg CPU usage

Set of Nodes	**Difference in CPU Usage (ms)**
10-20	2.60
20-30	2.52
30-40	2.12
40-50	2.99
50-60	2.67
60-70	2.83
70-80	5.74
80-90	5.81
90-100	5.13
Avg	**3.60**

o. **Scalability.** The proposed scheme does not require energy and computationally intensive PoW for the selection of a block proposer. The computation of the $Honesty_{Mat}$ requires meager

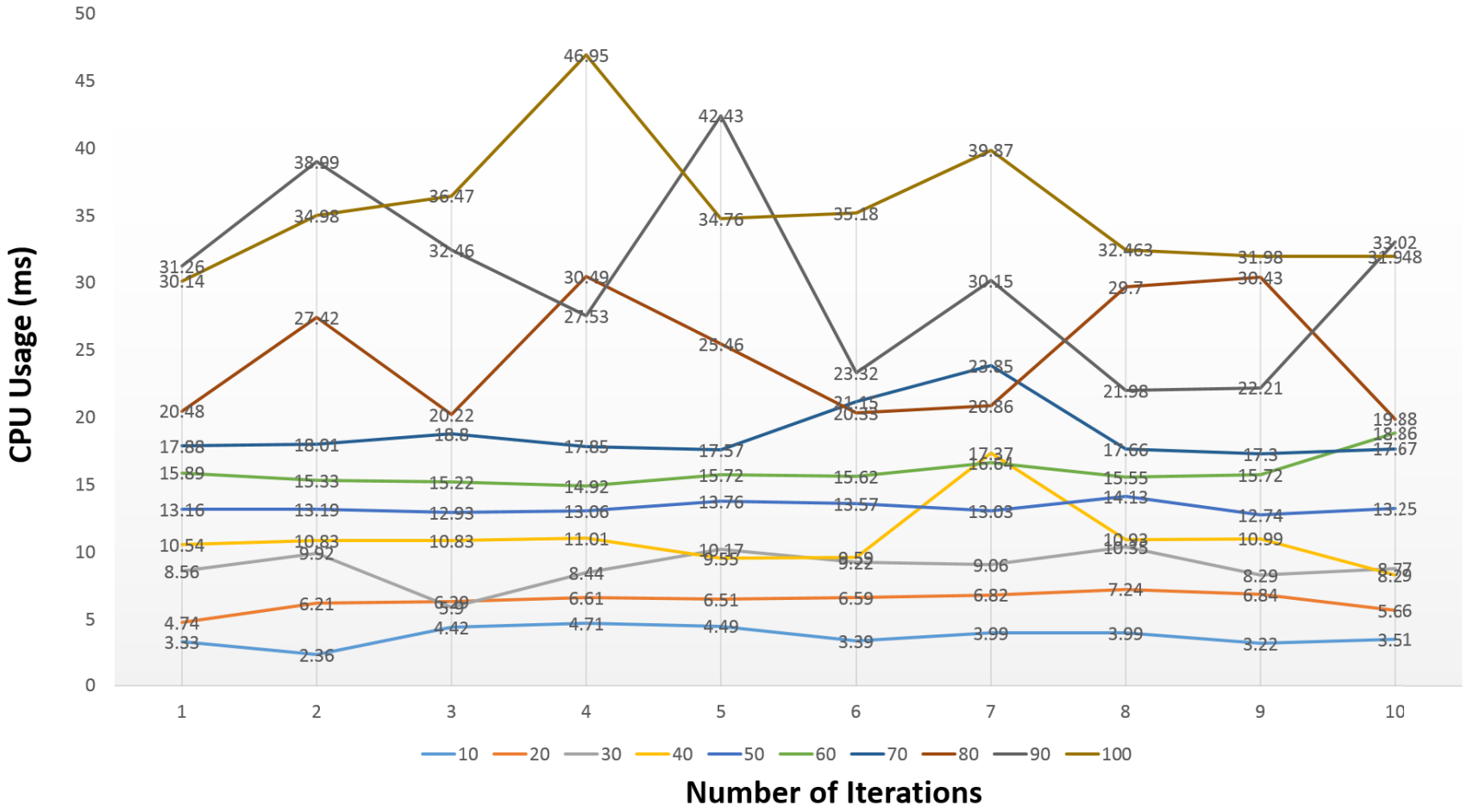

Figure 6.8: Ten iterations of CPU usage measurement vs Number of nodes

Table 6.4: Security and performance comparison of consensus protocols

Features	**PoW**	**PoS**	**PoET**	**PBFT**	**IoTA**	**PoH**
Area of application	Fintech	Fintech	Multiple	Multiple	Multiple	Multiple
Type of Blockchain	Permissionless	Permissionless and Permissioned	Permissionless and Permissioned	Permissioned	Permissionless	Currently Permissioned
Vulnerabilities	51% attack	51% attack, and malicious collusion of rich stakeholders	Node compromise	Fault tolerance of 1/3 faulty nodes, and DoS attack	vulnerability of Curl-P-27 hash function, and signature forging attacks [434]	Low $H_{MAT}Threshold$ at the start of the network, hence, probability of a node being selected as a primary or secondary block proposer is high with less number of nodes in the network
Address nothing at stake problem	No	No	No	No	No	Yes
Energy costs	High	Low	Low	Low	Low	Low
Computation costs	High	Low	Low	Low	Low	Low
Communication complexity	Low	Low	Low	High	Low	Low
Consensus Finality	Probabilistic	Probabilistic	Probabilistic	Instant	Probabilistic	Instant
Blockchain Forks	Yes	Yes	Yes	No	Yes	No
TX latency (Based on consensus finality)	High	Low	Low	Low	Moderate	Low
Scalable	Yes	Yes	Yes	Poor scalability concerning the number of validating nodes	Yes	Yes
The requirement of special hardware	Yes (mostly for mining)	No	Yes	No	No	No

resources. Moreover, the Pledge does not require excessive communication rounds to vote on the eligibility of the blocks or to propagate reputation scores between peers. Hence, there are no communication overheads other than routine TX and block propagation messages. Moreover, Pledge is scalable with an increase in the number of nodes as the list of K eligible nodes is dynamic. It includes all the nodes that have $H_{MAT}CumScore$ greater than or equal to the threshold. Hence, the increase in the number of nodes does not affect the performance of the consensus process. Moreover, the logic for selecting a primary and a secondary block proposer is that in case the primary block proposer fails to propose a block in time ζt, then to avoid latency in TX confirmation by starting the process all over again, the secondary node proposes the block.

Additionally, to measure the latency in TX confirmation, we measured the avg CPU time for the execution of Pledge protocol (including computation of $Honesty_{Mat}$, and selection of Primary and Secondary block proposers) for a range of nodes varying from ten to hundred. We had total hundred iterations of the experiment for ten different sets of nodes. As shown in Figure-6.7, although the avg CPU time rises with the increase in the number of nodes, yet for a hundred nodes, the computation time is merely 35.47 ms. Similarly, Figure-6.8 shows the ten iterations of CPU usage measurement for each set of ten, twenty, thirty, forty, fifty, sixty, seventy, eighty, ninety, and hundred nodes. It can be seen that there are many variations in the CPU usage for each set of nodes once the number of nodes goes above seventy. Correspondingly, based on the avg difference between avg CPU usage for each set of nodes (shown in Table-6.3), it is estimated that even if the number of potential block proposers increases to two thousand (which is very unlikely in a consortium/private blockchain) the latency in TX confirmation is expected to be 720 ms, which is still under one sec. Hence, it can be concluded that for a private/consortium blockchain settings, the latency in TX confirmation is very nominal and Pledge performs better than Bitcoin (TX confirmation is after 2-6 blocks, i.e., 10-60 mins) [435], IoTA (No specific time as it varies from 2-3 mins to even 30 mins depending upon the rate of input of new TXs in the network) [436], and Ethereum (15 sec) [435]. It is also inferred that a block proposed by an honest node, once accepted by the other honest nodes, would not be later purged from the chain. Correspondingly, a detailed comparison of the security and performance efficiency of PoH versus some renowned consensus protocols is shown in Table-6.4.

6.3.1 Limitations and A Way Forward

In addition to the security guarantees, we have also observed certain limitations of the Pledge protocol, that require further research.

- $H_{MAT}Threshold$ vs. Network Bootstrapping: The current selection of $H_{MAT}Threshold$ as the avg value of $H_{MAT}Max$ and $H_{MAT}Min$ scores seems workable once the blockchain network is running for some time. However, it is observed that in the current form, the avg value may not provide the desired security once the network is being bootstrapped. Because

at the start of the blockchain network, all the block proposers will have almost the same $H_{MAT}Min$ score. Hence, the selection of the block proposer will rely upon random selection from the list of K eligible nodes for quite some time. Therefore, there is a requirement of working out an appropriate value of $H_{MAT}CumScore$ for validator nodes, as a starting point. One option in this regard may be a random allocation of $H_{MAT}CumScore$ at the start to bootstrap the network.

- Adding a New Node to the Pledge Consensus: Currently, there is a question mark on how to onboard a new node into the honesty-based consensus protocol. It is perceived that it would take a long time for a new node to catch up with other nodes that already have a high honesty score.

- Selection and Scrapping of Block Proposers' Attributes From Blockchain: At the moment, only eleven attributes (listed in Table-6.1) have been identified, which seems common to most of the blockchain technologies. However, in practice, there would be a requirement of extracting those attributes from the blockchain that best describe the integrity and the performance of the block proposer nodes in a specific blockchain network. Therefore, it is envisaged that this aspect has to be catered for while developing a blockchain platform so that the desired attributes can be directly measured from the blockchain using inbuilt functions, e.g., methods/functions available in web3.js library to interact with Ethereum Blockchain.

6.4 Summary

In this chapter, we proposed "Pledge," a unique Proof-of-Honesty (PoH) based block proposer selection protocol that incorporates an IoT-centric TX validation scheme. Pledge reduces the probability of participation by non-performing, and potentially Byzantine nodes in the consensus process by restricting the block proposal responsibility to a couple of honest nodes in the network. It also prevents Sybil attack, avoids quasi-centralization, and averts various attacks against the reputation systems. Pledge is currently designed for consortium blockchains. However, with requisite modifications, it can be deployed in public blockchains as well. Based on our initial experiments and analysis, it is ascertained that Pledge not only satisfies most of the security requirements discussed in this chapter but is also computationally efficient with an insignificant change in communications overhead.

”The blockchain symbolizes a shift in power from the centers to the edges of the networks.”

- Warren Buffet

7

Conclusions and Future work

This chapter concludes the thesis and provides some direction for future work. The thesis aims to devise a defense strategy for integrity attacks on the IoT. The integrity attacks include any attack that threatens the purity of data. Similarly, security, privacy and availability of data are also essential requirements. Therefore, to find the exact nature of the IoT threat spectrum, a comprehensive study of IoT threats is carried out. This research helped us in exploring the methodology of IoT threats and conceiving a defense-in-depth strategy. While formulating security solutions, blockchain was identified as a potent defense tool to guard against most of the data integrity threats. However, initially being developed for financial technology, blockchain required a thorough evaluation of its adoption in IoT. Hence, a systematic study of the progression in blockchain technology and its impact on IoT is a major contribution of this thesis. It is followed by the introduction of a privacy-preserving and secure data sharing framework for smart cities. The proposed solution enables the data owners to control access to their data based on user-defined ACL rules embedded in blockchain smart contracts. The secure data sharing mechanism also complies with some of the significant EU GDPR requirements. However, no blockchain-based framework is secure and efficient without an appropriate consensus protocol. Therefore, another significant contribution of this research is the conception and design of an IoT-oriented consensus protocol with IoT-centric TX validation rules. To get a more holistic view of the research contribution of this study, a chapter-wise recap of important conclusions is delineated in succeeding paras.

7.1 Summary of the Thesis

7.1.1 Chapter 2

This chapter highlights a wide range of generalized as well as numerous threats specific to various layers of IoT architecture. The comprehensive literature review helped in deriving a comprehensive attack methodology adopted by most of the successful IoT malware attacks. We also presented the structure and procedure of an IoT-botnet based DDoS attack. The illustrious study of IoT threat spectrum enabled us to draw some important conclusions:

a. Due to the lack of standardization of IoT, most of the products are manufactured with a focus on performance rather than security.
b. Cryptographic security provided by IoT communications protocols is not enough to protect devices against physical compromise, remote code execution, and other integrity attacks.
c. Cyber attacks are considered to be one of the major threats to IoT applications, and mostly the network and the application layers are the focus of the attackers.
d. No operation in IoT can be termed safe unless the integrity of the IoT devices is ensured.
e. Absence of anti-malware mechanisms in the IoT is one of the causes of a successful compromise of IoT devices.
f. Cloud-supported IoT systems are vulnerable to a single point of failure, and threats to the security, privacy and integrity of data.

7.1.2 Chapter 3

In this chapter, we present a defense-in-depth strategy as a guideline to form a composite IoT security framework. The proposed security approach is formulated by reviewing and systematically integrating current industry best practices on IoT security. Hence, a comprehensive security mechanism comprising protective, detective, responsive, and corrective measures is proposed. Some vital lessons learned from this study include:

a. Standard IT security protocols cannot be deployed on resource-constraint IoT devices.
b. Security has to be viewed as a whole, and numerous supplementary measures need to be taken at different layers of IoT architecture.
c. There should always be a cost-benefit analysis while designing a secure and efficient IoT application.
d. Not all the IoT technologies/security protocols meet the needs of all possible IoT use cases. Instead, all technologies have adequate security for some specific IoT applications. Hence, IoT service providers and manufacturers need to have a clear understanding of what security features are required for which IoT use cases.
e. More research is required to ensure the security and privacy of user data during intra-cloud data processing and analytics.

f. IoT operational model needs to be transformed from a costly, trusted, and over-arched centralized architecture to a self-regulating, and self-managed decentralized model.

g. Blockchain can be the IoT savior to protect against a wide range of data integrity, privacy, and availability attacks.

7.1.3 Chapter 4

This part of the thesis carries out an in-depth study of blockchain technology and identifies some practical challenges to its adoption in IoT. The gaps are identified by mapping some peculiar security and performance requirements of IoT systems over the benefits inferred by the blockchain technologies. Some important takes from this research are:

a. IoT is the future of an autonomous digitized world, and to achieve this, IoT has to undergo a conceptual transformation at all stages, i.e., design, development and operation.

b. Due to its inherent cryptographic security, blockchain can protect IoT from data manipulation, and forgery attacks. It can also provide a trustless operational environment with decentralized control and user-defined access to data.

c. By leveraging the smart contract feature of the blockchain, numerous operations in an IoT system can be securely and autonomously performed.

d. Device integrity is the key requirement to ensure secure integration of blockchain with an IoT system.

e. There is a critical requirement of an IoT-oriented consensus protocol with IoT-focused TX/block validation rules.

f. Any blockchain-based IoT system should cater to constrained resources of IoT devices while tackling various security and performance issues concerning blockchain, such as data privacy, user anonymity, scalability related to blockchain size and latency in TX confirmation.

7.1.4 Chapter 5

Based upon blockchain's evaluation for IoT in Chapter-4, we have introduced a blockchain-based framework for privacy-preserving and secure data sharing, in this chapter. The proposed scheme preserves data privacy by dividing the blockchain network into various data specific Chs. Moreover, data security is further augmented by using a technique of private data collection. Also, to achieve data confidentiality, users have the option to encrypt their data. The primary hallmark of PrivySharing is empowering data owners to define access control rules concerning their data. Users are also at liberty to purge their data assets when they are no longer required. Some important findings from this research are as under:

a. A sensibly selected and carefully designed blockchain-based IoT application can provide some assurance to the users concerning the security and privacy of their data.

b. Scalability related to blockchain's size is an open challenge, especially for resource-constrained IoT devices.

c. PrivySharing, based on Hyperledger Fabric blockchain, exhibited faster TX confirmation time for all types of transactions, including plain text, private data, and encrypted, as compared to Bitcoin, Ethereum, and IoTA.

d. Three-Ch blockchain scenario outperformed a single-Ch blockchain setting by achieving a TX throughput of 42.4 TPS with an average latency of 1.54 sec at the TX send rate of 50 TPS.

e. In Hyperledger Fabric, the TX throughput, and latency in TX confirmation depend upon network architecture, consensus protocol, specifications of endorsement policies, complexities of ACL rules, chaincode language, number of endorsers and orderer nodes, batch timeout, and the physical/virtual network infrastructure.

f. Mobile edge computing technology can be integrated with the blockchain to relieve resource constraint IoT devices from storing and maintaining multiple distributed ledgers.

7.1.5 Chapter 6

This chapter introduces, "Pledge," a PoH based IoT-centric consensus protocol with IoT-oriented TX validation rules. The proposed consensus protocol aims at reducing the probability of participation by faulty, non-performing, and malicious/Byzantine nodes in the consensus process. The proposed consensus mechanism selects two block proposers from a list of K most honest nodes, thus avoiding Sybil attack, quasi-centralization, and numerous attacks against reputation systems. Based on the initial experimental results and analysis, it is ascertained that Pledge not only satisfies most of the requirements of an ideal consensus protocol for the IoT systems but is also scalable with computational and communications economy.

7.2 Future Research

No research is deemed complete in itself. Instead, every study unveils new avenues of research. Similarly, some open research issues identified in this thesis are enumerated as under:

a. Design and development of a dynamic anti-malware mechanism for resource constraint IoT devices.

b. Because of their critical functionalities, most of the IoT devices remain in continuous operation without any firmware or software updates. Hence, they are more vulnerable to cyber-attacks. Therefore, there is a need for a runtime firmware/software upgrading/updating mechanism. Similarly, due to the decentralized architecture of the blockchain, currently, there is no mechanism to ensure synchronized software upgrades in the end devices.

c. A scalable and economical blockchain platform is required to cater for the needs of all types of IoT devices and applications.

d. Currently, there is no mechanism to perform a real-time IoT device integrity check to ensure secure input of data from an end-device to the blockchain.

e. PoH-based consensus protocol can be further improved by incorporating an Intelligent Misbehavior Detection mechanism so that the $Honesty_{Mat}$ can be computed for a miner/validator node in any blockchain platform. Similarly, an extended security and performance analysis of Pledge will be helpful in the development of an optimum consensus protocol for IoT environment, tested in all aspects.

BIBLIOGRAPHY

[1] N. Cam-Winget, A. R. Sadeghi, and Y. Jin, "Can IoT be secured: Emerging challenges in connecting the unconnected," in *Proceedings of the 53rd ACM/EDAC/IEEE Design Automation Conference (DAC)*, TX, USA, June 2016, pp. 1–6. 1, 20, 23

[2] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, and A. Marrs, *Disruptive technologies: Advances that will transform life, business, and the global economy*. McKinsey Global Institute, CA, USA, 2013, vol. 180. 1

[3] D. Lund, C. MacGillivray, V. Turner, and M. Morales, "Worldwide and regional Internet of Things (IoT) 2014–2020 forecast: A virtuous circle of proven value and demand," International Data Corporation (IDC), MA, USA, Tech. Rep. 1, 2014. 1

[4] D. Evans, "The Internet of Things: How the next evolution of the internet is changing everything," *CISCO*, vol. 1, no. 2011, pp. 1–11, 2011. 1

[5] "Computer virus strikes CSX transportation computers," 2003. [Online]. Available: http://www.prnewswire.com/news-releases/computer-virus-strikes-csx-transportation-computers-70971537.html 1

[6] K. Poulsen, "Slammer worm crashed Ohio nuke plant network," *Security Focus*, vol. 19, 2003. 1

[7] A. Greenberg, "Hackers remotely kill a jeep on the highway - With me in it," *Wired*, vol. 7, pp. 1–21, 2015. 1

[8] S. A. Kumar, T. Vealey, and H. Srivastava, "Security in Internet of Things: Challenges, solutions and future directions," in *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS)*. HI, USA: IEEE, Jan 2016, pp. 5772–5781. 1, 9, 11, 12, 15, 20, 21, 22, 25, 62

[9] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial Internet of Things," in *Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, CA, USA, June 2015, pp. 1–6. 1, 15, 16, 20, 21, 25, 62, 72

[10] T. Borgohain, U. Kumar, and S. Sanyal, "Survey of security and privacy issues of Internet of Things," *arXiv preprint arXiv:1501.02211*, pp. 1–7, 2015. 1, 2, 9, 11, 15, 21, 25, 72

[11] J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eyers, "Twenty security considerations for cloud-supported Internet of Things," *IEEE Internet of Things Journal*, vol. 3, pp. 269–284, 2015. 1, 28, 50

[12] M. Andrew, "How the Internet of Things will affect security& privacy," 2016. [Online]. Available: http://www.businessinsider.com/internet-of-things-security-privacy-2016-8?IR=T 1

[13] J. Steinberg, "These devices may be spying on you (Even in your own home)," 2014. [Online]. Available: https://www.forbes.com/sites/josephsteinberg/2014/01/27/these-devices-may-be-spying-on-you-even-in-your-own-home/#73cc4556b859 1

[14] "Internet of Things security study: Smart watches," HP, Tech. Rep., 2017. [Online]. Available: http://go.saas.hpe.com/fod/internet-of-things 1

[15] T. Hahn, S. Matthews, L. Wood, J. Cohn, S. Regev, J. Fletcher, E. Libow, C. Poulin, and K. Ohnishi, "IBM point of view: Internet of Things security," *IBM White Paper*, 2015. [Online]. Available: http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=RAW14382USEN 2, 48, 54, 58

[16] S. Darlene, "SCADA Strangelove: Zero-days & hacking for full remote control," 2015. [Online]. Available: http://www.computerworld.com/article/2475789/cybercrime-hacking/hackers-exploit-scada-holes-to-take-full-control-of-critical-infrastructure.html 2

[17] "The CEO's guide to data security. Protect your data through innovation - AT&T Cybersecurity Insights (Vol 5)," pp. 1–20, 2016. [Online]. Available: https://www.business.att.com/cybersecurity/docs/vol5-datasecurity.pdf 2, 10, 16, 19, 30, 46, 47, 48, 49, 50, 54, 57, 64

[18] D. Paul, "Mirai Internet of Things malware from Krebs DDoS attack goes open source," 2016. [Online]. Available: https://nakedsecurity.sophos.com/2016/10/05/mirai/ 2, 82

[19] R. Brewer, "Ransomware attacks: detection, prevention and cure," *Network Security*, vol. 2016, no. 9, pp. 5–9, 2016. 2, 32, 34

[20] E. Kovacs, "Shamoon attacks possibly aided by Greenbug Group," 2017. [Online]. Available: http://www.securityweek.com/shamoon-attacks-possibly-aided-greenbug-group 2

[21] Pierluigi Paganini, "Duqu2.0: The most sophisticated malware ever seen," *Infosec*, 2015. [Online]. Available: http://resources.infosecinstitute.com/duqu-2-0-the-most-sophisticated-malware-ever-seen/#gref 2

[22] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, "Threats to networking cloud and edge datacenters in the Internet of Things," *IEEE Cloud Computing*, vol. 3, no. 3, pp. 64–71, 2016. 2, 20, 21, 22, 25, 29, 59, 86, 122

[23] P. Brody and V. Pureswaran, "Device democracy: Saving the future of the Internet of Things," *IBM*, pp. 1–28, Sep 2014. 2

[24] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *White Paper*, pp. 1–9, 2008. 2, 3, 48, 54, 65, 74, 75, 77, 80, 88, 89, 90, 95, 119, 159

[25] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016. 2, 70, 71, 79

[26] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of threats to the Internet of Things," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1636–1675, Secondquarter 2019. 2, 9, 43, 156

[27] S. Sara and N. Michael, "Facebook has been worried about data leaks like this since it went public in 2012,," *CNBC*, 2018. [Online]. Available: https://www.cnbc.com/2018/04/12/facebook-warned-of-data-breaches-years-ago-when-it-went-public-in-2012.html 3, 59, 87, 122

[28] S. Jason, "Hundreds of millions of Facebook user records were exposed on Amazon cloud server,," *CBS News*, 2019. [Online]. Available: https://www.cbsnews.com/news/millions-facebook-user-records-exposed-amazon-cloud-server/ 3, 122

[29] S. Sara, "A Google bug exposed the information of up to 500,000 users,," *CNBC*, 2018. [Online]. Available: https://www.cnbc.com/2018/10/08/google-bug-exposed-the-information-of-up-to-500000-users.html 3, 122

[30] E. Bertino, "Data security and privacy: Concepts, approaches, and research directions," in *Proceedings of the 40th IEEE Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2016, pp. 400–407. 3

[31] "Cryptocurrency market capitalizations," 2019. [Online]. Available: https://coinmarketcap.com/ 3

[32] M. Ahlmeyer and A. M. Chircu, "Securing the Internet of Things: A review," *Issues in Information Systems*, vol. 17, no. 4, 2016. 9, 10, 11, 14, 15

[33] M. Abomhara and G. M. Køien, "Security and privacy in the Internet of Things: Current status and open issues," in *Proceedings of the IEEE International Conference on Privacy and Security in Mobile Systems (PRISMS)*, Aalborg, Denmark, May 2014, pp. 1–8. 9, 10, 11, 18, 20, 65

[34] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015. 9, 10, 11, 12, 17, 21, 25, 39, 40

[35] V. Sivaraman, H. H. Gharakheili, A. Vishwanath, R. Boreli, and O. Mehani, "Network-level security and privacy control for smart-home IoT devices," in *Proceedings of the 11th IEEE*

International Conference onWireless and Mobile Computing, Networking and Communications (WiMob), Abu Dhabi, UAE, Oct 2015, pp. 163–167. 9, 10, 11, 27, 51, 55

[36] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the Internet of Things: Perspectives and challenges," *Wireless Networks*, vol. 20, no. 8, pp. 2481–2501, 2014. 9, 10, 11, 14, 18, 73, 94

[37] A. Banafa, "IoT standardization and implementation challenges," *IEEE Internet of Things, Newsletter*, July 2016. [Online]. Available: http://iot.ieee.org/newsletter/july-2016/iot-standardization-and-implementation-challenges.html 10, 13, 64

[38] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015. 12

[39] M. Khari, M. Kumar, S. Vij, P. Pandey, and Vaishali, "Internet of Things: Proposed security aspects for digitizing the world," in *Proceedings of the 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, Mar 2016, pp. 2165–2170. 12, 18, 19

[40] A. Reziouk, E. Laurent, and J.-C. Demay, "Practical security overview of IEEE 802.15.4," in *Proceedings of the IEEE International Conference on Engineering & MIS (ICEMIS)*, Agadir, Morocco, Sept 2016, pp. 1–9. 12, 17, 21, 25, 39

[41] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010. 12

[42] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: The Internet of Things architecture, possible applications and key challenges," in *Proceedings of the 10th IEEE International Conference on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan, Dec 2012, pp. 257–260. 12

[43] Z. Yang, Y. Yue, Y. Yang, Y. Peng, X. Wang, and W. Liu, "Study and application on the architecture and key technologies for IoT," in *Proceedings of the IEEE International Conference on Multimedia Technology (ICMT)*, Hangzhou, China, July 2011, pp. 747–751. 12

[44] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du, "Research on the architecture of Internet of Things," in *Proceedings of the 3rd IEEE International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 5, Chengdu, China, Aug 2010, pp. V5–484–V5–487. 12

[45] L. Tan and N. Wang, "Future internet: The Internet of Things," in *Proceedings of the 3rd IEEE International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 5, Chengdu, China, Aug 2010, pp. V5–376–V5–380. 12

[46] M. A. Chaqfeh and N. Mohamed, "Challenges in middleware solutions for the Internet of Things," in *Proceedings of the IEEE International Conference on Collaboration Technologies and Systems (CTS)*, CO, USA, May 2012, pp. 21–26. 12

[47] D. Uckelmann, M. Harrison, and F. Michahelles, "An architectural approach towards the future Internet of Things," in *Architecting the Internet of Things*. Heidelberg, Germany: Springer, 2011, pp. 1–24. 12

[48] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu, "Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the Internet-of-Things," in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, PA, USA, Nov 2015, pp. 1–5. 14, 56, 57

[49] "HPE Fortify and the Internet of Things," 2017. [Online]. Available: http://go.saas.hpe.com/fod/internet-of-things 15

[50] H. Ning, H. Liu, and L. T. Yang, "Cyberentity security in the Internet of Things," *Computer*, vol. 46, no. 4, pp. 46–53, 2013. 15, 20

[51] R. M. Savola, H. Abie, and M. Sihvonen, "Towards metrics-driven adaptive security management in e-health IoT applications," in *Proceedings of the 7th International Conference on Body Area Networks*. Oslo, Norway: Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering (ICST), Feb 2012, pp. 276–281. 15, 20, 21, 25, 52, 55

[52] M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: Models, methods, and metrics," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1283–1295, 2014. 15, 20, 62

[53] Tim Hahn and JR Rao, "IoT security: An IBM position paper," *Watson IoT*, pp. 1–21, 2016. [Online]. Available: https://www.iotca.org/wp-content/themes/iot/pdf/resources-page/iotca-resources-ibm-white-paper.PDF 15, 20, 46, 47, 50, 54, 55, 56, 57, 64

[54] Paul Ducklin, "Mirai Internet of Things malware from Krebs DDoS attack goes open source," *Naked Security by Sophos*, 2016. [Online]. Available: https://nakedsecurity.sophos.com/2016/10/05/mirai-internet-of-things-malware-from-krebs-ddos-attack-goes-open-source/ 15, 34, 50, 91, 114

[55] A. Burg, A. Chattopadhyay, and K.-Y. Lam, "Wireless communication and security issues for Cyber Physical Systems and the Internet of Things," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 38–60, 2018. 16, 20

[56] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016. 16

[57] M. Guri, G. Kedma, A. Kachlon, and Y. Elovici, "AirHopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies," in *Proceedings of the 9th IEEE International Conference on Malicious and Unwanted Software: The Americas (MALWARE)*, PR, USA, Oct 2014, pp. 58–67. 16, 20

[58] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *Proceedings of the IEEE Symposium on Security and Privacy*, CA, USA, May 2008, pp. 129–142. 16

[59] P. Schneider and G. Horn, "Towards 5G security," in *Proceedings of the IEEE Trustcom/BigDataSE/ISPA*, vol. 1, Helsinki, Finland, Aug 2015, pp. 1165–1170. 16

[60] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *ICT Express*, vol. 3, no. 1, pp. 14–21, 2017. 16, 17, 60, 63, 94

[61] M. Chen, Y. Miao, Y. Hao, and K. Hwang, "Narrow band Internet of Things," *IEEE Access*, vol. 5, pp. 20 557–20 577, 2017. 17, 61

[62] F. Koushanfar, A.-R. Sadeghi, and H. Seudie, "EDA for secure and dependable cybercars: Challenges and opportunities," in *Proceedings of the 49th ACM Annual Design Automation Conference*, CA, USA, June 2012, pp. 220–228. 17

[63] D. L. Lough, "A taxonomy of computer attacks with applications to wireless networks," Doctoral Dessertation, Department of Electrical and Computer Engineering, Virginia Tech, VA, USA, 2001. 17, 20

[64] M. Vanhoef and F. Piessens, "Advanced Wi-Fi attacks using commodity hardware," in *Proceedings of the 30th ACM Annual Computer Security Applications Conference*, NY, USA, Dec 2014, pp. 256–265. 17

[65] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-Low Latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 88–145, 2019. 17

[66] T. Mizrahi, E. Grossman, A. J. Hacker, S. Das, J. Dowdell, H. Austad, K. Stanton, and N. Finn, "Deterministic Networking (DetNet) security considerations," Internet Engineering Task Force, Internet-Draft draft-ietf-detnet-security-02, 2018, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-detnet-security-02 17

[67] M. Furdek, N. Skorin-Kapov, S. Zsigmond, and L. Wosinska, "Vulnerabilities and security issues in optical networks," in *Proceedings of the 16th IEEE International Conference on Transparent Optical Networks (ICTON)*, Graz, Austria, July 2014, pp. 1–4. 17, 51

[68] B. Everett, "Tapping into fibre optic cables," *Network Security*, vol. 2007, no. 5, pp. 13–16, 2007. 17

[69] *Alcatel-Lucent 1830 Photonic Service Switch (PSS-64 and PSS-36)*, Alcatel Lucent, 2015. [Online]. Available: http://lightspeedt.com/wp-content/uploads/2015/10/1830-PSS-Datasheet.pdf 17

[70] C. Mas, I. Tomkos, and O. K. Tonguz, "Failure location algorithm for transparent optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 8, pp. 1508–1519, 2005. 17

[71] A. Bononi, P. Serena, N. Rossi, and D. Sperti, "Which is the dominant nonlinearity in long-haul PDM-QPSK coherent transmissions?" in *Proceedings of the 36th IEEE European Conference and Exhibition on Optical Communication (ECOC)*, Turin, Italy, Sept 2010, pp. 1–3. 17

[72] R. Aparicio-Pardo, P. Pavon-Marino, and S. Zsigmond, "Mixed line rate virtual topology design considering nonlinear interferences between amplitude and phase modulated channels," *Photonic Network Communications*, vol. 22, no. 3, pp. 230–239, 2011. 17

[73] B. Gupta and A. Tewari, *A Beginners Guide to Internet of Things Security: Attacks, Applications, Authentication, and Fundamentals*. CRC Press, 2020. 18

[74] G. Hernandez, O. Arias, D. Buentello, and Y. Jin, "Smart nest thermostat: A smart spy in your home," *Black Hat USA*, pp. 1–8, 2014. 18, 20

[75] S. Zonouz, J. Rrushi, and S. McLaughlin, "Detecting industrial control malware using automated PLC code analytics," *IEEE Security & Privacy*, vol. 12, no. 6, pp. 40–47, 2014. 18

[76] Black-Lotus-Labs, "Attack of Things," 2016. [Online]. Available: https://blog.centurylink.com/attack-of-things/ 18, 62

[77] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, 2009. 18

[78] F. B. Thomas, "It's depressingly easy to spy on vulnerable baby monitors using just a browser," 2015. [Online]. Available: https://www.forbes.com/sites/thomasbrewster/2015/09/02/baby-surveillance-with-a-browser/#2508d85b1aa0 20, 22, 25

[79] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: Analysis & defenses," in *Proceedings of the 3rd ACM International Symposium on Information processing in sensor networks*, CA, USA, Apr 2004, pp. 259–268. 20, 22

[80] J. Wurm, K. Hoang, O. Arias, A.-R. Sadeghi, and Y. Jin, "Security analysis on consumer and industrial IoT devices," in *Proceedings of the 21st IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)*, Macau, Jan 2016, pp. 519–524. 20, 22, 62

[81] O. Arias, J. Wurm, K. Hoang, and Y. Jin, "Privacy and security in Internet of Things and wearable devices," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 1, no. 2, pp. 99–109, 2015. 20, 23, 72

[82] "Xbox 360 Timing attack," 2007. [Online]. Available: http://beta.ivc.no/wiki/index.php/Xbox_360_Timing_Attack 20, 23

[83] B. Balamurugan and B. Dyutimoy, "Security in network layer of iot: Possible measures to preclude," in *Security breaches and threat prevention in the Internet of Things*, J. N. and T. R., Eds. IGI Global, 2017, ch. 3, pp. 46–75. 20, 24, 72

[84] S. Skorobogatov, "Fault attacks on secure chips: From glitch to flash," *Design and Security of Cryptographic Algorithms and Devices (ECRYPT II)*, pp. 1–64, 2011. [Online]. Available: https://www.cosic.esat.kuleuven.be/ecrypt/courses/albena11/slides/sergei_skorobogatov_faults.pdf 20, 24

[85] B. Fowler, "Some top baby monitors lack basic security features," *4 New York*, 2015. [Online]. Available: https://www.nbcnewyork.com/news/local/Baby-Monitor-Security-Research-324169831.html 20, 25

[86] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, "A dynamic prime number based efficient security mechanism for big sensing data streams," *Journal of Computer and System Sciences*, vol. 83, no. 1, pp. 22–42, 2017. 20, 21, 25

[87] S. Mohammadi and H. Jadidoleslamy, "A comparison of link layer attacks on wireless sensor networks," *Journal of Information Security*, no. 2, pp. 69–84, 2011. [Online]. Available: https://www.researchgate.net/profile/Reza_Ebrahimi_Atani/publication/310510404_A_Comparison_of_Link_Layer_Attacks_on_Wireless_Sensor_Networks/links/5830d22308ae102f0731cf8c.pdf 20, 21, 25

[88] V. B. Misic, J. Fang, and J. Misic, "MAC layer security of 802.15.4-compliant networks," in *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, DC, USA, Nov 2005, pp. 1–8. 21, 25

[89] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," in *Proceedings of the Proceedings of the 3rd ACM workshop on Wireless security*, PA, USA, Oct 2004, pp. 32–42. 21, 25

[90] R. Riaz, K.-H. Kim, and H. F. Ahmed, "Security analysis survey and framework design for IP connected LoWPAN," in *Proceedings of the IEEE International Symposium on Autonomous Decentralized Systems, ISADS'09.* , Athens, Greece, Mar 2009, pp. 1–6. 21, 25

[91] D. R. Raymond and S. F. Midkiff, "Denial-of-service in wireless sensor networks: Attacks and defenses," *IEEE Pervasive Computing*, vol. 7, no. 1, pp. 74–81, 2008. 21

[92] Y. Sun, L. Wu, S. Wu, S. Li, T. Zhang, L. Zhang, J. Xu, Y. Xiong, and X. Cui, "Attacks and countermeasures in the internet of vehicles," *Annals of Telecommunications*, vol. 72, no. 5, pp. 283–295, Jun 2017. [Online]. Available: https://doi.org/10.1007/s12243-016-0551-6 21

[93] F. Shahzad, M. Pasha, and A. Ahmad, "A survey of active attacks on wireless sensor networks and their countermeasures," *CoRR*, vol. abs/1702.07136, pp. 54–65, 2017. [Online]. Available: http://arxiv.org/abs/1702.07136 21

[94] J. Murphy, "Enhanced Security Controls for IBM Watson IoT Platform," *IBM*, 2016. [Online]. Available: https://developer.ibm.com/iotplatform/2016/09/23/enhanced-security-controls-for-ibm-watson-iot-platform/ 21, 25, 48, 55, 64

[95] "Secure adaptive routing protocol for Wireless Sensor Networks," 2018. [Online]. Available: https://www.dfcsc.uri.edu/docs/posters/sarp.pdf 21

[96] A. Kanuparthi, R. Karri, and S. Addepalli, "Hardware and embedded security in the context of Internet of Things," in *Proceedings of the ACM workshop on Security, privacy & dependability for cyber vehicles*, Berlin, Germany, Nov 2013, pp. 61–64. 21, 25

[97] "OWASP Top 10 2017 - The ten most critical web application security risks," 2017. [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_2017_Project 21, 26, 30

[98] "SQLi, XSS zero-days expose Belkin IoT devices, Android smartphones," 2016. [Online]. Available: https://www.csoonline.com/article/3138935/security/sqli-xss-zero-days-expose-belkin-iot-devices-android-smartphones.html 21, 26, 28

[99] "Cross-Site Scripting (XSS) attack," 2018. [Online]. Available: https://www.acunetix.com/websitesecurity/cross-site-scripting/ 21, 28

[100] K. Hamlen, M. Kantarcioglu, L. Khan, and B. Thuraisingham, "Security issues for cloud computing," *Optimizing Information Security and Advancing Privacy Assurance: New Technologies*, vol. 150, pp. 36–48, 2012. 21, 28

[101] F.-X. Standaert, "Introduction to side-channel attacks," in *Secure Integrated Circuits and Systems*. MA, USA: Springer, 2010, pp. 27–42. 22

[102] L. Robert, "Sony left passwords, code-signing keys virtually unprotected," 2014. [Online]. Available: https://www.eweek.com/security/sony-left-passwords-code-signing-keys-virtually-unprotected 24

[103] T. Pietraszek and C. V. Berghe, "Defending against injection attacks through context-sensitive string evaluation," in *Proceedings of the International Workshop on Recent Advances in Intrusion Detection*. WA, USA: Springer, 2005, pp. 124–145. 25

[104] "What is cloud computing)," 2018. [Online]. Available: https://www.ibm.com/cloud/learn/what-is-cloud-computing 28

[105] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: Towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2008. 28, 30

[106] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *Journal of internet services and applications*, vol. 1, no. 1, pp. 7–18, 2010. 28, 29, 64

[107] A. Oliner, A. Ganapathi, and W. Xu, "Advances and challenges in log analysis," *Communications of the ACM*, vol. 55, no. 2, pp. 55–61, 2012. 30

[108] S. Yu, *Distributed denial of service attack and defense*. NY, USA: Springer, 2014. 30

[109] T. Moore, J. Clulow, S. Nagaraja, and R. Anderson, "New strategies for revocation in ad-hoc networks," in *Proceedings of the European Workshop on Security in Ad-hoc and Sensor Networks*. Cambridge, UK: Springer, July 2007, pp. 232–246. 30

[110] Cisco, "Fog computing and the Internet of Things: Extend the cloud to where the things are," *Cisco Whitepaper*, pp. 1–6, 2015. 31

[111] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014. 31

[112] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2017. 31

[113] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing fog computing for Internet of Things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2017. 31, 65

[114] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018. 31

[115] K. Zhang, X. Liang, R. Lu, K. Yang, and X. S. Shen, "Exploiting mobile social behaviors for Sybil detection," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Hong kong, Apr-May 2015, pp. 271–279. 31

[116] S. Bruce, "Schneier on security: The future of ransomware," 2017. [Online]. Available: https://www.schneier.com/blog/archives/2017/05/the_future_of_r.html 31

[117] F. B. Lorenzo, "Hackers make the first-ever ransomware for smart thermostats," 2016. [Online]. Available: https://motherboard.vice.com/en_us/article/aekj9j/internet-of-things-ransomware-smart-thermostat 31

[118] "History of viruses, NIST computer security resource center," 1994. [Online]. Available: http://csrc.nist.gov/publications/nistir/threats/subsubsection3_3_1_1.html 31

[119] "Ransomware holding your data hostage, Deloitte: Threat intelligence and analytics," 2016. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/us/Documents/risk/us-aers-ransomware.pdf 31, 32

[120] "Timeline of computer viruses and worms," 2017. [Online]. Available: https://en.wikipedia.org/wiki/Timeline_of_computer_viruses_and_worms 31

[121] "National cyber awareness system, US CERT," 2019. [Online]. Available: https://www.us-cert.gov/ncas/alerts 31

[122] "New Petya / NotPetya / ExPetr ransomware outbreak, Kaspersky Lab," 2017. [Online]. Available: https://blog.kaspersky.com/new-ransomware-epidemics/17314/ 32

[123] "Duqu 2.0: The most sophisticated malware ever seen," 2015. [Online]. Available: http://resources.infosecinstitute.com/duqu-2-0-the-most-sophisticated-malware-ever-seen/#gref 32

[124] "Microsoft fixes 3 zero-day and many other flaws being exploited in the wild," 2014. [Online]. Available: http://securityaffairs.co/wordpress/29270/security/microsoft-fixes-3-zero-day.html 32

[125] "Microsoft issued a critical out-of-band patch for Kerberos flaw," 2014. [Online]. Available: http://securityaffairs.co/wordpress/30320/security/microsoft-patch-kerberos-bug.html 32

[126] "What exactly is Duqu 2.0?" 2015. [Online]. Available: https://community.rapid7.com/community/infosec/blog/2015/06/12/what-exactly-is-duqu-20 32

[127] C. Bronk and E. Tikk-Ringas, "The cyber attack on Saudi Aramco," *Survival*, vol. 55, no. 2, pp. 81–96, 2013. 32

[128] S. Zhioua, "The Middle East under malware attack dissecting cyber weapons," in *Proceedings of the 33rd IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW)*, PA, USA, July 2013, pp. 11–16. 32

[129] "Shamoon attacks possibly aided by Greenbug Group," 2017. [Online]. Available: http://www.securityweek.com/shamoon-attacks-possibly-aided-greenbug-group 32

[130] "Shamoon return prompts Saudi Arabia cyber warning," 2017. [Online]. Available: http://www.smh.com.au/world/shamoon-return-prompts-saudi-arabia-cyber-warning-20170124-gtxggi.html 32

[131] B. Miller and D. Rowe, "A survey of SCADA and critical infrastructure incidents," in *Proceedings of the ACM Special Interest Group for Information Technology Education Conference, SIGITE' 12*, AB, Canada, Oct 2012, pp. 51–56. 32

[132] E. Nakashima, G. Miller, and J. Tate, "US, Israel developed Flame computer virus to slow Iranian nuclear efforts, officials say," The Washington Post, 2012. 32

[133] B. Bencsáth, G. Pék, L. Buttyán, and M. Felegyhazi, "The cousins of Stuxnet: Duqu, Flame, and Gauss," *Future Internet*, vol. 4, no. 4, pp. 971–1003, 2012. 32

[134] A. Gostev, "The Flame: Questions and answers," *Securelist,{Online resource} Available at: https://www. securelist. com/en/blog/208193522/The_Flame_Questions_and_Answers*, 2012. 32

[135] "Common Vulnerabilities and Exposures-CVE-2010-2568," 2010. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2568 32

[136] GReAT, "Gauss: Nation-state cyber-surveillance meets banking trojan; Technical report; Kaspersky Labs ," Moscow, Russia, 2012. [Online]. Available: https://securelist.com/gauss-nation-state-cyber-surveillance-meets-banking-trojan-54/33854/ 32

[137] "The Icefog APT: A tale of cloak and three daggers, Kaspersky Labs," 2013. [Online]. Available: https://securelist.com/the-icefog-apt-a-tale-of-cloak-and-three-daggers/57331/ 32

[138] "Russian-Based Dragonfly Group attacks energy industry, RISI Online Incident database," 2015. [Online]. Available: http://www.risidata.com/Database/event_date/desc 32

[139] "Dragonfly: Western energy companies under sabotage threat. Symantec security response," 2016. [Online]. Available: https://www.symantec.com/connect/blogs/dragonfly-western-energy-companies-under-sabotage-threat 32

[140] "GReAT. Red October - Diplomatic cyber attacks investigation, Kaspersky Labs," 2014. [Online]. Available: https://securelist.com/red-october-diplomatic-cyber-attacks-investigation/36740/ 32

[141] G. Wangen, "The role of malware in reported cyber espionage: A review of the impact and mechanism," *Information*, vol. 6, no. 2, pp. 183–211, 2015. 32

[142] N. Virvilis and D. Gritzalis, "The big four: What we did wrong in advanced persistent threat detection?" in *Proceedings of the 8th IEEE International Conference on Availability, Reliability and Security (ARES)*, Regensburg, Germany, Sept 2013, pp. 248–254. 32

[143] "Night Dragon attacks target technology in energy sector, Forbes," 2011. [Online]. Available: http://www.forbes.com/sites/williampentland/2011/02/19/night-dragon-attacks-target-technology-in-energy-industry/#28c010114301 32

[144] "Trojans exploit WAP subscriptions to steal money, Kaspersky Lab," 2017. [Online]. Available: https://www.kaspersky.com/blog/wap-billing-trojans/18080/ 33

[145] J. M. Ehrenfeld, "WannaCry, cybersecurity and health information technology: A time to act," *Journal of Medical Systems*, vol. 41, no. 7, p. 104, 2017. [Online]. Available: http://dx.doi.org/10.1007/s10916-017-0752-1 33

[146] W. Victoria, "WannaCry ransomware: What is it and how to protect yourself," 2017. [Online]. Available: http://www.wired.co.uk/article/wannacry-ransomware-virus-patch 33

[147] C. Roger, "NHS ransomware attack spreads worldwide," 2017. [Online]. Available: http://www.cmaj.ca/content/189/22/E786 33

[148] R. Carol, "The impact of WannaCry on Industrial Control Systems (ICS)," 2016. [Online]. Available: http://iiot-world.com/cybersecurity/the-impact-of-wannacry-on-industrial-control-systems-ics/ 34

[149] D. Goodin, "Youre infected: If you want to see your data again, pay US $300 in Bitcoins," *Ars Technica*, Oct 2013. [Online]. Available: https://arstechnica.com/information-technology/2013/10/youre-infected-if-you-want-to-see-your-data-again-pay-us-300-in-bitcoins/ 34

[150] D. Oberhaus, "This luxury hotel is sick of ransomware attacks, so it's going analog," 2017. [Online]. Available: https://motherboard.vice.com/en_us/article/nzdznb/luxury-hotel-goes-analog-to-fight-ransomware-attacks 34

[151] "Mirai: What you need to know about the botnet behind recent major DDoS attacks, Symantec," 2016. [Online]. Available: https://www.symantec.com/connect/blogs/mirai-what-you-need-know-about-botnet-behind-recent-major-ddos-attacks 34

[152] "ICS-ALERT-14-176-02A," 2014. [Online]. Available: https://ics-cert.us-cert.gov/alerts/ICS-ALERT-14-176-02A 34, 52, 55

[153] "Havex hunts for ICS/SCADA systems," 2014. [Online]. Available: https://www.f-secure.com/weblog/archives/00002718.html 35

[154] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet dossier," *White Paper, Symantec Corp., Security Response*, vol. 5, no. 6, pp. 1–69, 2011. 35

[155] R. Langner, "To kill a centrifuge: A technical analysis of what Stuxnets creators tried to achieve," *The Langner Group Tech Report*, 2013. [Online]. Available: https://www.langner.com/wp-content/uploads/2017/03/to-kill-a-centrifuge.pdf 35

[156] T. M. Chen and S. Abu-Nimeh, "Lessons from Stuxnet," *Computer*, vol. 44, no. 4, pp. 91–93, 2011. 35

[157] K. Angrishi, "Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT botnets," *arXiv preprint arXiv:1702.03681*, pp. 1–17, Feb 2017. [Online]. Available: https://arxiv.org/pdf/1702.03681.pdf 38

[158] T. Tsvetkov and A. Klein, "RPL: IPv6 routing protocol for low power and lossy networks," *Network*, vol. 59, pp. 1–8, 2011. 40

[159] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security version 1.2," *RFC 6347*, 2012. 40

[160] "ISO 27001 Risk assessments, IT governance U.K." 2017. [Online]. Available: https://www.itgovernance.co.uk/iso27001/iso27001-risk-assessment 44, 54

[161] "NIST: Guide for conducting risk assessment," 2012. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf 44, 54

[162] "Do you have a defense-in-depth security strategy?, CISCO," 2017. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf 46, 54

[163] B. Greenstein, "IoT devices used in DDoS attacks," *IBM Internet of Things Blogs*, 2016. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/ddos-iot-platform-security/ 46, 54

[164] K. Lewis, "IoT security: What are the keys to protecting the castle 247?" *IBM Internet of Things Blogs*, 2017. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/security-iot-ibm/ 46, 48, 54, 63

[165] "Guidance for securing IoT using TCG technology, version 1, revision 21," 2015. [Online]. Available: https://www.business.att.com/cybersecurity/docs/vol5-datasecurity.pdf 46, 52, 53, 54, 55, 56, 58, 59, 63

[166] "TCG infrastructure WG TPM keys for platform identity for TPM 1.2," 2015. [Online]. Available: https://trustedcomputinggroup.org/wp-content/uploads/TPM_Keys_for_Platform_Identity_v1_0_r3_Final.pdf 46, 50, 54, 55, 59, 63

[167] "Five indisputable facts about IoT security, IBM Security," 2017. [Online]. Available: https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=SEF03018USEN 47, 48

[168] "ARM mbed," 2017. [Online]. Available: https://www.mbed.com/en/ 47, 54, 64

[169] M. Petko and B. Mark D., "IMA/EVM: Real applications for embedded networking systems," in *Proceedings of the Linux Security Summit*, WA, USA, Aug 2015. 47, 54

[170] "Secure authentication and anti-counterfeit technology," 2017. [Online]. Available: http://www.nxp.com/products/identification-and-security/secure-authentication-and-anti-counterfeit-technology:MC_71548 47, 54, 63

[171] GDPR, "General Data Protection Regulation (GDPR)," 2018. [Online]. Available: https://gdpr-info.eu/ 47, 122

[172] S. Carpov, T. H. Nguyen, R. Sirdey, G. Constantino, and F. Martinelli, "Practical privacy-preserving medical diagnosis using homomorphic encryption," in *Proceedings of the 9^{th} IEEE International Conference on Cloud Computing (CLOUD)*, CA, USA, June 2016, pp. 593–599. 48, 54, 64, 119

[173] B. Smith and K. Christidis, "IBM blockchain: An enterprise deployment of a distributed consensus-based transaction log," in *Proceedings of the 4^{th} International IBM Cloud Academy Conference*, Edmonton, Canada, June 2016, pp. 140–143. 48, 54

[174] "Microsoft Azure," 2017. [Online]. Available: https://azure.microsoft.com/en-au/?&WT.srch=1&WT.mc_ID=AID623263_SEM_MmqDz7Ol 48, 54

[175] "Hyperledger business blockchain technologies, The Linux Foundation," 2017. [Online]. Available: https://www.hyperledger.org/projects 48, 54, 77, 88, 89

[176] J. Matias, J. Garay, A. Mendiola, N. Toledo, and E. Jacob, "Flownac: Flow-based network access control," in *Proceedings of the 3^{rd} IEEE European Workshop on Software Defined Networks (EWSDN)*, Budapest, Hungary, Sept 2014, pp. 79–84. 48

[177] F. Jazib, C. Pignataro, A. Jeff, and M. Monique, "Securing the Internet of Things: A proposed framework," *Cisco Security Research & Operations*, 2015. [Online]. Available: http://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html 49, 55, 56, 64

[178] S. Chen, M. Ma, and Z. Luo, "An authentication scheme with identity-based cryptography for M2M security in cyber-physical systems," *Security and Communication Networks*, vol. 9, no. 10, pp. 1146–1157, 2016. 49, 54, 64

[179] Y. Qiu and M. Ma, "A mutual authentication and key establishment scheme for M2M communication in 6LoWPAN networks," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2074–2085, 2016. 49, 54, 64

[180] Y. Qiu, M. Ma, and S. Chen, "An anonymous authentication scheme for multi-domain machine-to-machine communication in cyber-physical systems," *Computer Networks*, vol. 129, pp. 306–318, 2017. 49, 54, 64

[181] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, and G. Ferrari, "Iot-OAS: An oauth-based authorization service architecture for secure services in IoT scenarios," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 1224–1234, 2015. 49

[182] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1832–1843, 2017. 50, 55

[183] M. Benmalek, Y. Challal, A. Derhab, and A. Bouabdallah, "Versami: Versatile and scalable key management for smart grid ami systems," *Computer Networks*, vol. 132, pp. 161–179, 2018. 50, 55

[184] A. Ghosal and M. Conti, "Key management systems for smart grid advanced metering infrastructure: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2831–2848, 2019. 50, 55

[185] "Information security advice: Network segmentation and segregation, Australian Government Department of Defence," 2012. [Online]. Available: https://www.asd.gov.au/publications/protect/network_segmentation_segregation.htm 51, 55

[186] O. Flauzac, C. González, A. Hachani, and F. Nolot, "SDN based architecture for IoT and improvement of the security," in *Proceedings of the 29th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Gwangiu, South Korea, Mar 2015, pp. 688–693. 51, 55

[187] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, "Software Defined Optical Networks (SDONs): A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2738–2786, 2016. 51

[188] B. Mokhtar and M. Azab, "Survey on security issues in vehicular ad hoc networks," *Alexandria Engineering Journal*, vol. 54, no. 4, pp. 1115–1126, 2015. 52

[189] M. Lee, "2016 saw an insane rise in the number of ransomware attacks, Forbes," 2016. [Online]. Available: https://www.forbes.com/sites/leemathews/2017/02/07/2016-saw-an-insane-rise-in-the-number-of-ransomware-attacks/#2aad814658dc 52

[190] "Ransomware: 5 Dos and Don'ts, Symantec Corporation," 2016. [Online]. Available: https://us.norton.com/internetsecurity-malware-ransomware-5-dos-and-donts.html 52

[191] "Save the data: Self encrypting drives, TCG," 2017. [Online]. Available: https://trustedcomputinggroup.org/wp-content/uploads/Infographic-TCG-SED.pdf 53, 55

[192] F. Abdi, M. Hasan, S. Mohan, D. Agarwal, and M. Caccamo, "ReSecure: A restart-based security protocol for tightly actuated hard real-time systems," *IEEE CERTS*, pp. 47–54, 2016. 53, 55, 56

[193] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for iot devices: survey, challenges and solutions," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 2017. 55

[194] B. J. Mohd, T. Hayajneh, and A. V. Vasilakos, "A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues," *Journal of Network and Computer Applications*, vol. 58, pp. 73–93, 2015. 55

[195] M. Agrawal, J. Zhou, and D. Chang, *A Survey on Lightweight Authenticated Encryption and Challenges for Securing Industrial IoT*. Springer, 2019. 55

[196] A. Meshram and C. Haas, "Anomaly detection in industrial networks using machine learning: A roadmap," in *Machine Learning for Cyber Physical Systems*. Heidelberg, Germany: Springer, 2017, pp. 65–72. 56

[197] I. Indre and C. Lemnaru, "Detection and prevention system against cyber attacks and botnet malware for information systems and Internet of Things," in *Proceedings of the 12th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, Sept 2016, pp. 175–182. 56

[198] S. M. A. M. Gadal and R. A. Mokhtar, "Anomaly detection approach using hybrid algorithm of data mining technique," in *Proceedings of the International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, Khartoum, Sudan, Jan 2017, pp. 1–6. 56

[199] C.-K. Chen, Z.-K. Zhang, S.-H. Lee, and S. Shieh, "Penetration testing in the iot age," *Computer*, vol. 51, no. 4, pp. 82–85, 2018. 56

[200] A. Gupta, *Performing an IoT Pentest*. Springer, 2019. 56

[201] V. Sivaraman, H. H. Gharakheili, C. Fernandes, N. Clark, and T. Karliychuk, "Smart iot devices in the home: Security and privacy implications," *IEEE Technology and Society Magazine*, vol. 37, no. 2, pp. 71–79, 2018. 57, 63

[202] "LPWA Technology : Security Comparison, A White paper by Franklin Health Ltd," 2017. [Online]. Available: https://fhcouk.files.wordpress.com/2017/05/lpwa-technology-security-comparison.pdf 58, 60, 61, 63

[203] N. Kshetri, "Can blockchain strengthen the Internet of Things?" *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017. 59, 63, 86

[204] E. Gaetani, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Blockchain-based database to ensure data integrity in cloud computing environments," *ePrint Research Center of Cyber Intelligence and Information Security, La Sapienza University of Rome*, 2017. 59, 86, 115, 119

[205] K. Granville, "Facebook and Cambridge Analytica: What you need to know as fallout widens," *New York Times*, 2018. [Online]. Available: https://nytimes.com/2018/03/19/technology/facebook-cambridge-analytica-explained.html 59, 87

[206] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018. 59

[207] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of BFT protocols," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, Vienna, Austria, Oct 2016, pp. 31–42. 59, 80, 82, 83, 96, 119, 160, 162

[208] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, 1978. 59, 82, 119

[209] F. B. Schneider, "Implementing fault-tolerant services using the state machine approach: A tutorial," *ACM Computing Surveys (CSUR)*, vol. 22, no. 4, pp. 299–319, 1990. 59, 82, 119

[210] "NEO - White Paper," 2017. [Online]. Available: http://docs.neo.org/en-us/ 59, 83, 96, 119, 160

[211] L. Aniello, R. Baldoni, E. Gaetani, F. Lombardi, A. Margheri, and V. Sassone, "A prototype evaluation of a tamper-resistant high performance blockchain-based transaction log for a distributed database," in *Proceedings of the 13th IEEE European Dependable Computing Conference (EDCC)*, Geneva, Switzerland, Sept 2017, pp. 151–154. 59, 115, 119

[212] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," *CoRR*, vol. abs/1506.03471, pp. 1–14, Jun 2015. [Online]. Available: http://arxiv.org/abs/1506.03471 59, 102, 110, 113, 114, 119

[213] Y. Li, X. Cheng, Y. Cao, D. Wang, and L. Yang, "Smart choice for the smart grid: Narrowband Internet of Things (NB-IoT)," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1505–1515, 2018. 60

[214] S. Johan, "Firmware updates over Low-Power Wide Area Networks, The Things Network," 2017. [Online]. Available: https://www.thethingsnetwork.org/article/firmware-updates-over-low-power-wide-area-networks 61, 62

[215] A. Elsaeidy, I. Elgendi, K. S. Munasinghe, D. Sharma, and A. Jamalipour, "A smart city cyber security platform for narrowband networks," in *Proceedings of the 27th IEEE InternationalTelecommunication Networks and Applications Conference (ITNAC)*, Melbourne, Aus, Nov 2017, pp. 1–6. 62, 63, 64

[216] T. Pecorella, L. Brilli, and L. Mucchi, "The role of physical layer security in IoT: A novel perspective," *Information*, vol. 7, no. 3, pp. 1–17, 2016. 63, 64

[217] O. Vermesan and P. Friess, *Internet of Things: Converging technologies for smart environments and integrated ecosystems*. River Publishers, 2013. 64

[218] P. Paillier *et al.*, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 99. Heidelberg, Germany: Springer, May 1999, pp. 223–238. 65

[219] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proceedings of the Theory of Cryptography Conference (TCC)*, vol. 3378. Heidelberg, Germany: Springer, Feb 2005, pp. 325–341. 65

[220] C. Gentry *et al.*, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the STOC*, vol. 9, no. 2009, MD, USA, May 2009, pp. 169–178. 65

[221] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Heidelberg, Germany: Springer, May 2010, pp. 24–43. 65

[222] N. Asokan, F. Brasser, A. Ibrahim, A.-R. Sadeghi, M. Schunter, G. Tsudik, and C. Wachsmann, "Seda: Scalable embedded device attestation," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CO, USA, Oct 2015, pp. 964–975. 65

[223] K. Biswas and V. Muthukkumarasamy, "Securing smart cities using blockchain technology," in *Proceedings of the 14th IEEE International Conference on Smart City High Performance Computing and Communications*, Sydney, Australia, Dec 2016, pp. 1392–1393. 65, 102, 103, 118, 123

[224] W. Reid, "How Bitcoin's technology could make supply chains more transparent," 2015. [Online]. Available: http://www.coindesk.com/how-bitcoins-technology-could-make-supply-chains/ 65, 106

[225] "Implement IoT and blockchain for accountability and security, IBM Watson IoT," 2017. [Online]. Available: https://www.ibm.com/internet-of-things/platform/private-blockchain/ 65

[226] "Blockchain startup Factom, Inc. raises series A funding," 2016. [Online]. Available: https://www.factom.com/news/factom-raises-series-a-funding 65

[227] D. Lee, "Arachneum: Blockchain meets distributed web," *arXiv preprint arXiv:1609.02789*, pp. 1–6, 2016. [Online]. Available: https://pdfs.semanticscholar.org/8146/0fdefc53e68a6a2f4198c1858984304671d5.pdf 65

[228] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proceedings of the 30th IEEE Symposium on Security and Privacy*, CA, USA, May 2009, pp. 173–187. 67

[229] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1900–1910, 2016. 67

[230] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 76, pp. 37–48, 2016. 67

[231] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, and X. Yao, "Security and privacy preservation scheme of face identification and resolution framework using fog computing in Internet of Things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1143–1155, 2017. 67

[232] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Heidelberg, Germany: Springer, May 2011, pp. 568–588. 67

[233] A. Sahai, B. Waters *et al.*, "Fuzzy identity-based encryption," in *Proceedings of the EUROCRYPT*, vol. 3494. Heidelberg, Germany: Springer, May 2005, pp. 457–473. 67

[234] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in iot: The challenges, and a way forward," *Journal of Network and Computer Applications*, vol. 125, pp. 251 – 279, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804518303473 69, 124, 131, 151, 160, 161

[235] I. Makhdoom, M. Abolhasan, and W. Ni, "Blockchain for IoT: The challenges and a way forward," in *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications - Volume 2: SECRYPT*, INSTICC. SciTePress, 2018, pp. 428–439. 69, 88, 93, 98, 100, 114, 117

[236] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology? - A systematic review," *PloS one*, vol. 11, no. 10, p. e0163477, 2016. 70, 119

[237] "Survey on blockchain technologies and related services ," Nomura Research Institute, Japan, 2015. [Online]. Available: http://www.meti.go.jp/english/press/2016/pdf/0531_01f.pdf 70

[238] M. Pilkington, "Blockchain technology: principles and applications," *Research handbook on digital transformations*, p. 225, 2016. 70, 77, 78

[239] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2015. 70, 80, 81

[240] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in Internet of Things: Challenges and solutions," *arXiv preprint arXiv:1608.05187*, 2016. 70, 102, 104, 124

[241] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in *Proceedings of the 19th IEEE International Conference on Advanced Communication Technology (ICACT)*, Phoenix Park, South Korea, Feb 2017, pp. 464–467. 70, 124

[242] M. Conoscenti, A. Vetr, and J. C. D. Martin, "Blockchain for the internet of things: A systematic literature review," in *Proceedings of the 13th IEEE/ACS International Conference of Computer Systems and Applications (AICCSA)*, Agadir, Morocco, Nov 2016, pp. 1–6. 70, 101, 119

[243] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, CA, USA, May 2015, pp. 104–121. 70, 71

[244] S. K. Lo, Y. Liu, S. Y. Chia, X. Xu, Q. Lu, L. Zhu, and H. Ning, "Analysis of blockchain solutions for iot: A systematic literature review," *IEEE Access*, vol. 7, pp. 58 822–58 835, 2019. 70, 71

[245] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *White Paper*, vol. 3, pp. 1–37, 2014. 70, 71, 79, 85, 88, 89, 90, 97, 98, 119, 124

[246] "XRP: The digital asset for payments," 2013. [Online]. Available: https://ripple.com/xrp/ 70

[247] "Gridcoin - White paper," 2018. [Online]. Available: https://www.gridcoin.us/assets/img/whitepaper.pdf 70

[248] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018. 75, 77, 78, 80, 81

[249] "How does Bitcoin work?" Bitcoin.org, 2017. [Online]. Available: https://bitcoin.org/en/how-it-works 75

[250] B. Vitalik, "The value of blockchain technology," 2015. [Online]. Available: https://blog.ethereum.org/2015/04/13/visions-part-1-the-value-of-blockchain-technology/ 75, 77

[251] "Bitcoin developer guide," 2017. [Online]. Available: https://bitcoin.org/en/developer-guide#block-chain 76, 78, 79, 89

[252] Bitcoin-Forum, "Difference between miners and nodes," 2016. [Online]. Available: https://bitcointalk.org/index.php?topic=1734235.0 76

[253] "Ethereum computer built on embedded devices," 2017. [Online]. Available: http://ethembedded.com/ 76, 94

[254] "Byzantine Consensus algorithm," 2018. [Online]. Available: https://tendermint.readthedocs.io/en/master/specification/byzantine-consensus-algorithm.html 77, 84, 96

[255] J. Garzik, "Public versus private blockchains part 1: Permissioned blockchains," 2015. 77, 78, 90

[256] K. Lukas, "In-depth on differences between public, private and permissioned blockchains," 2018. [Online]. Available: https://medium.com/@lkolisko/in-depth-on-differences-between-public-private-and-permissioned-blockchains-aff762f0ca24 77, 78

[257] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014. 77, 89

[258] "What is IOTA?" 2017. [Online]. Available: https://iota.readme.io/v1.5.0/docs 77, 89, 116

[259] "Litecoin," 2011. [Online]. Available: https://litecoin.org/ 77

[260] "Lisk SDK," 2018. [Online]. Available: https://lisk.io/documentation 77

[261] "Hyperledger-Fabric documentation," 2018. [Online]. Available: https://media.readthedocs.org/pdf/hyperledger-fabric/latest/hyperledger-fabric.pdf 77, 89, 90, 101, 119

[262] "Introduction to Hyperledger-Fabric," 2018. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/blockchain.html 78, 89

[263] G. Gideon, "MultiChain private blockchain - White paper," 2015. [Online]. Available: https://www.multichain.com/download/MultiChain-White-Paper.pdf 78, 119

[264] "Quorum - White paper," 2016. [Online]. Available: https://github.com/jpmorganchase/quorum-docs/blob/master/Quorum%20Whitepaper%20v0.1.pdf 78, 101, 119

[265] B. Vitalik, "On settlement finality," 2016. [Online]. Available: https://blog.ethereum.org/2016/05/09/on-settlement-finality/ 78, 80

[266] B. Vitalik, "On public and private blockchains," 2015. [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/ 78

[267] L. Alex, "A History of Bitcoin Forks: Top 5 Bitcoin Forks, Rated and Reviewed," 2018. [Online]. Available: https://www.bitcoinmarketjournal.com/bitcoin-forks/ 79

[268] P. Sebastin, "An introduction to Ethereum and smart contracts: A programmable blockchain," 2017. [Online]. Available: https://auth0.com/blog/an-introduction-to-ethereum-and-smart-contracts-part-2/ 79

[269] D. Tuesta, J. Alonso, N. Cámara *et al.*, "Smart contracts: The ultimate automation of trust," *Abgerufen am*, vol. 3, p. 2016, 2015. 79

[270] S. Huckle, R. Bhattacharya, M. White, and N. Beloff, "Internet of Things, blockchain and shared economy applications," *Procedia Computer Science*, vol. 98, pp. 461–466, 2016. 79

[271] A. Baliga, "Understanding blockchain consensus models," *Persistent Systems Ltd. White Paper*, 2017. 80, 95

[272] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-Work vs. BFT replication," in *Proceedings of the International Workshop on Open Problems in Network Security*. Zurich, Switzerland: Springer, Oct 2015, pp. 112–125. 80, 82, 96, 162, 163

[273] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proceedings of the IEEE International Congress on Big Data (BigData Congress)*, HI, USA, June 2017, pp. 557–564. 80, 81

[274] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Commun. ACM*, vol. 61, no. 7, pp. 95–102, Jun. 2018. [Online]. Available: http://doi.acm.org/10.1145/3212998 80

[275] N. Szabo, "The idea of smart contracts," *IEEE International Workshop on Electronic Contracting (WEC)*, 2004. 80, 95, 119

[276] N. Houy, "It will cost you nothing to 'kill'a Proof-of-Stake cryptocurrency," *Econ.Bull*, vol. 34, no. 2, pp. 1038–1044, 2014. 81

[277] Y. Gao and H. Nobuhara, "A Proo-of-Stake sharding protocol for scalable blockchains," *Proceedings of the Asia-Pacific Advanced Network*, vol. 44, pp. 13–16, 2017. 81

[278] EconoTimes, "Blockchain project Antshares explains reasons for choosing dBFT over PoW and PoS," 2017. [Online]. Available: http://www.econotimes.com/Blockchain-project-Antshares-explains-reasons-for-choosing-dBFT-over-PoW-and-PoS-659275 81, 83, 95, 159

[279] Bitcoinwiki, "Scalability," 2017. [Online]. Available: https://en.bitcoin.it/wiki/Scalability 81

[280] D. Larimer, "Delegated Proof-of-Stake (DPOS)," *Bitshare whitepaper*, 2014. 81, 85

[281] J. Kwon, "Tendermint: Consensus without mining," *Draft v. 0.6, fall*, 2014. 81

[282] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of Activity: Extending Bitcoin's Proof-of-Work via Proof-of-Stake," *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 3, pp. 34–37, 2014. 81

[283] Ethcore, "Parity," 2018. [Online]. Available: https://wiki.parity.io/ 81, 119

[284] "Proof of Authority: Consensus model with identity at stake," 2017. [Online]. Available: https://wiki.parity.io/ 81, 119

[285] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "BLOCKBENCH: A framework for analyzing private blockchains," in *Proceedings of the ACM International Conference on Management of Data*, ser. SIGMOD '17, IL, USA, May 2017, pp. 1085–1100. [Online]. Available: http://doi.acm.org/10.1145/3035918.3064033 81

[286] R. Kastelein, "Intel jumps into blockchain technology storm with Sawtooth Lake distributed ledger," 2016. [Online]. Available: http://www.the-blockchain.com/2016/04/09/ 81, 95, 119, 169

[287] S. Iain, "Proof of Burn," 2018. [Online]. Available: https://en.bitcoin.it/wiki/Proof_of_burn#cite_note-1 82, 119

[288] "Slimcoin: The next generation of cryptocurrencies," 2014. [Online]. Available: http://slimco.in/ 82

[289] M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems (TOCS)*, vol. 20, no. 4, pp. 398–461, 2002. 82, 90, 96, 97, 160

[290] E. A. Brewer, "Towards robust distributed systems," in *Proceedings of the PODC*, vol. 7, 2000. 82

[291] T. Hardjono and N. Smith, "Cloud-based commissioning of constrained devices using permissioned blockchains," in *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*, 2016, pp. 29–36. 82

[292] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proceedings of the 13th IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2013, pp. 1–10. 82, 96, 160

[293] Z. Erik, "A Byzantine fault tolerance algorithm for blockchain," 2017. [Online]. Available: http://docs.neo.org/en-us/node/whitepaper.html 83, 119

[294] Neo.org, "Consensus," 2017. [Online]. Available: http://docs.neo.org/en-us/node/consensus.html 83

[295] Steemit, "Neo's consensus protocol: How delegated Byzantine fault tolerance works," 2017. [Online]. Available: https://steemit.com/neo/\spacefactor\@m{}basiccrypto/neo-s-consensus-protocol-how-delegated-byzantine-fault-tolerance-works 83

[296] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling Byzantine agreements for cryptocurrencies," in *Proceedings of the 26th Symposium on Operating Systems Principles*. Shanghai, China: ACM, Oct 2017, pp. 51–68. 84, 97, 160

[297] S. Popov, "The Tangle," *cit. on*, pp. 1–28, 2016. 84, 88, 89, 90, 95, 116

[298] "IOTA vulnerability report: Cryptanalysis of the Curl hash function enabling Practical Signature Forgery attacks on the IOTA cryptocurrency," 2017. [Online]. Available: https://github.com/mit-dci/tangled-curl/blob/master/vuln-iota.md 85

[299] "Blockchain size," 2017. [Online]. Available: https://www.blockchain.com/charts/blocks-size 86

[300] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018. 87

[301] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger-Fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the 13th EuroSys Conference*. Porto, Portugal: ACM, Apr 2018, pp. 1–15. 87, 89, 90, 93, 154

[302] "Hyperledger-Fabric Model - Privacy," 2017. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.2/fabric_model.html 88, 90

[303] "Hyperledger-Fabric: Security and access Control," 2017. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.2/Fabric-FAQ.html 88

[304] X. Min, Q. Li, L. Liu, and L. Cui, "A permissioned blockchain framework for supporting instant transaction and dynamic block size," in *Proceedings of the IEEE Trustcom/BigDataSE/I SPA*, Tianjin, China, Aug 2016, pp. 90–96. 88

[305] G. S. Ramachandran and B. Krishnamachari, "Blockchain for the IoT: Opportunities and challenges," *CoRR*, vol. abs/1805.02818, 2018. [Online]. Available: http://arxiv.org/abs/1805.02818 88

[306] K. Preethi, "Blockchains dont scale. Not today, at least. But theres hope." 2017. [Online]. Available: https://hackernoon.com/blockchains-dont-scale-not-today-at-least-but-there-s-hope-2cb43946551a 88

[307] P. K. Sharma, M. Y. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2018. 88

[308] Bitcoin.org, "Warning: Better security has costs," 2017. [Online]. Available: https://bitcoin.org/en/bitcoin-core/features/requirements 89

[309] R. James, "Ethereum/Wiki - mining," 2018. [Online]. Available: https://github.com/ethereum/wiki/wiki/Mining 89

[310] "Ethereum transaction chart," 2018. [Online]. Available: https://etherscanio/chart/tx 89

[311] "Hyperledger - White Paper," 2016. [Online]. Available: https://github.com/hyperledger/hyperledger/wiki/Whitepaper-WG 89, 91

[312] C. Cachin, "Architecture of the Hyperledger blockchain fabric," in *Proceedings of the Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, vol. 310, IL, USA, July 2016. 89

[313] "Gas with hyperledger-fabric?" 2016. [Online]. Available: https://stackoverflow.com/questions/38635778/gas-with-hyperledger-fabric 89

[314] "Hyperledger Aries," 2019. [Online]. Available: https://www.hyperledger.org/blog/2019/05/14/announcing-hyperledger-aries-infrastructure-supporting-interoperable-identity-solutions 91

[315] M. Scherer, "Performance and scalability of blockchain networks and smart contracts," Master's thesis, Umea University, Sweden, 2017. 91

[316] C. Cachin and M. Vukolić, "Blockchains consensus protocols in the wild," *arXiv preprint arXiv:1707.01873*, 2017. 93

[317] "Light client protocol," 2018. [Online]. Available: https://github.com/ethereum/wiki/wiki/Light-client-protocol 94

[318] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, "IEEE 802.15.4: A developing standard for low-power low-cost wireless personal area networks," *IEEE network*, vol. 15, no. 5, pp. 12–19, 2001. 94

[319] S. Chen, R. Ma, H.-H. Chen, H. Zhang, W. Meng, and J. Liu, "Machine-to-machine communications in ultra-dense networks - A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1478–1503, 2017. 94

[320] S. C. Ergen, "ZigBee/IEEE 802.15. 4 summary," *UC Berkeley, September*, vol. 10, p. 17, 2004. 94

[321] "Sigfox services," 2018. [Online]. Available: https://www.sigfox.com/en 94

[322] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, "On security analysis of proof-of-elapsed-time (PoET)," in *Proceedings of the International Symposium on Stabilization, Safety, and Security of Distributed Systems*. MA, USA: Springer, Nov 2017, pp. 282–297. 95

[323] A. Bessani, J. Sousa, and E. E. Alchieri, "State machine replication for the masses with BFT-SMaRt," in *Proceedings of the 44th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, GA, USA, June 2014, pp. 355–362. 97

[324] L. Luu, V. Narayanan, K. Baweja, C. Zheng, S. Gilbert, and P. Saxena, "SCP: A computationally-scalable Byzantine consensus protocol for blockchains," *IACR Cryptology ePrint Archive*, vol. 20, no. 20, pp. 1–16, 2015. 97

[325] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, and J. Melia, "Understanding the limits of LoRaWAN," *arXiv preprint arXiv:1607.08011*, 2016. 97, 110

[326] Bitcoin-Developer-Guide, "Transactions," Developer Guide, 2018. [Online]. Available: https://bitcoin.org/en/developer-guide#transactions 97

[327] "Transaction flow," 2019. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html 98

[328] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *Proceedings of the NSDI*, CA, USA, Mar 2016, pp. 45–59. 99, 119

[329] Oraclize, "How it works," 2018. [Online]. Available: http://www.oraclize.it/ 100

[330] "Monero: Private digital currency," 2017. [Online]. Available: https://getmonero.org/ 101, 119

[331] "Zerocoin project," 2018. [Online]. Available: http://zerocoin.org/ 101, 119

[332] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, CA, USA, May 2016, pp. 839–858. 101, 119

[333] IBM, "ADEPT: An Internet of Things practitioner perspective," IBM, Tech. Rep., 2015. [Online]. Available: https://archive.org/details/pdfy-esMcC00dKmdo53-_ 101, 102, 103, 114, 118, 119

[334] B. Lee and J.-H. Lee, "Blockchain-based secure firmware update for embedded devices in an Internet of Things environment," *The Journal of Supercomputing*, pp. 1–16, 2016. 102, 103, 113, 119

[335] A. Dorri, S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proceedings of the 2nd IEEE Workshop on security, privacy, and trust in the Internet of things (PERCOM)*, HI, USA, Mar 2017. 102, 104, 124

[336] B. Leiding, P. Memarmoshrefi, and D. Hogrefe, "Self-managed and blockchain-based vehicular ad-hoc networks," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, Heidelberg, Germany, Sept 2016, pp. 137–140. 102, 105, 124

[337] Y. Zhang and J. Wen, "The IoT electric business model: Using blockchain technology for the Internet of Things," *Peer-to-Peer Networking and Applications*, pp. 1–12, 2016. 102, 105, 124

[338] S. Underwood, "Blockchain beyond Bitcoin," *Communications of the ACM*, vol. 59, no. 11, pp. 15–17, 2016. 102, 106

[339] R. Kastelein, "Everledger rolls out blockchain technology to digitally certify Kimberley Diamonds," 2016. [Online]. Available: http://www.the-blockchain.com/2016/09/20/everledger/ 102, 106

[340] Y. P. Tsang, K. L. Choy, C. H. Wu, G. T. S. Ho, and H. Y. Lam, "Blockchain-driven iot for food traceability with an integrated consensus mechanism," *IEEE access*, vol. 7, pp. 129 000–129 017, 2019. 102, 108

[341] J. Christoph, "Slock.it 3 minutes demo," 2015. [Online]. Available: https://slock.it/index.html 102, 108

[342] M. Castro, B. Liskov *et al.*, "Practical Byzantine Fault Tolerance (PBFT)," in *Proceedings of the OSDI*, vol. 99, no. 1999, LA, USA, Feb 1999, pp. 173–186. 104, 160

[343] M. Michael and D. Buell, "Bluemix is now IBM cloud," 2018. [Online]. Available: https://www.ibm.com/blogs/bluemix/2017/10/bluemix-is-now-ibm-cloud/ 106

[344] K. Wüst and A. Gervais, "Do you need a blockchain?" *IACR Cryptology ePrint Archive*, vol. 2017, p. 375, 2017. 106

[345] G. Prisco, "Slock.it to introduce smart locks linked to smart ethereum contracts, decentralize the sharing economy," *Bitcoin Magazine*, 2015. [Online]. Available: https://bitcoinmagazine.com/articles/slock-it-to-introduce-smart-locks-linked-to-smart-ethereum-contracts-decentralize-the-sharing-economy-144 108

[346] G. Zyskind, O. Nathan *et al.*, "Decentralizing privacy: Using blockchain to protect personal data," in *Proceedings of the IEEE Security and Privacy Workshops (SPW)*, CA, USA, May 2015, pp. 180–184. 109, 114, 118, 119

[347] MIT-Media-Lab, "Ethos," 2014. [Online]. Available: http://viral.media.mit.edu/projects/ethos/ 110

[348] F. F. E. Dabek, "A distributed hash table," Ph.D. dissertation, Massachusetts Institute of Technology, MA, USA, 2005. 114, 119

[349] K. Konstantinos, S. Angelos, B. Irena, V. Jeff, and T. Grance, "Leveraging blockchainbased protocols in IoT systems," *NIST - Computer Security Resource Center*, 2016. 115, 119

[350] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," *White Paper*, 2017. 115, 119

[351] "REX-Blog", "Sharding, Raiden, Plasma: The scaling solutions that will unchain Ethereum," 2017. [Online]. Available: https://blog.rexmls.com/sharding-raiden-plasma-the-scaling-solutions-that-will-unchain-ethereum-c590e994523b 115, 116

[352] "BigchainDB: The blockchain database," 2018. [Online]. Available: https://www.bigchaindb.com/ 115, 119

[353] R. James, "On sharding blockchains," 2018. [Online]. Available: https://github.com/ethereum/wiki/wiki/Sharding-FAQ 115, 116

[354] "An introduction to IOTA," 2017. [Online]. Available: http://www.iotasupport.com/whatisiota.shtml 116, 117

[355] F. Armknecht, G. O. Karame, A. Mandal, F. Youssef, and E. Zenner, "Ripple: Overview and outlook," in *Proceedings of the International Conference on Trust and Trustworthy Computing*. Crete, Greece: Springer, Aug 2015, pp. 163–180. 119

[356] P. Ruckebusch, E. De Poorter, C. Fortuna, and I. Moerman, "Gitar: Generic extension for Internet-of-Things architectures enabling dynamic updates of network and application modules," *Ad-Hoc Networks*, vol. 36, pp. 127–151, 2016. 119

[357] A. Taherkordi, F. Loiret, R. Rouvoy, and F. Eliassen, "Optimizing sensor network reprogramming via in situ reconfigurable components," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 2, p. 14, 2013. 119

[358] "Alastria: National blockchain ecosystem ," 2017. [Online]. Available: https://alastria.io/#1 119

[359] Telehash, "Telehash encrypted mesh protocol," 2017. [Online]. Available: http://telehash.org/ 118

[360] I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, "Privysharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities," *Computers & Security*, vol. 88, p. 101653, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016740481930197X 121

[361] I. Makhdoom., I. Zhou., M. Abolhasan., J. Lipman., and W. Ni., "Privysharing: A blockchain-based framework for integrity and privacy-preserving data sharing in smart cities," in *Proceedings of the 16th International Joint Conference on e-Business and Telecommunications - Volume 2: SECRYPT,*, INSTICC. Prague, Czech Republic: SciTePress, 2019, pp. 363–371. 121

[362] V. Moustaka, Z. Theodosiou, A. Vakali, and A. Kounoudes, "Smart cities at risk!: Privacy and security borderlines from social networking in cities," *Athena*, vol. 357, pp. 905–910, 2018. 122

[363] A. Bartoli, J. Hernández-Serrano, M. Soriano, M. Dohler, A. Kountouris, and D. Barthel, "Security and privacy in your smart city," in *Proceedings of the Barcelona Smart Cities Congress*, vol. 292, Barcelona, Spain, Dec 2011, pp. 1–6. 122

[364] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129, 2017. 122

[365] L. Cui, G. Xie, Y. Qu, L. Gao, and Y. Yang, "Security and privacy in smart cities: Challenges and opportunities," *IEEE Access*, vol. 6, pp. 46 134–46 145, 2018. 122, 128

[366] R. A. Michelin, A. Dorri, M. Steger, R. C. Lunardi, S. S. Kanhere, R. Jurdak, and A. F. Zorzo, "SpeedyChain: A framework for decoupling data from blockchain for smart cities," in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. NY, USA: ACM, Nov 2018, pp. 145–154. 123

[367] P. K. Sharma and J. H. Park, "Blockchain based hybrid network architecture for the smart city," *Future Generation Computer Systems*, vol. 86, pp. 650–655, 2018. 123

[368] M. A. Rahman, M. M. Rashid, M. S. Hossain, E. Hassanain, M. F. Alhamid, and M. Guizani, "Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city," *IEEE Access*, vol. 7, pp. 18 611–18 621, 2019. 123

[369] D. A. Kountché, J.-M. Bonnin, and H. Labiod, "The problem of privacy in Cooperative Intelligent Transportation Systems (C-ITS)," in *Proceedings of the Computer Communications Workshops (INFOCOM WKSHPS)*. GA, USA: IEEE, May 2017, pp. 482–486. 124

[370] F. Haidar, A. Kaiser, and B. Lonc, "On the performance evaluation of vehicular PKI protocol for V2X communications security," in *Proceedings of the 86th Vehicular Technology Conference (VTC-Fall)*. Toronto, Canada: IEEE, Sept 2017, pp. 1–5. 124

[371] K. N. Krishnan, R. Jenu, T. Joseph, and M. Silpa, "Blockchain based security framework for IoT implementations," in *Proceedings of the International CET Conference on Control, Communication, and Computing (IC4)*. Thiruvananthapuram, India: IEEE, July 2018, pp. 425–429. 124

[372] Y. Qian, Z. Liu, J. Yang, and Q. Wang, "A method of exchanging data in smart city by blockchain," in *Proceedings of the 16th International Conference on Smart City*. Exeter, UK: IEEE, June 2018, pp. 1344–1349. 124

[373] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, "GDPR-compliant personal data management: A blockchain-based solution," *IEEE Transactions on Inromation Forensics and Security*, pp. 1–13, 2019. 124

[374] B. Faber, G. C. Michelet, N. Weidmann, R. R. Mukkamala, and R. Vatrapu, "BPDIMS: A blockchain-based personal data and identity management system," in *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS)*. HI, USA: IEEE, Jan 2019, pp. 6855–6864. 125

[375] R. Neisse, G. Steri, and I. Nai-Fovino, "A blockchain-based approach for data accountability and provenance tracking," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*. Reggio Calabria, Italy: ACM, Aug 2017, pp. 1–10. 125

[376] K. Rantos, G. Drosatos, K. Demertzis, C. Ilioudis, A. Papanikolaou, and A. Kritsas, "ADvoCATE: A consent management platform for personal data processing in the IoT using blockchain technology," in *Proceedings of the International Conference on Security for Information Technology and Communications*. Bucharest, Romania: Springer, Nov 2018, pp. 300–313. 125

[377] N. Kaaniche and M. Laurent, "A blockchain-based data usage auditing architecture with enhanced privacy and availability," in *Proceedings of the 16th International Symposium on Network Computing and Applications (NCA)*. MA, USA: IEEE, Oct 2017, pp. 1–5. 126

[378] W. J. Gordon and C. Catalini, "Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 224–230, 2018. 126

[379] Hyperledger-Fabric, "Smart contracts and chaincode," 2019. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/smartcontract/smartcontract.html 127

[380] "Blockchain network," 2019. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/network/network.html 127

[381] O. Mazhelis, A. Hämäläinen, T. Asp, and P. Tyrväinen, "Towards enabling privacy preserving smart city apps," in *Proceedings of the International Smart Cities Conference (ISC2)*. Trento, Italy: IEEE, Sept 2016, pp. 1–7. 128

[382] I. Makhdoom, M. Abolhasan, and W. Ni, "Blockchain for IoT: The challenges and a way forward," in *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications - Volume 2: SECRYPT*, INSTICC. Porto, Portugal: SciTePress, Jul 2018, pp. 428–439. 131, 160

[383] J. C. Anderson, J. Lehnardt, and N. Slater, *CouchDB: The definitive guide: Time to relax*. O'Reilly Media, Inc., 2010. 131

[384] A. Dent, *Getting started with LevelDB*. Packt Publishing Ltd, 2013. 131

[385] H. Mike and G. B. Richard, "Corda: A distributed ledger," 2019. [Online]. Available: https://www.r3.com/wp-content/uploads/2019/08/corda-technical-whitepaper-August-29-2019.pdf 131

[386] R. G. Brown, "Introducing r3 corda: A distributed ledger designed for financial services," *R3 Blog*, vol. 5, 2016. 131

[387] M. Valenta and P. Sandner, "Comparison of ethereum, hyperledger fabric and corda.[ebook] frankfurt school," *Blockchain Center*, pp. 1–8, 2017. 131

[388] M. Marchesi, L. Marchesi, and R. Tonelli, "An agile software engineering method to design blockchain applications," in *Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia*. Moscow, Russia: ACM, Oct 2018, pp. 1–8. 135

[389] N. Apthorpe, D. Reisman, and N. Feamster, "A smart home is no castle: Privacy vulnerabilities of encrypted IoT traffic," *arXiv preprint arXiv:1705.06805*, pp. 1–6, 2017. 141, 142

[390] Hyperledger-Fabric, "Identity,," 2019. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/identity/identity.html 141

[391] A. Dorri, S. S. Kanhere, and R. Jurdak, "MOF-BC: A memory optimized and flexible blockchain for large scale networks," *Future Generation Computer Systems*, vol. 92, pp. 357–373, 2019. 141

[392] C. Roulin, A. Dorri, R. Jurdak, and S. Kanhere, "On the activity privacy of blockchain for IoT," *arXiv preprint arXiv:1812.08970*, pp. 1–8, 2018. 141, 142

[393] D. Hardt, "The OAuth 2.0 authorization framework," Internet Requests for Comments, RFC Editor, RFC 6749, 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6749.txt 144

[394] S. Pongnumkul, C. Siripanpornchana, and S. Thajchayapong, "Performance analysis of private blockchain platforms in varying workloads," in *Proceedings of the 26th International Conference on Computer Communication and Networks (ICCCN)*. Vancouver, Canada: IEEE, July-Aug 2017, pp. 1–6. 151

[395] N. J. Salkind, "Student's t-Test," *Encyclopedia of Research Design*, 2010. 152

[396] "Hypothesis testing (P-value approach)," 2019. [Online]. Available: https://onlinecourses.science.psu.edu/statprogram/reviews/statistical-concepts/hypothesis-testing/p-value-approach 152

[397] F. Christopher, "Hyperledger-Fabric performance and scale ," 2019. [Online]. Available: https://www.ibm.com/blogs/blockchain/2019/01/answering-your-questions-on-hyperledger-fabric-performance-and-scale/ 154

[398] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computingA key technology towards 5G," *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015. 155

[399] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, "Edge computing enabling the Internet of Things," in *Proceedings of the 2nd World Forum on Internet of Things (WF-IoT)*. Milan, Italy: IEEE, Dec 2015, pp. 603–608. 155

[400] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Replisom: Disciplined tiny memory replication for massive IoT devices in LTE edge cloud," *Internet of Things Journal*, vol. 3, no. 3, pp. 327–338, 2016. 155

[401] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the Internet of Things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016. 155

[402] FP7 European Project, "Distributed computing, storage and radio resource allocation over cooperative femtocells (TROPIC)," 2012. [Online]. Available: http://www.ict-tropic.eu/ 155

[403] F. Lobillo, Z. Becvar, M. A. Puente, P. Mach, F. L. Presti, F. Gambetti, M. Goldhamer, J. Vidal, A. K. Widiawan, and E. Calvanesse, "An architecture for mobile computation offloading on cloud-enabled LTE small cells," in *Proceedings of the Wireless Communications and Networking Conference Workshops (WCNCW)*. Istanbul, Turkey: IEEE, Apr 2014, pp. 1–6. 155

[404] S. Wang, G.-H. Tu, R. Ganti, T. He, K. Leung, H. Tripp, K. Warr, and M. Zafer, "Mobile micro-cloud: Application classification, mapping, and deployment,," in *Proceedings of the Annual Fall Meeting of ITA (AMITA)*, 2013, pp. 1–7. 155

[405] K. Wang, M. Shen, J. Cho, A. Banerjee, J. Van der Merwe, and K. Webb, "Mobiscud: A fast moving personal cloud in the mobile network," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. London, UK: ACM, Aug 2015, pp. 19–24. 155

[406] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017. 156

[407] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, "Overview of 5G security challenges and solutions," *Communications Standards Magazine*, vol. 2, no. 1, pp. 36–43, 2018. 156

[408] M. A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, "Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes," *Journal of Network and Computer Applications*, vol. 101, pp. 55–82, 2018. 156

[409] M. Imran, T. Farzad, Z. Ian, A. Mehran, and L. Justin, "PLEDGE: A Proof-of-Honesty based Consensus Protocol for Blockchain-based IoT Systems," in *Press in Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency*, IEEE ComSoc. Toronto, Canada: IEEE, 2020. 159

[410] V. Gramoli, "From blockchain consensus back to Byzantine consensus," *Future Generation Computer Systems*, vol. 88, pp. 173–90, 2017. 159

[411] A. Agbaria and R. Friedman, "Overcoming Byzantine failures using checkpointing," *Coordinated Science Laboratory Report no. UILU-ENG-03-2228, CRHC-03-14*, 2003. 160

[412] "Cliques algorithm - Proof of Authority consensus," 2018. [Online]. Available: https://steemit.com/steemstem/@mareng/clique-algorithm-proof-of-authority-consensus 160

[413] J. Bridges, "Reputation lessons from Warren Buffett," 2013. [Online]. Available: https://www.reputationdefender.com/blog/orm/reputation-lessons-warren-buffett 160

[414] A. Y. Al-Hibri, "Islamic constitutionalism and the concept of democracy," *Case W. Res. j. Int'l L.*, vol. 24, pp. 1–29, 1992. 160, 161

[415] U. Shavit, "Is Shura a muslim form of democracy? Roots and systemization of a polemic," *Middle Eastern Studies*, vol. 46, no. 3, pp. 349–374, 2010. 161

[416] R. Dennis and G. Owen, "Rep on the block: A next generation reputation system based on the blockchain," in *Proceedings of the 10th International Conference for Internet Technology and Secured Transactions (ICITST)*. London, UK: IEEE, Dec 2015, pp. 131–138. 161, 162

[417] Y. Wang and J. Vassileva, "Trust and reputation model in peer-to-peer networks," in *Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P2003)*. Linkoping, Sweden: IEEE, Sept 2003, pp. 150–157. 161

[418] M. Gupta, P. Judge, and M. Ammar, "A reputation system for peer-to-peer networks," in *Proceedings of the 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video*. CA, USA: ACM, June 2003, pp. 144–152. 161

[419] M. Herlihy and M. Moir, "Enhancing accountability and trust in distributed ledgers," *arXiv preprint arXiv:1606.07490*, pp. 1–15, 2016. 162

[420] A. Mohan and D. M. Blough, "Attributetrust a framework for evaluating trust in aggregated attributes via a reputation system," in *Proceedings of the 6th Annual Conference on Privacy, Security and Trust*. Brunswick, Canada: IEEE, Oct 2008, pp. 201–212. 162, 173, 174

[421] L. Bahri and S. Girdzijauskas, "When trust saves energy: A reference framework for Proof of Trust (PoT) blockchains," in *Companion Proceedings of The Web Conference*. Lyon , France: International World Wide Web Conferences Steering Committee, Apr 2018, pp. 1165–1169. 162

[422] Q. Zhuang, Y. Liu, L. Chen, and Z. Ai, "Proof of Reputation: A Reputation-based Consensus Protocol for Blockchain Based Systems," in *Proceedings of the International Electronics Communication Conference*. ACM, 2019, pp. 131–138. 162

[423] J. Kwon, "Tendermint: Consensus without mining," *Draft v. 0.6, Fall*, vol. 1, pp. 1–11, 2014. 163

[424] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018. 163

[425] Solidity-Documentation, "Events," 2019. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/solidity/develop/solidity.pdf 164

[426] "Remix-Ethereum-IDE," 2019. [Online]. Available: https://remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.5.11+commit.c082d0b4.js 170

[427] Eth-Gas-Station, "Whats the maximum Ethereum block size?" 2019. [Online]. Available: https://ethgasstation.info/blog/ethereum-block-size/ 170

[428] J. Linn and M. Nyström, "Attribute certification: An enabling technology for delegation and role-based controls in distributed environments," in *Proceedings of the 4th ACM Workshop on Role-based Access Control*. VA, USA: ACM, Oct 1999, pp. 121–130. 173

[429] M. K. Reiter and S. G. Stubblebine, "Authentication metric analysis and design," *ACM Transactions on Information and System Security (TISSEC)*, vol. 2, no. 2, pp. 138–158, 1999. 173

[430] D. W. Chadwick, "Authorisation using attributes from multiple authorities," in *Proceedings of the 15th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'06)*. Manchester, UK: IEEE, June 2006, pp. 326–331. 173

[431] N. Klingenstein, "Attribute aggregation and federated identity," in *Proceedings of the International Symposium on Applications and the Internet Workshops*. Hiroshima, Japan: IEEE, Jan 2007, pp. 1–4. 173

[432] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007. 173

[433] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Computing Surveys (CSUR)*, vol. 42, no. 1, pp. 1–34, 2009. 174

[434] E. Heilman, N. Narula, G. Tanzer, J. Lovejoy, M. Colavita, M. Virza, and T. Dryja, "Cryptanalysis of Curl-P and other attacks on the IOTA cryptocurrency," *IACR Cryptology ePrint Archive*, vol. 2019, p. 344, 2019. 177

[435] Ethereum-StackExchange, "Block time and confirmation time," 2019. [Online]. Available: https://ethereum.stackexchange.com/questions/56338/block-time-and-confirmation-time 178

[436] StackExchange, "What is the average transaction time in IOTA?" 2019. [Online]. Available: https://iota.stackexchange.com/questions/88/what-is-the-average-transaction-time-in-iota 178