Real-time Integration of IoT Sensor and IOTA Tangle for Securing IoT Infrastructure

by

Md Abdullah Al Mamun

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
November 2021

Thesis advisor
Dr. Rasit Eskicioglu and Dr. Cuneyt Akcora

Author
Md Abdullah Al Mamun

Real-time Integration of IoT Sensor and IOTA Tangle for Securing IoT Infrastructure

Abstract

In recent years, the Internet of Things (IoT) has seen tremendous growth. The fast-growing IoT field is facing many security issues because infrastructure configuration and security policies employed in this nascent field have not matured yet. In this research, we argue that the security issues on IoT can be solved using Distributed Ledger Technology (DLT). The integration of IoT devices and Distributed Ledger Technology is getting more recognition in distributed device settings. In a Distributed Ledger IoT environment, DLT manages a distributed public ledger that stores the communication and transaction data of multiple parties without requiring a trusted central authority. A prime example of the IoT and DLT integration is the IOTA Tangle. IOTA Tangle is a new and world's first distributed ledger technology specially developed for IoT devices. In its core structure, it stores data in a directed acyclic graph. IOTA is not only built to ensure secure communication among the IoT devices but also to provide some extra leverage to the IoT devices compared to the traditional blockchains. Regular blockchains have dedicated miners and they are mostly fee-dependent for making any kind of transaction. IOTA ledger is more scalable, has neither transaction fees nor traditional miners, and it also has the technology to

protect the network against Quantum Computer's attack. Before widely bringing the IoT devices into the IOTA network, we need to justify the feasibility of IoT and IOTA integration making a use case prototype to ensure the security of the IoT devices in real-time experiments. Hence, the main goal of this research is to integrate IoT sensor devices and IOTA Tangle to scale up the security of the IoT infrastructure and at the same time execute an efficiency test of the network.

Contents

List of Figures

List of Tables

Acronyms

AES Advanced Encryption Standard. 15

AWS Amazon Web Services. 16

CPU Central Processing Unit. 29

DACT Data Attach Conformation Time. 58

DAG Directed Acyclic Graph. 20

dApps Decentralized Apps. 22

DC Direct Current. 33

DDoS Distributed Denial of Service. 6

DLT Distributed Ledger Technology. ii, 2, 5

GB Giga Byte. 29

GND Ground. 31

GPIO General Purpose Input/Output. 29

GPU Graphics Processing Unit. 29

IaaS Infrastructure as a Service. 16

IIoT Industrial Internet of Things. 7

IoT Internet of Things. ii, 2

IP Internet Protocol. 6

LPDDR Low Power Double Data Rate Random Access Memory. 29

MAM Masked Authenticated Messaging. iv, 26

MWM Minimum Weight Magnitude. 26

NFC Near Field Communication. 13

PaaS Platform as a Service. 16

PKI Public Key Infrastructure. 6

PoW Proof of Work. 9, 19

RAM Random Access Memory. 29

RFID Radio Frequency Identification. 12

RL Read Latency. 58

RRT Response Receive Time. 58

SaaS Software as a Service. 16

SBC Single-board Computer. 7

SDST Sensor Data Submit Time. 58

St. Dev. Standard Deviation. 58

TCP Transmission Control Protocol. 6

TL Transaction Latency. 58

TLS Transport Layer Security. 15

VCC Voltage Common Collector. 31

WSN Wireless Sensor Networks. 13

Acknowledgments

I would like to express my cordial gratitude to Dr. Rasit Eskicioglu and Dr. Cuneyt Akcora for being my mentors throughout this whole research. The amount of time they have given me to guide me in improving my research skills is invaluable. I would also like to thank both of my thesis committee members Dr. Bob McLeod and Dr. Sara Rouhani for their valuable suggestions while completing this thesis. I am grateful to all my labmates for their support, special thanks to my labmate Baha Rababah for his help and suggestions. I would also like to thank all the faculty members of the Computer Science Department who helped me to increase my skills while completing their courses. Finally, I would like to acknowledge the support of the staff members at the Department of Computer Science and the University of Manitoba for providing me the amazing opportunities to fulfill my dream to get a Master's degree in Computer Science.

This work is dedicated to my beloved parents.

Chapter 1

Introduction

With the immense growth of the internet, smart devices are getting popularity worldwide which is part of the Internet of Things (IoT). In the IoT infrastructure, mainly low-powered sensor-dependent devices remain interconnected through a private or public network [1, 2]. The functionality of these devices can be controlled from anywhere in the world through the internet. IoT devices range from simple smart TV to Internet-based large machines. As these IoT devices have limited computational power, storage, and network capacity, they are more vulnerable to be hacked. In recent years huge efforts have been applied to solve the security issues of IoT infrastructure.

Blockchain or Distributed Ledger Technology (DLT) is one of the new revolutionary technologies in recent years. Without the need for a trusted central authority blockchain's core consists of a distributed public ledger that stores two parties' transaction data [3–5]. It is openly shared among users and it creates an immutable record [6, 7] of data. That's why blockchain technology provides strong guarantees against

data manipulation in a network of users who do not and cannot trust each other [8–10]. This capability of blockchain technology has grabbed the attention of diverse industries [11, 12].

Blockchain or DLT platforms have been developed to embed software code, called a smart contract, in transactions where the code can be called by any participant in the network [13–15]. Smart contracts are publicly verifiable self-executing Turing complete contracts that contain code and arguments. Merging smart contracts with a currency to facilitate transactions, these platforms have the potential to change how personal smart devices, the Internet of Things devices, cars, and smart homes can communicate with each other to solve daily problems in societies. A typical example can be a car visiting a technician's shop at night to fix an issue without supervision from the car owner. The car can share its private data with the technician's server and record this interaction in a DLT or blockchain. The car can also pay for the repair costs by using cryptocurrency. If the technician charges too much or fails to provide a good service, the car owner's auditing software can identify this charge. This data cannot be modified by the technician to evade fraud detection. In reality in all these steps, neither the technician nor the car owner is needed to store the data, or analyze the results. All payments and interactions can be carried out by using a specialized distributed ledger. These advantages can easily be implemented for certain developments and improvements in traditional businesses. As a result, in recent times blockchain technology has grabbed the attention of both consumers and industries. Its application can be seen in many areas such as in digital finance, supply chain management, health care, and food safety.

Conventionally IoT devices share their data through a cloud-based centralized infrastructure. If any security issue arises in this type of communication model, a single point failure can interrupt the function of the whole system. Adoption of a distributed model can reduce the potential detriment. IOTA Tangle is a Distributed Ledger Technology (DLT) specially designed to run cryptocurrency and IoT devices' communication on its network. As the distributed ledgers have the potential to enhance the security of the Internet of Things infrastructure it's necessary to test this type of distributed ledger's feasibility to successfully integrate with IoT devices. As IOTA Tangle is the first generation of DLT as its type, it might have some potential integration issues with IoT. The main goal of this research was to make a prototype integrating IoT devices and IOTA Tangle to find out potential pros and cons and check its data communication efficiency.

Chapter 2

Related Work

Research work is still very limited on Distributed Ledger Technology (DLT) and IoT integration to enhance security. Some of the related research works which are discussed here will help us to understand how researchers are trying to integrate blockchain or distributed ledgers with IoT devices, the architecture of blockchain-based IoT devices, and the security problem they have in this framework.

2.1 Security Issues in IoT Infrastructure

IoT technology works as a combination of different information and communication technology to achieve multiple goals. It consists of electronic sensors, single board computers, smart grids, vehicle networks, wearable technologies, cloud computing, actuator technology, wireless sensor devices, etc. To solve a complex problem integrating a lot of computer technologies sometimes makes a whole system incompatible to work with every component flawlessly from a security and privacy point of

view. It happens because in the IoT world we get to see a lot of new devices coming in the market frequently which run on different security and communication protocols.

Kumar et al. [16] describe the common issues and challenges for IoT devices. Among these issues, security is one of the main concerns and it happens because of lack of powerful hardware components, lack of standards metrics, lack of security laws, fewer efforts in developing standard protocols, centralized system, less data protection, lack of encryption, interoperability issues, limited investments to develop sustainable technical resources. Because of this immature stage of IoT, it is susceptible to different types of threats and attacks. Damghani et al. [17] discuss and classify these attacks, among them some common attacks are Distributed Denial of Service (DDoS) attack, Wormhole attack, Sinkhole attack, Flood attacks, Spoof attacks, Sybil attack, Man in the middle attack, Mirai botnet attack. Kolias et al. [18] describe the most common DDoS attack and Mirai botnet attack in IoT infrastructure. A DDoS attack occurs when an attacker blocks an IoT device to provide its services. The attacker sends malicious data or packet requests from multiple systems to a specific device to overwhelm the system by wasting the system's internet bandwidth, CPU, and RAM capacity. It disconnects the devices from the application layer by denying services. The authors discuss here Mirai is a malware that infects smart devices and turns all of them into a network of bots which is known as a botnet. This botnet launches a DDoS attack on a system by scanning random public Internet Protocol (IP) addresses through Transmission Control Protocol (TCP) ports. To address these problems many solutions are proposed. Public Key Infrastructure (PKI) is one of the newest approaches. Diaz-Sanchez et al. [19] describe how a PKI authentication sys-

tem work for ensuring IoT security. It works based on digital certificates that verify the identity of smart devices. It has three key components digital certificates, certificate authority, and registration authority. Two machines communicate in this system by verifying themselves using the digital certificates which are issued by a certificate authority (e.g., Let's Encrypt) through registration authority which is stored in an encrypted database. In this system, the device verification thing is dependent on a third party which is the certificate authority and if its private key is attacked, then the system will be at risk.

Because of the vulnerabilities, all these problems are occurring that have already been mentioned here, blockchain or distributed ledger solves most of them. This technology is decentralized, it has a robust authentication system, it protects the data through encryption and its distributed common ledger is immutable which means if anybody tries to alter the data in this ledger that's impossible.

2.2 DLT and IoT Integration

Bahga et al. [20] introduce a blockchain-based Industrial Internet of Things (IIoT) framework. This framework helps IIoT devices to work together using the cloud through a blockchain network. These IIoT devices use a Single-board Computer (SBC) having control and communication capabilities to cloud and Ethereum blockchain. Data is sent by these devices to the cloud for storage and analysis. Transaction of data among all these devices is done through the blockchain network. The execution of this process is done using a smart contract. For smart diagnostics and machine maintenance, the authors implement a platform using the Arduino Uno

board. The platform is implemented as a proof of concept using the Ethereum smart contracts.

Conoscenti et al. [21] mention four cases of IoT and blockchain out of many blockchain cases which are data access control management and an immutable log of events [22], IoT data trading [23, 24], symmetric and asymmetric key management [25, 26]. Friese et al. [27] discuss challenges for identity in IoT. These challenges are authentication and authorization, governance of data and privacy, identity relationship, and ownership.

Christidis et al. [28] review the application of smart contracts for IoT devices. The authors discuss how smart contracts can support the autonomous workflow of IoT devices and make sharing services effective among them which is proposed by Brody et al. [29]. In one scenario the authors elaborate on how assets can be tracked of container shipment using smart contracts and IoT. The authors show how supply chain management, e-trading, billing, shipping can be benefited using blockchain in IoT. Blockchain smart contracts can make a revolution in the automatic buying and selling of energy among smart meters. For energy trading, a user-defined policy can be used through smart contracts.

Data counterfeiting security issue is common to occur using smart meters. Lee et al. [30] propose to use blockchain smart contracts to prevent data forgery and personal information infringement. A blockchain anonymity enhancement technology is used here that is called Zero-Knowledge proof to prevent this kind of security threat.

Slock.it [31] works on the security of IoT devices using blockchain smart contracts. It works on smart electronic locks ("Slocks") where a token is used on a device to

unlock. These token are bought on the Ethereum blockchain [32]. It is a smart contract based public cryptocurrency blockchain network. If an owner of a house or a car wants to rent it, the owner can use Slock to set a specific time and price to access that electronic lock which is used for that house or car. For renting those, a person can use a mobile app to identify the slock and pay the particular amount in Ethers, communicating with the Ethereum blockchain network.

Liang et al. [33] work on the security of drone communication data. While a drone is collecting and transmitting data, it should be done securely. The authors propose a distributed solution here that is blockchain-based. They don't register the drone itself to the blockchain rather anchor the hash data records to the blockchain network. These data are collected from the drone. After that, a blockchain receipt is generated for each data record stored in the cloud. The process is very effective for data integrity and cloud auditing and it is reliable for drone data assurance and resilience with the scalability for multiple drones.

Dorri et al. [34] propose a novel instantiation of blockchain for smart home security and privacy. In their design, three core tiers are considered which are smart home, cloud storage, and overlay. Smart devices are located inside these smart home tiers and centrally managed by a miner. The miner is responsible for handling internal and external communication with the home. The miner contains a private and secure blockchain for controlling and auditing communication data. In this mechanism, the authors eliminate the concept of Proof of Work (PoW) [35] which is a cryptographic puzzle and by solving this puzzle a node gets the opportunity to mine it to the blockchain. Thus, there is no need for coins. To maintain smart home security this

framework relies on a hierarchical structure. Through their implementation of this framework by a simulation they make it clear that this type of blockchain-based system can achieve the fundamental security goals of integrity and confidentiality.

Khan et al. [36] review IoT security using blockchain technology and the challenges in implementing it for smart devices. The authors show that IoT devices are structurally vulnerable to security issues as they have constrained resources. These devices use limited hardware to reduce the power consumption as they remain active most of the time and because of that, the developers can't use very sophisticated software on these limited resource devices. As IoT devices are diverse in terms of resources, a global robust mechanism can't be implemented easily for securing IoT layers. The authors also categorize the security issues depending on the IoT device layers. They are high-level, intermediate-level, and low-level. The researchers show that a blockchain smart contract-based distributed ledger can leverage IoT security if it can be designed in a proper way for low-resource devices.

Even though there isn't that much research available on protecting the security for specialized smart contracts developed for IoT infrastructure but there is some research for enhancing the security of cryptocurrency smart contracts [37]. Kalra et al. [38] analyze and discuss the tools and verification methods to detect buggy smart contracts [39, 40]. These papers mention that it is very challenging to eliminate the vulnerabilities of cryptocurrency-based smart contracts. To implement the blockchain in IoT along with understanding the security issues of an Ethereum smart contract, it is very necessary to understand how the smart contract's bytecode [41, 42] works. This bytecode includes constructor logic and parameters of a smart contract and it

is executable on Ethereum Virtual Machine.

2.3 IOTA Structure and Integration with IoT

Silvano et al. [43] discuss the ecosystem and the mathematical foundation of IOTA Tangle which is a whole different type of distributed ledger technology that acts not only as a cryptocurrency but also as a distributed communication protocol for IoT devices. In this paper, the authors emphasize the scalability of the IOTA Tangle for IoT infrastructure.

Shabandri et al. [44] theoretically show how an automatic and secure payment system can be built for car renting services using IoT and IOTA Tangle. Similarly, Vieira et al. [45] also demonstrate how an IOTA-based automatic payment system can be implemented for the public transport system. They show that when the passengers get into the buses or trains they can be detected using Bluetooth beacons and using the IOTA Tangle a charge will be deducted communicating with passengers' smartphones according to their journey destination. As the payment system is decentralized here, if the company's payment server ever goes offline that will still be able to get the missing payments data when it will be online again. IOTA Tangle provides a very strong access control mechanism [46, 47] for this type of automatic payment system.

In the investigation of Cyber-Forensics, log data of any computer system is very important and its integrity is more important than this. Bhandary et al. [48] demonstrate how investigators can take advantage of a decentralized log system built on IOTA Tangle to ensure the integrity of the log file.

Chapter 3

Background

3.1 Internet of Things

Internet of things or IoT devices is simply those digital devices around the world that are connected to the internet to collect and share data. The idea of IoT devices came in the early 1990s. But because of too big computer chips and inefficient ways of communication the IoT industry couldn't flourish. Next, with the gradual growth of small and powerful chips, this technology took almost one decade to catch up with the vision of a connected world. In 1999, Kavin Ashton first used the term "Internet of Things" [49]. He made the concept popular by showing how IoT can be created by merging Radio Frequency Identification (RFID) and sensors to objects. There is no specific definition of IoT devices. Different authors [50–52] have used this term differently keeping the general concept of IoT as same. Very common IoT devices are smartphones, smart locks, single board computers, smart bulbs, actuators, sensors, smart home assistant devices, cloud servers, etc. IoT devices don't have any standard

architecture. IoT devices range from smart home devices to different industrial devices and are manufactured by thousands of companies according to the needs of different industries. Even though they don't follow any specific architecture but generally they are seen having three basic layers.

• Physical layer - This part consists of different sensors to find and gather data from the environment. RFID, Bluetooth, Near Field Communication (NFC), Wireless Sensor Networks (WSN) are embedded in this layer to identify other smart devices and communicate with them.

• Network layer - All the network protocols are operated in this layer. This layer transmits and processes data to connect with other smart devices and servers.

• Application layer - Application-specific services are given to the users from this layer. Sensor's collected data are stored in this layer to process and measure and actions are taken from this layer based on the processed data.

3.2 Importance of Security in IoT Infrastructure

The Internet has become an inseparable part of the 21st Century. The rapid growth of the fastest internet access is seen all over the world. With easy access to the internet, the number of connected devices is increasing unbelievably. The concept of the smart city and smart home [53, 54] is fully based on the massive use of IoT devices. According to a recent survey of "Statista" (a popular market and consumer data statistics company) in every coming year, more than one billion connected devices will be added to the realm of the internet and by the next decade, the number of total connected devices will be around more than 25 billion.

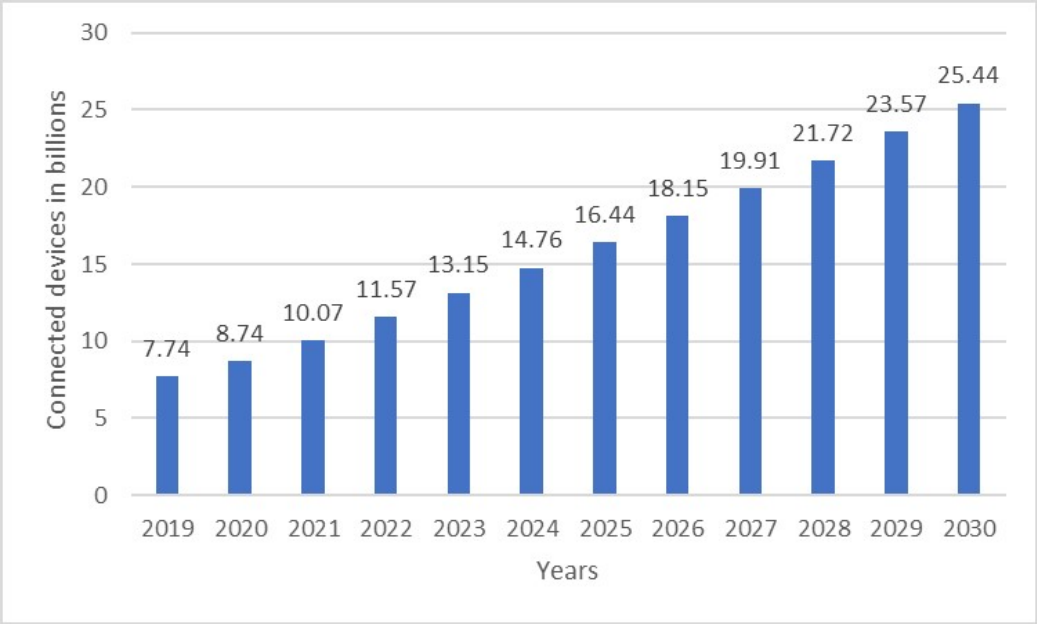

Figure 3.1: Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030 (Source: Statista 2021)

These IoT devices will be everywhere to collect a huge number of data and simultaneously share it through the internet to ease the modern lifestyle. Such a big number of devices dealing with very sensitive data can any time be targeted to breach their security system. This type of attack is very common and happening now and then all over the world. Mirai Botnet attack [18] is one of the largest DDoS attacks that happened with smart devices which has already been discussed in section 2.1. IoT devices face security issues for different reasons. If we see some of the major reasons we will understand why it is so important to ensure the security of data transacted by IoT devices.

• Heterogeneity - IoT devices have less powerful and diverse hardware and this creates a huge variation in the performance of devices based on the protocols, platforms, policies. The absence of a common security service [55] is the major issue of

heterogeneity which leads to a lack of interoperability.

• Resource Constraint - It is hardly possible to directly implement legacy security services [56] such as Transport Layer Security (TLS), Advanced Encryption Standard (AES) in IoT devices because of their less powerful resources.

• Old Operating System - IoT devices sometimes use the backdated version of the operating systems. This helps the adversaries to exploit the devices using the publicly known vulnerabilities of the outdated operating system.

• Integrated Security Issue - IoT devices mostly don't come with any built-in antivirus software that makes them target to be infected with malware. That's how an attacker gets access to the sensitive data of the device and can use the device to originate a botnet attack.

• Insecure Protocols - IoT devices sometimes use some protocols which lack built-in security. Telnet is one of the protocols which is out of fashion because of its in-built security risk.

• Lack of Update - Users hardly care about updating their internet-connected bulbs, ovens, air conditioners, toasters, etc. To close the security holes these devices need periodic updates.

• Vulnerable Deployment Locations - IoT devices are sometimes located in a public place where it is very easy to reach them physically. An attacker can take advantage by bypassing the existing defense system of the device.

Because of all these issues, IoT devices are prone to the different botnet and Distributed Denial of Service (DDoS) attacks that lead to a huge threat of losing sensitive data transacted by these devices.

3.3 Cloud Computing

Cloud computing is the ubiquitous utilization of internet-connected remote servers to store, process, and manage data [57]. It is an on-demand computing resource sharing service through the internet where a user gets the desired storage and sophisticated computing processing capabilities. In recent years cloud computing has grabbed the huge attention of information technology-based businesses. Big IT corporations such as Amazon, Microsoft, Google, IBM are the main market leaders which are successfully providing cloud computing services all over the world. Cloud computing service is classified into three main categories [58].

• Software as a Service (SaaS) - When applications are running on a cloud infrastructure provided by different cloud service providers and those applications are used through the browser that computing model falls into this category [59]. Example: Dropbox, Salesforce, Google Workspace.

• Platform as a Service (PaaS) - In this cloud computing model, a remote operating system is provided along with frameworks for software development [60]. For developers, this is a convenient way to access the on-demand processing and memory power in a development environment. Example: Google App Engine, Apache Stratos, AWS Elastic Beanstalk.

• Infrastructure as a Service (IaaS) - IaaS is instant computing infrastructure. Clients get on-demand processing power, cloud storage, and network resources to manage an operating system on top of the cloud [61]. According to the need of the client, they can set their policy (e.g., when to turn on/off the server) for the installed virtual operating system. Example: Amazon Web Services (AWS), Cisco Metapod,

Microsoft Azure.

IoT devices are an integral part of these cloud computing models. Interacting with the environment, IoT devices collect different sorts of data and send it to the cloud to store and manage. Users are regularly accessing these stored or live data from their end through browser or mobile applications. In a smart city, every Internet-connected device will share its data through the cloud with other devices. Because of the many vulnerabilities of internet-connected devices discussed in section 3.1 in the process of sharing the data, the security and privacy of the devices can be compromised and the data can be stolen or manipulated through the process. Hence the receiving end will never get the data or will get the tempered data.

3.4 Distributed Ledger Technology (DLT)

Cryptocurrency is one of the significant inventions of the last decade. The underlying technology of these digital currencies is known as Blockchain that is a variant of Distributed Ledger Technology (DLT). DLT is an evolving technology where a distributed database with specific properties is maintained across a distributed network of computers to record and share information [62, 63]. This common database is known as the ledger and each of the distributed computers is referred to as the node. DLT is a peer-to-peer data sharing process through the help of the internet where the data is not shared through or stored in a centralized server like cloud computing. That's why there is no way to hack any central server which may lead to the failure of a full data sharing system. In a DLT based network system when data is shared by any node that is recorded in a common ledger and this change is reflected

throughout the whole network. This digital ledger is immutable which means that no one can change the data recorded on it as it is constrained by specific policies of the network. This immutability is secured using a strong cryptographic algorithm that keeps up the integrity of the network. All the nodes of a DLT based network always have an identical and detailed copy of the ledger for any specific time. If a node is offline or online that doesn't matter in this network as whenever the node comes back online, it gets its ledger updated with all the activities recorded with timestamps. As the data recording and sharing process through this ledger is encrypted using a complex cryptography-based mathematical system, it is very secure against security and privacy threats. As DLT mainly evolved based on the digital currency market it maintains two core components to protect the validity and efficiency of the digital currency. The first aspect is it ensures a central authorityless information recording and exchanging system across multiple consenting parties that helps the system to be more autonomous. The second core component is it helps to avoid the double-spending problem of the same digital asset. In a central authorityless system, double-spending of the same digital asset can be very high as no one knows each other on this type of network.

3.4.1 Types of DLT

Generally, distributed ledgers can be divided into two categories. One is permissioned and another is permissionless [64]. In permissioned DLT nodes access the network and modify the ledger with the permission of the central entities verifying their identity. On the other hand, permissionless DLT always gives authority to all the nodes to access and add any information to the ledger anytime. Following the protocol of the network, the nodes can locally execute new addition to the network and that new update is communicated throughout the network. In both of the categories to ensure the consistency of the newly added data, the nodes play an important role to validate the modification on the ledger collectively through the execution of a consensus algorithm. Most recently we see a special type of hybrid DLT that combines properties of both the permissioned and permissionless DLT. Based on these basic mechanisms there have emerged quite a few DLT [65]. To understand it better we will discuss some of them.

• Blockchain - Undoubtedly because of the massive popularity of cryptocurrencies Blockchain is the most renowned DLT in the world. Cryptocurrencies are working successfully on Blockchain technology. When peer-to-peer transactions happen, it is transmitted across the world's computers in its network. To confirm the validity of the transaction the computers solve a mathematical puzzle which is known as Proof of Work (PoW). The nodes that complete PoW are called miners and the whole process is called mining. Once the transactions are solved they are counted as legitimate transactions and accumulated together in the form of a chain of blocks [66]. A hash function is used to distinguish the transaction blocks in the ledger. In digital

currencies, all the transactional records such as date, time, amount are stored in the digital ledger. These information blocks also contain the sender's information using a unique digital signature that ensures anonymity. Senders initiate the encrypted transaction using anyone's public key and to access that transaction the valid receiver needs to decrypt it using a specific private key. Example: Bitcoin network. Blockchain can be of two types.

Public Blockchain: A public blockchain is fully decentralized and open for anybody to participate in the network and its consensus process. Public blockchains have common regulations for all the participating nodes in the network. Anyone can check the public blockchain's ledger that makes this type of network transparent but every user remains anonymous in this network. Anonymous users might involve in any illegal activities in this type of network. Example: Ethereum.

Private Blockchain: The use and authorization of a private blockchain are limited within a single organization. This type of network system is built for a company's internal communication. Private blockchains have their specific regulation on the network and it is partially decentralized. Because of having limited users, private blockchains are fast in terms of communication. As private blockchains have internal authority to control the network there is no chance of illegal activities. Example: Hyperledger

• Directed Acyclic Graph (DAG) - DAG is a different structured DLT than the regular Blockchain. In terms of scalability, this DLT supports different types of functionality on its network. A huge number of nano-transactions are possible in this network. In its consensus mechanism, a node that is initiating a transaction has

to validate two previous transactions on the digital ledger to confirm its transaction [67]. As the transaction generators are doing the mining here, there is no need for any third-party miners like blockchain. Attendance of more nodes makes the faster transaction process on this type of DLT. Which companies are building their DLT based applications focusing on huge transactions they mostly prefer DAG. Example: IOTA Tangle.

• Hashgraph - Hashgraph uses the gossip-about-gossip protocol to validate the transactions. Each nodes gossip about the transactions in order to create a directed acyclic graph to make a time sequence of the transactions instead of creating blocks like blockchain [68]. Gossip messages consist of transactions, digital signature, timestamp, and the hash value of previous transactions. This type of DLT doesn't store all the transaction information forever that's why less storage is needed for the nodes in the network. Example: HBAR

• Tempo (Radix) - This DLT follows almost the same working mechanism of DAG but in a different approach. It works based on Leslie Lamport's logical clock theory. In this type of DLT, a node is different than the regular users whereas in a DAG they are all the same. A node is responsible to provide the resources for the network. Nodes provide the service of validating transactions following the same consensus mechanism of DAG and resolve conflicts of different events in exchange for a fee that is in proportion to their service. The user of this network is called a client who needs to request all the services through these nodes. This technology is still not open-source. To run a node its network does not require expensive computers and huge mining power that ensures less power consumption for the network. Now its ledger

is working on a test net. Example: Rad

• Holochain - This project has combined the working mechanism of BitTorrent, Blockchain, and Github. It is mainly an open-source cryptographically secured framework to develop Decentralized Apps (dApps) [69]. This DLT doesn't need any miners as all the nodes maintain their own ledgers. That's why it has no congestion of transactions. Tampered data is invalidated in this network by finding out the inconsistency between the data and hash. Example: Holo

3.4.2 Possibilities and Challenges of DLT

DLT emerges to eliminate the trusted third parties that sometimes help us to process and record our bartering by taking a portion of the fee. As exchange of value is a daily necessity of human life and DLT can get that job done so securely without a trusted third party in a form of a digital token, its possibilities are huge [70]. Because of DLT, The monetary system is on the verge of a new era of decentralization. One aspect of this decentralization is it can save a lot of money to manage a country's monetary system and make the system very fast. Different DLT is having a programmable feature that is called a smart contract where people can build different decentralized applications and the users can easily make transactions in digital currencies as the service fee. In a cloud computing-based service a server can get hacked at any time which may lead to the failure of the whole service. But in decentralized applications, there will be no single point of failure. As IoT devices are being autonomous they can securely communicate among them through DLT. For example, a car can use a gas or charging station and after taking the service the car will automatically pay for

the gas or electricity and all of this can happen through a decentralized app running on a DLT.

DLT has got enlightenment because of the hype of cryptocurrency. As cryptocurrencies still lack proper regulations so does the DLT. Lack of regulatory infrastructure in DLT can compromise the rights of the users. The technology is particularly evolving to elevate the digital currency and there is still no solid impact on other possible areas. It creates doubt about the scalability of this technology. Without Blockchain, other DLTs are still too immature and have their flaws. Balancing the data security with necessary transparency is still a challenge for DLT.

3.4.3 IoT and DLT Integration

Even though most of the DLTs are in the early stages of their development, still because of DLTs transparent data recording system on an immutable ledger, IoT devices are getting traction to be integrated with DLT [71]. In recent days companies are so much dependent on cloud computing for storing data and creating software as a service. Most of these cloud services are Amazon's AWS, Microsoft's Azure, and Google Cloud. Averagely these services can cost 23.55USD/TB where distributed ledgers can be free. Even though there is no other alternative of cloud services for storing big data, but sensitive small data like currency transaction data, IoT devices' peer-to-peer communication data, medical record data, etc can be vulnerable to store in the clouds. In the related work of this thesis, some of the instances have already been discussed. Still, we will discuss some of the use cases here to demonstrate their importance over typical cloud integration.

Because of too many party's involvements, moving freight is a difficult task. Data integrity is the top priority. IoT devices integrated with DLT can ensure the storage of authentic temperature, arrival time, real-time location data of shipping containers so that all parties can reliably trust each other to move the products on time efficiently. In many places, IoT devices track the state of the safety of sensitive machinery and report it to the authorities. Sometimes the government entities also need to verify compliance. IoT data sent through the immutable DLT can be tamper-free in this case and the proper monitoring authorities don't have to be worried about the trust issue of the real data. Different sensitive components such as medical equipment, flammables carried by aircraft and automobiles are needed to be shared their live temperature, pressure data with regulatory agencies or manufacturers. In these scenarios, a shared ledger can help the authorities to maintain the safety measures reliably. Distributed ledger has three main effective aspects over cloud storage [72];

• User Ownership of Data : In distributed ledger system the ownership of the data remains within the users whereas public data can be misused by cloud service providers.

• Client-side Authentication : As there is no server system available here so there is no single source of massive user names and passwords that eliminates the leak of user credentials.

• Transparency Accountability : The user can see every action that is taken in a distributed ledger system whereas in a cloud system admin has authority over the data that might get modified by mistake or intentionally.

Nowadays, most of these IoT devices interactions discussed here are done through

the cloud and we know the potential threat and vulnerabilities of IoT and the cloud. If IoT devices data can be shared through a secure, scalable, and efficient blockchain or distributed ledger system then it can be possible to avoid the possible vulnerabilities or potential data privacy and security issues of conventional clouds.

3.5 IOTA Tangle

IOTA Tangle is a novel distributed ledger technology that has been specially developed for IoT devices. The IOTA network is designed for exchanging data values between humans to machines and machines to machines. In a regular blockchain network (for example Bitcoin) different types of transactions are stored in blocks and blocks are sequentially connected maintaining a chain. IOTA Tangle is not the same as the traditional blockchains. Instead of following the traditional block-based distributed ledger model, it is constructed based on a Directed Acyclic Graph model (DAG) [73]. There are no traditional miners here for proof of work who complete mining for incentives. Rather, the mining process is relegated to each transaction creator. Each new or unconfirmed transaction known as Tip (filled gray box on Figure 3.1) needs to validate 2 previous non-validated transactions (white boxes on Figure 3.1) as proof of work. While creating the transaction a node (computer) signs in with its private key. This digital signature is known as Winternitz one-time signature [74]. To select the other two unconfirmed transactions this node uses the Random Walk Monte Carlo algorithm [75]. To validate the other two transactions and complete the proof of work, the node solves a cryptographic puzzle (Hashcash). Because of this mining mechanism, the network is feeless for any sort of transaction. While a node

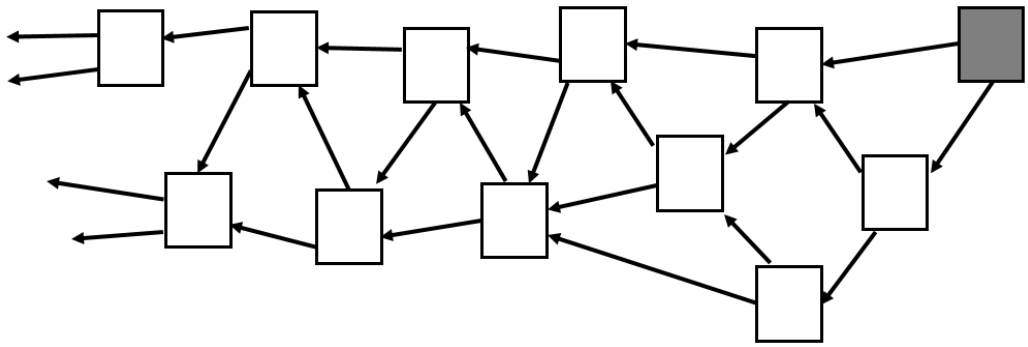

Figure 3.2: IOTA Tangle

is completing a proof of work task, it is possible to define on this network that how much work should be done by specifying a Minimum Weight Magnitude (MWM). IOTA has two types of networks, one is the Mainnet and another one is Devnet or Testnet. On the Mainnet, the MWM must have to be at least 14 to keep the proof of work difficulty level strong as all the main token-based transactions happen here. The Devnet is for testing and experimenting with different types of transactions. The graph is always non-circular on this Tangle to prevent the repetition of the same transaction.

3.5.1 Masked Authenticated Messaging (MAM)

Masked Authenticated Messaging (MAM) is a data communication protocol on IOTA Tangle which allows IoT devices to publish their encrypted data streams as a transaction on the tangle. This communication with the Tangle network is considered as zero value transactions as there is no fee or IOTA token is used while communicating with the network. Another interesting thing for this network is, if the network gets more and more transactions it becomes faster to validate the trans-

actions as it gets more nodes to validate the transactions. Previously, IOTA had a centralized node called coordinator to approve the valid transactions. A decentralized IOTA Tangle network was released last year even though its development is still in progress. On the IOTA Tangle network, a low-configuration IoT device can act as a light node without storing huge transaction data, and connecting with a full public node it can keep itself updated with the distributed ledger [76]. A full node is a powerful computer on the Tangle network that always stays connected with other neighbour nodes and keeps its ledger updated by storing huge transaction data that enables it to do the proof of work fast. IOTA Tangle supports mechanisms to balance the difficulty of proof of work in between light node and full node to get almost a similar throughput for a specific transaction [77]. While sending a message to the IOTA Tangle, a public full node's address can be hardcoded on a local node to keep the local node updated about the ledger without communicating with many neighbour nodes. Many full nodes are operated by the IOTA Foundation. Which nodes keep all the history of old transactions that are called Permanodes. To save the storage, periodically these old data are archived or removed which is known as Snapshot in IOTA. When a message is sent using MAM protocol the message is attached to the Tangle using a channel and an address is generated to read the message. This address is known as the Root address or Seed. These channels are generated by the Markel tree which is nothing but data streams attached with the Seed. Kerl Hash Function generates the addresses which have a private key, index, and security level. This index is used to create multiple addresses or sub seeds from the main Seed address that allow a user to get a continuous message where each message contains the root ad-

dress of the next Markle tree. This index value remain between 0 to 9, 007, 199, 254, 740, 991. To create a sub seed, Kerl Function is applied on seed+index. This root address is 81 characters long which is made in combination of 26 upper case alphabets and the number 9. An example of a Root address is, "PWCRTMNOBLSFHTTFXV9ACRHDM9XZVMFPVYPRVRFFAHGMPYTPIBHLZFYXTOI9JOXEALAVXSPJVMKDIAKLN" To generate each character in seeds, IOTA uses a ternary number system which is a combination of these three numbers -1, 0,1. Each of these 27 characters has its correspondent value in the combination of these 3 numbers. For example, the alphabet A represents this combination 1, 0, 0 which is equivalent to decimal number 1. A combination of these 3 digits is called a Tryte and each of these 3 numbers is called Trit. Even though this type of numbering system is uncommon in blockchain IOTA considers it a strong security system as a Tryte can be made of 33 possible values. IOTA has 3 security levels in between 1 to 3 to protect against brute-force attacks. As we mentioned before while creating a transaction a node sings with its private key this signature length is determined by the security level. For security levels 1, 2, 3 private key signature length is respectively 2187 trytes, 4374 trytes, 6561 trytes. The more the number of trytes the more strong the security of an address. Message can be sent in public or private mode. Everyone can see the messages sent in a public mode that is not possible in a private mood. A password can be set on private mood messages to restrict the viewer and this mode is known as a restricted mode. This password is known as a side key in IOTA. After sending an encrypted message to IOTA Tangle only a Root address and the previously hardcoded side key are needed to receive the message on a data receiving end.

3.6 Raspberry Pi

Raspberry Pi is a single board credit-card-sized all-in-one computer. It can be plugged into a full-size monitor and can be operated through a regular mouse and a keyboard. Raspberry Pi is made by Raspberry Pi Foundation and the first Raspberry Pi came into the market in February 2012. This device is capable of doing the task of a full desktop computer such as playing high-definition videos, internet browsing, playing games, doing full-fledged programming, etc. Till now Raspberry Pi has released several models with several configurations. It uses an ARM-based Central Processing Unit (CPU) and onboard Graphics Processing Unit (GPU). Most of its model has HDMI port, 3.5mm audio jack, ethernet, USB ports, interactive General Purpose Input/Output (GPIO) pins to connect with external sensors. For wireless communication, most of the recent models have built-in Bluetooth and WiFi for connectivity and run on a 5V/2A power. The most recent model Raspberry Pi 4 Model B has a 64-bit quad-core processor with a maximum speed of 1.4GHz and 8GB LPDDR4 RAM. The operating system used on this device is Raspbian which is a Linux-based operating system.

Raspberry Pi is so popular and widely used for multiple purposes in IoT development because of its low cost and powerful all-in-one architecture [78]. There are some other popular microcontrollers for IoT development such as Arduino and NodeMCU. But they lack high memory and computational power and they also only work with a specific programming language. Through Raspberry Pi's General Purpose Input/Output (GPIO) pins most of the sensors in the market can be connected and programmed according to needs. Most of the popular programming languages

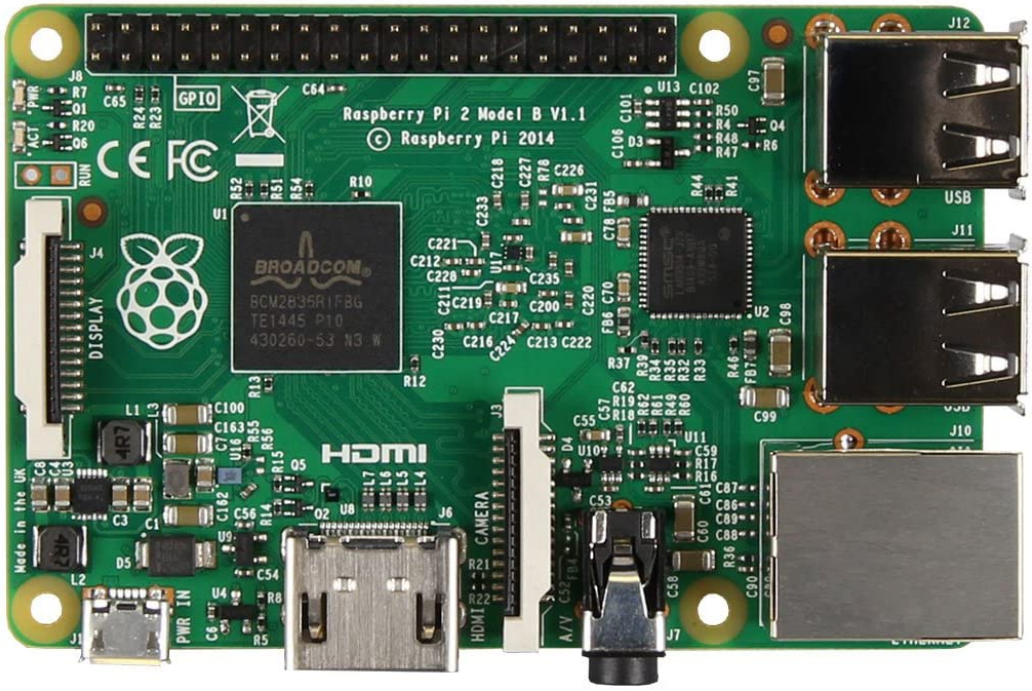

Figure 3.3: Raspberry Pi 2 Model B (Image Source: www.raspberrypi.org)

are supported in Raspberry Pi's architecture that's why it becomes comparatively less complicated to program the connected sensors. Along with the data transfer pins, this device has power supply pins for the sensors that can supply 3.3V with a maximum current draw of 16mA. General-purpose programmable pins can be used as either input mode or output mode. In our experiment, we used the Raspberry Pi 2 Model B which has a 900 MHz quad-core Arm Cortex-A7 CPU with 1GB RAM. For connecting with other devices it also has 1 Ethernet port, 4 USB ports, 40 GPIO pins, Full HDMI port, 3.5mm audio jack, Micro SD card slot, Camera interface, and Display interface.

3.7 DHT 11 Temperature and Humidity Sensor

DHT11 is a low-cost basic temperature and humidity sensor. It is widely used in IoT developments with different microcontrollers such as Arduino, Raspberry Pi [79]. To measure the temperature this digital sensor uses a thermistor and to measure the humidity it uses a capacitive humidity detecting element. Humidity determines the water vapor level in the air. This sensor has huge industrial use as temperature control is a very important thing in most of the industries such as food industries, chemical industries, hospitals, weather stations, and also humidity control is a significant thing in semiconductor industries and for air conditioning systems. This sensor can measure temperature between 0 to 50 degrees Celsius with an accuracy fluctuation of 2 degrees. Its humidity measurement range is 20% to 80% with 5% accuracy. This sensor provides 1 reading per second taking 2.5mA of electricity and operates in between 3 to 5 Volts. It has 4 connection pins and they are Voltage Common Collector (VCC), Ground (GND), Data Pin, and Not Connected pin.

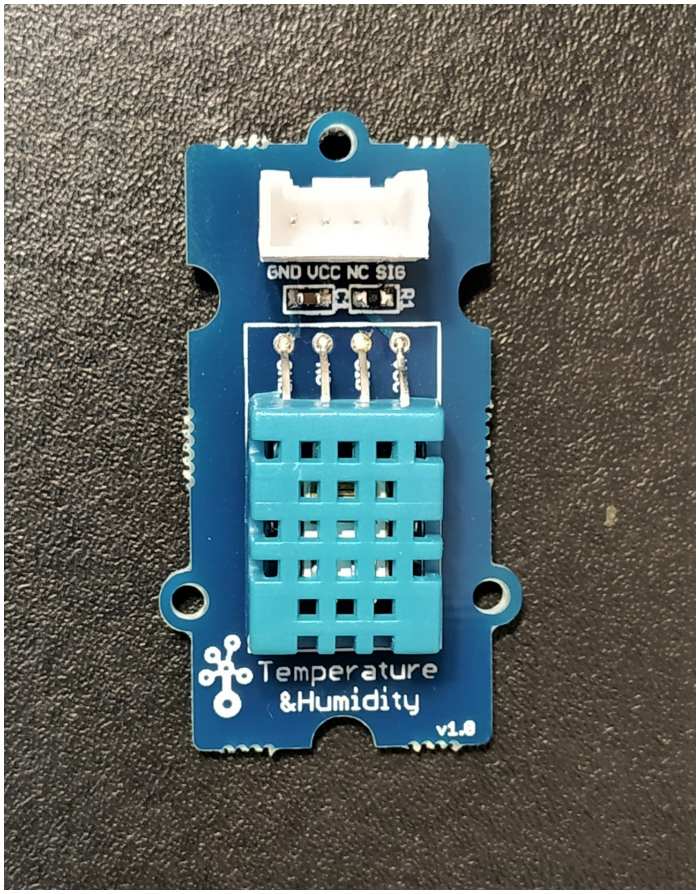

Figure 3.4: DHT11 Temperature and Humidity Sensor (Image Source: Photo taken by the author)

3.8 HC-SR04 Ultrasonic Sensor

HC-SR04 is an ultrasonic sensor that uses sound waves to measure distance. Because of its low cost and multiple usabilities, this sensor is used in many IoT projects integrating with Arduino and Raspberry Pi [80]. Most commonly it is used in direction and speed measurements, wireless charging, medical ultrasonography, sonar, burglar alarms, humidifiers. In non-contact detection, this sensor provides stable readings with high accuracy. HC-SR04 has a sound wave transmitter and receiver.

To find out the distance from an object this sensor first transmits the sound wave through its transmitter and receives the reflected sound waves back on its receiver. Calculating the time difference between wave sending and receiving, the sensor finds out the distance. Based on this principle if any object moves in front of it, this sensor can be used to count the number of moving objects. This sensor can measure the distance between 2cm to 400cm with 3mm accuracy. It takes around 5V DC and 15mA of current to be operated. The sensor has four pins and they are VCC, Trigger, Echo, and Ground pins. VCC and Ground are used for the power supply to the sensor. Trigger pin is used to send the command of generation of the ultrasonic sound wave and Echo pin receives returned waves data to send on the microcontroller or a computer used with it. Its measuring angle is 15 degrees with a working frequency of 40KHz that means the module will generate 40,000 Hz of sound waves each second from the source which is not audible by the human hearing system. We know that the speed of sound in dry air is 34300 cm/s. To calculate the distance, sound wave sending and receiving time is recorded to find out the time difference between those two events. Later on, in that particular time how much distance sound wave can travel that is calculated. Here is the formula to find out the distance using this sensor; Distance= (TimeElapsed*34300)/2.

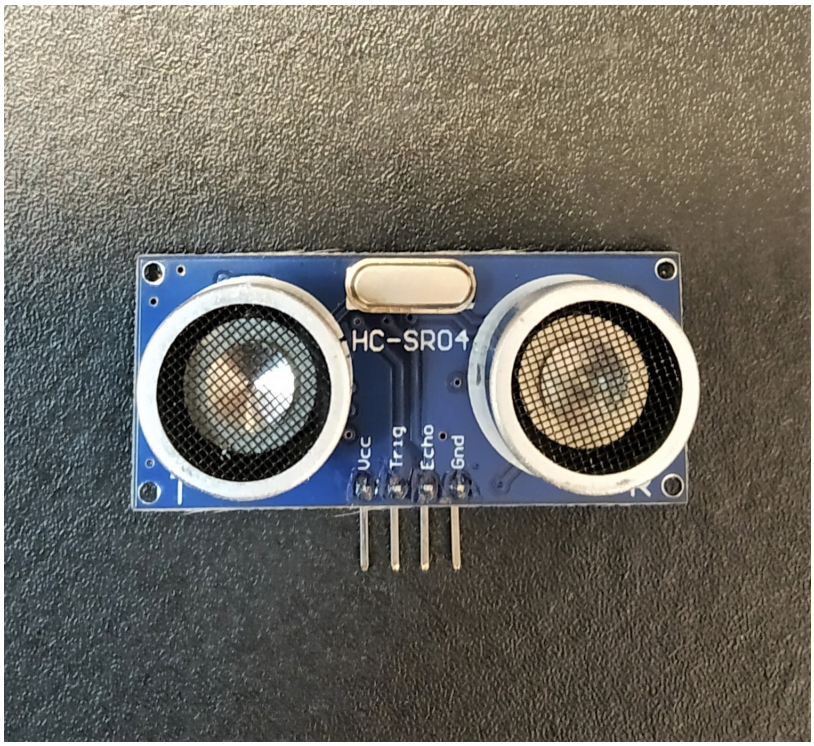

Figure 3.5: HC-SR04 Ultrasonic Sensor (Image Source: Photo taken by the author)

Chapter 4

Prototype Design Integrating Sensors and IOTA Tangle

4.1 Problem Overview

IoT devices are growing rapidly and because of that, there are going to be more communications or transactions among them within a short period. In real-time we need to send different types of IoT devices' data such as temperature and humidity sensors, object counter ultrasonic sensors, location detecting sensors, gas sensors, wearable devices' data to the servers to monitor them. Sometimes these sensors' data are very sensitive and need to be protected very carefully [81]. For example, temperature and humidity sensors can be used in food preserving rooms, smart homes, smart cities, chemical factories, nuclear power plants, hospitals, data centers, and even in food and vaccine carrying trucks. In all of these places, any type of temperament with temperature data can create a disastrous occurrence. In different places where these

temperature data are monitored using regular online dashboards, those systems are remaining in the most vulnerable condition. For example, if a smart home's temperature sensor system gets hacked, the sleeping residents can be harmed by increasing the air conditioner's temperature. The same types of unimaginable disastrous incidents can happen in a temperature-sensitive chemical factory or in a nuclear plant if an online-based temperature control system goes out of hand. In the same way, using an ultrasonic sensor humans counting system is used in shops or offices, an object counting system is used in warehouses, an anti-collision remote robotic system is made and all of these systems use an online-based system to record or live monitor the data. There are a plethora of use cases of these internet-dependent devices, so it is crucial to ensure the highest level of security for these IoT devices' data.

In a regular scenario, all these systems would be created using a database to store the sensors' data which would be stored on the device itself or in a cloud server. Later that data could be accessed from either of the databases through any web portal or dashboard. As cloud servers are centralized systems and store huge data, they are always targeted by hackers. A DDoS attack is a common event for both of these local or cloud-based databases. Blockchain or Distributed Ledger Technologies are widely accepted as almost hackproof technology because of complicated proof of work, and their open ledger system where every event is tracked using timestamping. This distributed ledger technology can play an important role to reduce these security issues for IoT devices. But still, it is a big challenge for these low memory and computational architecture-dependent devices to use blockchain or distributed ledger, as sometimes it requires huge computational power and time to solve the proof of work

puzzle.

IOTA Tangle is a specially designed distributed ledger keeping in mind the low architecture of IoT devices so that validation of these huge transactions takes lesser time. As IOTA is a Distributed Ledger Technology, an intruder can not target any specific cloud-based database server port here. The whole data is being published on a cryptography-dependent distributed ledger and everyone on the network always has a copy of this ledger. Even if an intruder tries to alter any data on the Tangle network, other nodes will be able to see the trail of change, hence the change won't be accepted by the network. Not only that but also through our private key, the published data will be encrypted and can only be decrypted by the access of this key on the receiving end. As IOTA Tangle is a newly introduced distributed ledger technology, there haven't been many experiments yet to publish the different types of sensor's real-time data to the decentralized Tangle network. Verification of the successful real-time integration of IoT sensors and IOTA Tangle to scale up the security of the IoT devices is a very important question for the researchers to widely implement the technology. At the same time, it is also necessary to find out the limitation and communication efficiency of this network.

4.2 Proposed Design

From our previous chapters, we know that IoT devices face different security issues because of different challenges, and to address the issue scholars are trying to integrate different blockchain or distributed ledger technologies with IoT devices as a proof of concept. To be part of finding out the real-time feasibility of IoT and

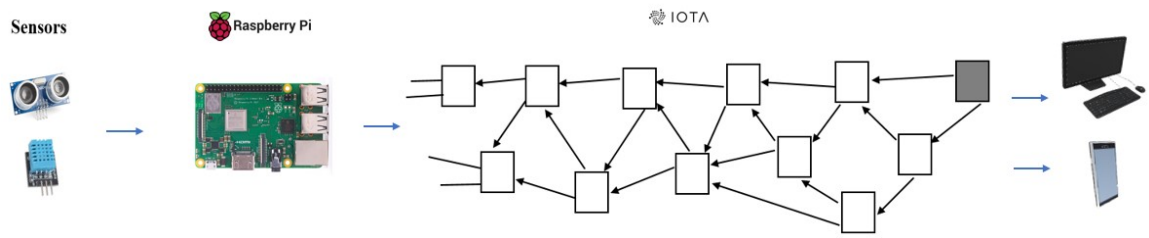

Figure 4.1: Sending sensors data through IOTA Tangle

DLTs efficient integration we have chosen IOTA Tangle as our desired DLT to be integrated with because of its previously discussed IoT-centric design. To integrate IoT and IOTA Tangle we have done two experiments to create two real-time working prototypes as a use case. Our testbed was the combination of the following devices,

i. DHT-11 Temperature and Humidity sensor

ii. HC SR04 Ultrasonic Sensor

iii. Raspberry Pi 2 Model B

In each of the experiments, the sensor was connected to a gateway which is a Raspberry Pi computer in our case. Using the internet this Raspberry Pi had sent these sensors' data to the IOTA Tangle and using the Raspberry Pi device or any Android mobile we received the data on the other side. For each of the sensors, we wrote our program to run it on the Raspberry Pie to read the sensors' data and publish it on the Tangle. Similarly, the data receiving program was written to run on Raspberry Pi to receive from the Tangle. Also, a simple Android program was

written to receive sensors data on Android mobile. We wrote the program to encrypt our sensors' data so that it can be read decrypting only using a password. All of these data sending and receiving procedures were done using IOTA's MAM protocol.

To build the prototype and complete the evaluation, we followed a few steps. Now we will discuss those steps.

4.3 Common Steps for Both of the Prototypes

To build each of the prototypes we have to follow some common steps for once. Here are the steps,

i) Installing Raspbian OS on the Raspberry Pi Device- Before starting the experiment with the sensors, the latest Raspbian operating system on a 16GB memory card of the Raspberry Pi was installed. To interact visually with the Raspberry Pi computer we connected it with a touchscreen 7-inch monitor which is SmartiPi Touch.

ii) Installing Node.js on the Raspberry Pi- We have already discussed that for sending and receiving data IOTA Tangle uses its MAM protocol (Masked Authenticated Messaging). To use the MAM communication protocol we have to use JavaScript programming language as MAM has its JavaScript library to send and receive the data. As the data sending and receiving programs are written in JavaScript, to run these programs on the Raspberry Pi device Node.js was installed on the device as a JavaScript runtime environment. Node.js was installed using the following commands on a Raspberry Pi command terminal.

Type: curl –sl https://deb.nodesource.com/setup$_8$.x|$sudo - Ebash$–

Type: sudo apt-get install –y nodejs

For checking the Node.js version: node -v

After installing it the Raspberry Pi device is needed to be restarted to work with Node.js properly.

iii) Installing Git on the Raspberry Pi- Git is an open-source code management tool. All the developed programs for both of the prototypes are kept on GitHub and the program will be installed on the Raspberry Pi device from there. To control the changes of the program from the Raspberry Pi, version control tool Git was installed using the following command on the terminal,

Type: sudo apt-get install git

iv) Installing BCM2835 library which provides access to GPIO for the sensor- For interfacing the sensors with the Raspberry Pi, access is needed on Raspberry's GPIO pins. To control the input and output (IO) function on Raspberry Pi, a C-based BCM2835 library needs to be installed. To download and install it the following commands are used on a terminal of the device,

For Download Type: wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.56.tar.gz

Build and install bcm2835-1.56:

Type: tar zxvf bcm2835-1.56.tar.gz

Type: cd bcm2835-1.56

Type: ./configure

Type: make

Type: sudo make check

Type: sudo make install

4.4 Integration of Raspberry Pi, DHT 11, and IOTA

To send the temperature and humidity data to IOTA Tangle we followed two steps. Firstly, we integrated the DHT11 temperature and humidity sensor with Raspberry Pi to collect and show the real-time room temperature and humidity. Secondly, we made a specific program for DHT11 to send the live temperature and humidity data to IOTA Tangle. These steps are given below,

i) Connecting DH11 with Raspberry Pi- To connect the DH11 sensor with Raspberry Pi 3 jumper wires were used. Sensor's Vcc and GND were connected with Raspberry's power and ground pins respectively and to get the data from the sensor, GPIO pin 4 was connected with the sensor's data pin. In the code of the written program, this data read pin 4 was specified.

ii) Download and install the project file from GitHub- After connecting the sensor with Raspberry Pi and following the common steps described above, the program that was built to read the sensor's data and to attach with IOTA Tangle had been downloaded on the device using the following commands on the terminal,

Type: cd

Figure 4.2: Prototype for Measuring Temperature and Humidity

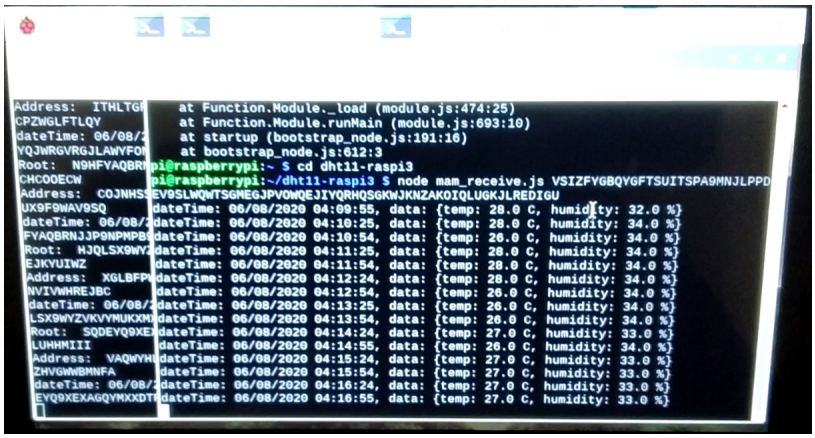

Figure 4.3: Sending and Receiving Temperature and Humidity with Timestamp on Raspberry Pi

Type: git clone https://github.com/mamunma1/DHT11-IOTA.git

Type: cd DHT11-IOTA

Type: npm install

4.4.1 DH11 and IOTA Prototype Program Files

This project has several JavaScript software files. If we want to run any of these we just have to execute a command opening a terminal on Raspberry Pi. Every time we run a command we just have to type node and then the software file name with extension such as "node sensor.js" and press enter.

• mam_publish: This program was written to send the randomly generated number to the IOTA tangle using MAM so that it can be tested.

• sensor.js: By this command sensor's temperature and humidity data are read and shown on the screen. As this sensor is a basic one there was no need to set any complicated algorithm for this sensor. The program was written to listen to the GPIO pin 4 continuously. By default, the data read time was set for every 5 minutes but it is modifiable easily by changing the interval value on the code.

• mam_sensor.js: By executing this command sensors data is read and published to the IOTA Tangle.

Sensor's Data is sent and received in the following format

dateTime: 06/08/2020 04:09:55, data: {temp: 28.0 C, humidity: 32.0 %}

4.5 Integration of Raspberry Pi, HC-SR04, and IOTA

To send the HC-SR04 ultrasonic sensor data to IOTA Tangle we followed two steps like before. Firstly, we integrated the HC-SR04 sensor with Raspberry Pi and made a program to count the human passed in front of it. Secondly, made a full

program to take the previous counting data and send it to the IOTA Tangle. These steps are given below,

i) Connecting HC-SR04 with Raspberry Pi- To connect the HC-SR04 sensor with Raspberry Pi 4 jumper wires were used. Sensor's Vcc and GND were connected with Raspberry's power and ground pins respectively and to trigger the ultrasonic sound wave GPIO pin 18 was connected with the sensor's trigger pin and to receive the returned sound wave sensor's echo pin was connected with Raspberry's GPIO pin 24.

ii) Installing Python- Sensor's counting program was written in the Python programming language. That's why we needed to install Python 3 on the Raspberry Pi device and it was done using the following command on the terminal,

Type: sudo apt-get install python3.8

iii) Download and install the project file from GitHub- After connecting the sensor with Raspberry Pi and following the common steps described above, the program that was built to read the sensor's data and to attach with IOTA Tangle had been downloaded on the device using the following commands on the terminal,

Type: cd

Type: git clone https://github.com/mamunma1/ HC-SR04-IOTA.git

Type: cd HC-SR04-IOTA

Type: npm install

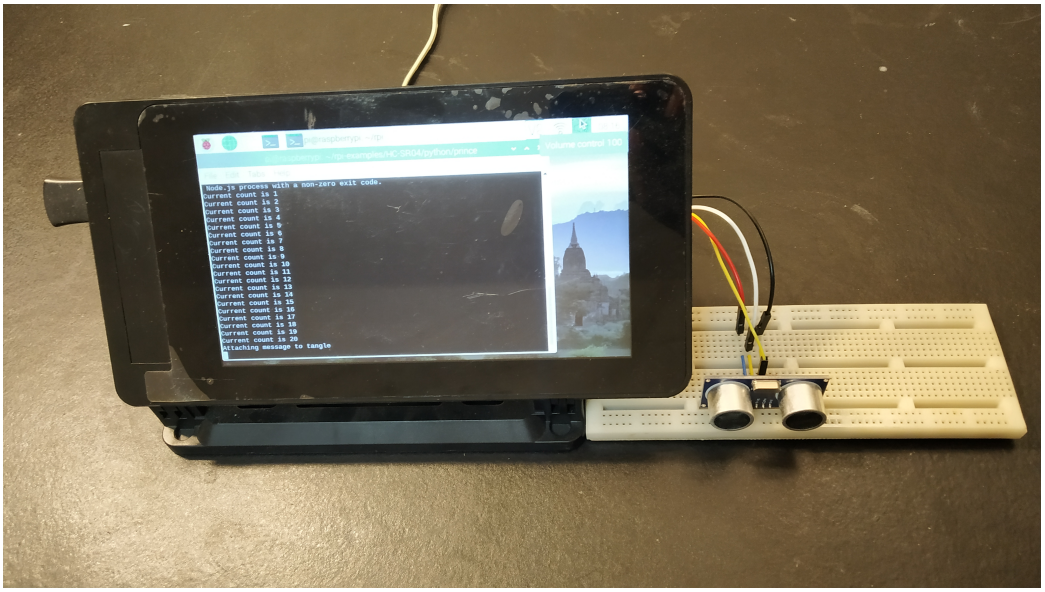

Figure 4.4: HC-SR04 Counter Prototype

iv) Running the Program- This project has several files. But to run the full program only one program file "distance.py" is needed to be run on the Raspberry Pi terminal. The command to run this file is "python3 distance.py". For this program, the counting program is made using python, and later on, the counting data has been transferred as a payload to a JavaScript program file "gateway.js" which is attaching the data to the IOTA Tangle as IOTA's communication protocol MAM only supports JavaScript library. Sensor's Data is sent and received in the following format,

Count: 20, Time: 21-02-15, 23: 27: 47, Message: "Time to Clean"

4.5.1 HC-SR04 Human Counting Working Method

To make a human counter using the HC-SR04 we imagined a scenario where the sensor will be in front of a washroom entrance or door. It has been considered that the prototype with the sensor will be set up on one side of the wall of an entrance or door. The width of the entrance or from one wall to the other is imagined around 110cm. Considering an adult person's physical structure and the width of the entrance, if a person crosses through the entrance he or she will always be within the distance of 100cm from the sensor. When the sensor is turned on it always keeps measuring the distance. If the prototype is placed on one side of a wall it will always find the distance as 110cm until any obstruction comes through the door and stands still lowering the distance. So in this prototype's case when anyone is coming through this entrance the sensor will find the distance lower than 100cm or at least equal to 100cm. So if this distance condition is filled out then the prototype will count the number of persons or obstruction as one. If the distance goes more than 100cm, it will not count. The counter will increase the count by another person when again the same condition will be met and just after that the distance will be more than 100cm. By this condition, if a person stays continuously in front of the sensor it will not increase the count. In this way, it will count till any number that will be set by the operator. In our case, we are considering counting 10 persons and after that, the sensor will publish the data that 10 people have entered the washroom and now its time to clean. In this imaginary scenario when the entered person will get out of the washroom the sensor will count the person again. So every entered person will be counted twice. If we want the sensor to report or publish the data of 10 people then we will have to count till

the double of 10 which is 20. To make the algorithm we have used this count number 20 but any limit can be set. To check if the sensor is working well all the time or not we have programmed it to publish a message on IOTA Tangle every 30 minutes that it is still active. The imaginary washroom cleaner or the proper authority will get this count data and cleaning message through the IOTA Tangle.

Algorithm 1 HC-SR04 Counting and Data Publishing

unpublishedSince=currentTime

countFlag is **true**

messageCounter=0

While (true)

Distance= readProximity()

If *Distance is less than or equal to 100cm*

countFlag= **true**

messageCounter=1

elseif *Distance is more than 100cm*

countFlag= **false**

if *countFlag is false and Distance is less or equal to 100cm*

messageCounter= messageCounter+1

if *messageCounter is equal to 20*

upload to the Tangle ***"Time to clean the washroom! The number of person used the washroom is the value of messageCounter"***

reset the messageCounter to zero

unpublishedSince=currentTime

elseif unpublishedSince – currentTime $\geq 30min$

upload to the Tangle ***"The sensor is still active and current number of persons used the washroom is the value of messageCounter"***

unpublishedSince = currentTime

4.6 Data Receiving Program

To receive the published data for Raspberry Pi a JavaScript program named mam_receive.js is made that's location link is mentioned on the DHT11 program section. To check or receive the data on one command terminal of Raspberry Pi we just have to use a root address of a published data after the command "node mam_receive.js". The published data with timestamp will be shown. Not only that a simple android app has been made just to check the live data of both prototypes on a mobile. A root address and a password (known as side key in IOTA) are needed to check the published data on mobile as all programs' published data is encrypted. IOTA Foundation has an Explorer Server to check the published data that has been integrated into this app.

This app is available on this link https://github.com/mamunma1/SenIOTA.

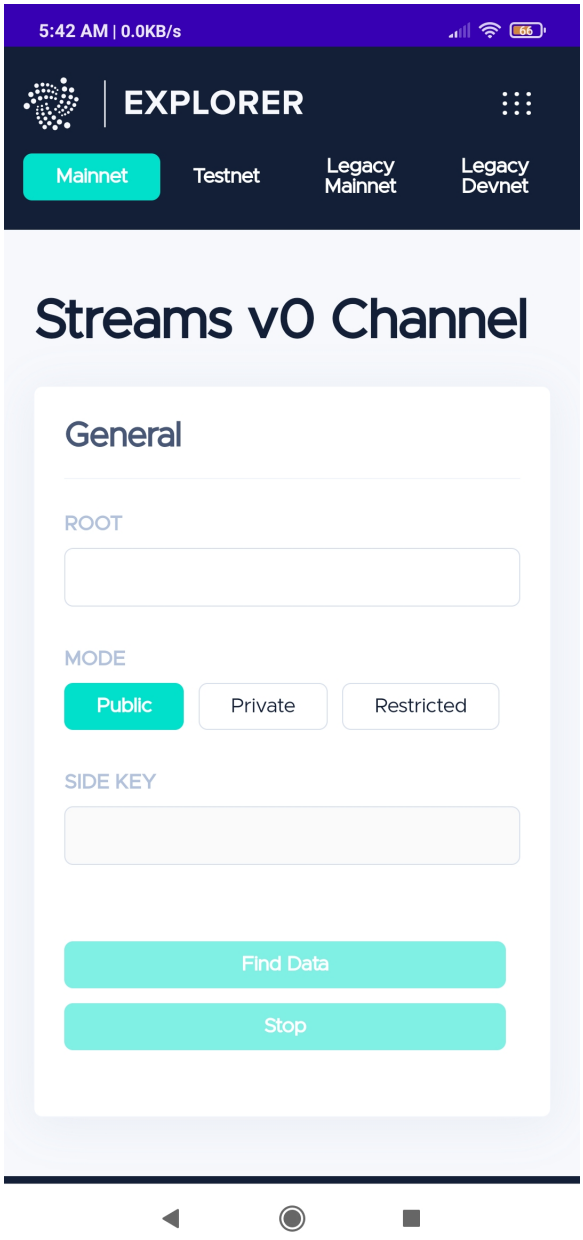

Figure 4.5: Sensor Data Receiving App Interface

Figure 4.6: Sample Data Receiving through App

Chapter 5

Evaluation and Discussion

5.1 Evaluation Method

We have already discussed what are the vulnerabilities of sending IoT data through regular cloud or data servers. We needed a secure way to send IoT data. Because of the distributed and immutable data recording ledger system of the DLTs, we wanted to integrate IoT devices with DLT. To make our integration project successful we chose a newly emerged DLT called IOTA Tangle because of its IoT-centric design. We proposed that we would try to test the feasibility of making a real-time prototype integrating IoT and IOTA Tangle. Not only that but also we proposed to check the data sharing efficiency through the IOTA Tangle upon successful prototype making.

From our previous prototype design segments, we have already seen how the two successful prototypes were built and how are they working. So it is fully clear that IoT and IOTA integration is quite possible. Now we have to check the data sharing efficiency of the IOTA Tangle network. To check the data sharing efficiency of the

IOTA network, both of the sensors' data were sent and received back to calculate those time data according to our evaluation metric. In the following description, we will discuss what was the data collection procedure for each of the prototypes.

5.1.1 Procedure for DHT11 Prototype

Temperature and humidity data monitoring is a common and very important thing in countless places. From smart home's air conditioners to sensitive drug delivery trucks, nuclear plants, medical incubators, food carrying vehicles, smart cities weather monitoring systems everywhere it is being monitored continuously. In most of these places, it is monitored live through any website's or mobile app's dashboard. If the data server that is delivering this live data gets hacked and the data can be tempered. Keeping these vulnerable scenarios in mind we have created this live temperature and humidity measuring prototype where the DHT11 sensor collected the data and through the Raspberry Pi computer this data was sent to the IOTA tangle. As no one can tamper with the published data this model can be applied anywhere where the live data has to be very secure. This prototype was set in my living room. Every 5 minutes, the sensed temperature and humidity data had been sent automatically in the IOTA network. Sometimes the sensor's data-taking interval time had been tweaked from 2 minutes to 15 minutes. The data was monitored and sent live on the IOTA network at different times of the day from morning till night to check if there are any changes in data attaching time on the IOTA network. From our data receiving program, we received or monitored this data live after publishing it to the IOTA network. As IOTA's data publishing and receiving MAM library has a timestamping

mechanism to record the time for every event from data sending to data read from the network, we have stored all that timestamping data to use it in our efficiency metric. Each day 40 data were recorded and the process had been done for 10 days. The Cron job program was used to automate the prototype so that whenever the power is on it can start sending temperature and humidity data without any human intervention.

5.1.2 Procedure for HC-SR04 Prototype

This prototype was built keeping in mind a washroom cleaning scenario where how many people used the washroom that data is needed to be monitored in real-time and based on that data the washroom cleaning person will clean the washroom each time. This prototype can be used anywhere where it is needed to count people or any passing object. For this prototype, we have imagined that it can be set up at the entrance of a washroom. The prototype will count the number of people who used the washroom. Suppose when it reaches the number of 10, the prototype will send this counting data to IOTA Tangle instead of any regular data server with a message saying that "Count is 10, Time to Clean" and the responsible cleaning person will be able to see this data on a mobile app with date and time. Every 30 minutes this prototype will also inform that it is still active through the IOTA network so that the cleaning person can know that it is working. This is a use case scenario for real-time use of this prototype. But for our evaluation, it is not needed to implement it in front of any washroom. Because we are monitoring the data attach and read time of this prototype to the IOTA Tangle network to calculate the efficiency. This prototype

Figure 5.1: HC-SR04 Data Counting Demonstration

can be set in front of any door and in our case, we have done so. This prototype was placed in my apartment in front of a door of my room. I have crossed through that door many times from morning till night. I set the sensor to count for 4 people. So every time I was passing in front of the sensor it was publishing that count data to the IOTA Tangle with a timestamp. As the MAM library has the functionality to record the data publishing and receiving time with a timestamp that data has been recorded for our efficiency calculation. Again for this prototype 40 timestamping data were recorded for each day and the process had been repeated for 10 days and the Cron job program was used here as well to automate the prototype so that whenever the power is on it can start sending counting data without any human intervention.

5.2 Evaluation Metrics

As DLT based technologies are still in their early stage, it is very tough to evaluate their performance from different aspects. There is no standard method yet to check their efficiency and scalability. Analyzing different research papers we have found that transactional efficiency is one of the major key components of a DLT. Because the data sending and receiving process through a DLT is not straightforward like the conventional cloud-based system. After publishing data on this type of distributed network, it is needed to be approved by other validators to broadcast the data over the whole network that is another extra work to consume time. Where access to data is needed at a lightning speed, there a delayed secure system might not be helpful. To accept and use IOTA Tangle's technology widely it is very important to determine how fast the sensors' data can be attached to the Tangle and receive it back from the IOTA Tangle fulfilling all the transactional conditions of the network. For different strings of data, it might take a different amount of time. Depending on the computational power of a node, the proof of work puzzle-solving time can vary, hence the data publishing time can also vary. If real-time transactional efficiency can be evaluated, based on it different users of this technology can decide where the implementation of this technology can be useful for them without compromising the potential security risk. Based on this, we have evaluated the performance of the prototype using the metrics of the table given below,

Table 5.1: Time Measurement Metrics

Performance parameter	Definition	Formula for calculation
Read latency	Time calculated between submission of a data read request and receipt of a reply	Read Latency = Time when the response received– Submit time
Read Throughput	The number of data read operations completed in a defined time, expressed as reads per second (RPS)	Read Throughput = Total read operations / Total time in seconds
Transaction Latency	Time taken for a transaction's effect to be usable across the network	Transaction Latency = Conformation time - Submit time
Transaction Throughput	The rate at which valid transactions are committed by the IOTA's distributed ledger in a defined time. This rate is expressed as transactions per second (TPS) at a network size.	Throughput = Total committed transactions / Total time in seconds

5.3 Collected Data

The data shown here in this section are two prototypes' data to send and receive the information from the IOTA tangle that has been recorded using IOTA's timestamping feature for 20 days over a month. Even though timestamping data is saved on a file with date and time, for our calculation exact date is not necessary that's why the date is not shown on the table. Only time is needed to find out the values

according to the mentioned metrics. After taking data for one sensor for 10 days, another sensor's data had been taken for another 10 days so that the IOTA network's data transactional speed consistency can be seen for a longer period which is around a month in this research. In this section, two sample timestamp data table is shown for the two prototypes and rest of the data tables are included in the Appendices section of this thesis. In the data tables, data is presented in each column according to our metrics. Column headings are named shortly where they are denoted as follows,

Response Receive Time (RRT)

Sensor Data Submit Time (SDST)

Data Attach Conformation Time (DACT)

Transaction Latency (TL)

Read Latency (RL)

Standard Deviation (St. Dev.) (±)

5.3.1 DH11 Sample Data

One sample timestamp data table is given here for the DHT11 prototype and the rest of the tables are in the Appendices section.

Table 5.2: DHT 11 Data Taken Every two Minutes: Day 1

RRT	SDST	DACT	TL	RL
10:40:49	10:40:32	10:40:46	14	17
10:42:44	10:42:33	10:42:42	09	11
10:44:48	10:44:33	10:44:44	11	15
10:46:39	10:46:31	10:46:37	06	08
10:48:45	10:48:32	10:48:43	11	13
10:50:40	10:50:32	10:50:37	05	08
10:52:44	10:52:34	10:52:42	08	10
10:54:43	10:54:31	10:54:40	09	12
10:56:39	10:56:32	10:56:37	05	07
10:58:41	10:58:32	10:58:38	06	09
11:00:41	11:00:31	11:00:39	08	10
11:02:46	11:02:32	11:02:42	10	14
11:04:51	11:04:33	11:04:48	15	18
11:06:49	11:06:32	11:06:45	13	17
11:08:51	11:08:32	11:08:48	16	19
11:10:43	11:10:31	11:10:41	10	12
11:12:47	11:12:32	11:12:44	12	15
11:14:46	11:14:32	11:14:42	10	14
11:16:44	11:16:31	11:16:41	10	13
11:18:45	11:18:31	11:18:42	11	14
21:13:38	21:13:22	21:13:33	11	16
21:15:37	21:15:21	21:15:34	13	16
21:17:35	21:17:22	21:17:31	09	13
21:19:39	21:19:22	21:19:36	14	17
21:21:35	21:21:23	21:21:31	08	12
21:23:30	21:23:21	21:23:28	07	09
21:25:31	21:25:21	21:25:29	08	10
21:27:34	21:27:22	21:27:31	09	12
21:29:35	21:29:22	21:29:32	10	13
21:31:39	21:31:22	21:31:34	12	17
21:33:38	21:33:23	21:33:34	11	15
21:35:41	21:35:22	21:35:37	15	19
21:37:33	21:37:22	21:37:29	07	11
21:39:35	21:39:22	21:39:31	09	13
21:41:32	21:41:22	21:41:30	08	10
21:43:29	21:43:21	21:43:27	06	08
21:45:32	21:45:21	21:45:30	09	11
21:47:35	21:47:21	21:47:31	10	14
21:49:34	21:49:22	21:49:31	09	12
21:51:34	21:51:21	21:51:32	11	13
		Average and St Dev.	9.88(±2.74)	12.93(±3.17)

5.3.2 HC-SR04 Sample Data

One sample timestamp data table is given here for the HC-SR04 prototype and the rest of the tables are in the Appendices section.

Table 5.3: Table 1: HC-SR04 Data for Day 1

RRT	SDST	DACT	TL	RL
07:40:43	07:40:23	07:40:38	15	20
08:05:35	08:05:13	08:05:32	19	22
08:30:34	08:30:10	08:30:30	20	24
09:10:51	09:10:33	09:10:48	15	18
09:30:32	09:30:17	09:30:29	12	15
10:03:23	10:03:08	10:03:21	13	15
10:15:57	10:15:41	10:15:52	11	16
10:37:26	10:37:12	10:37:23	11	14
10:55:48	10:55:31	10:55:43	12	17
11:10:55	11:10:37	11:10:50	13	18
11:26:03	11:25:44	11:26:00	16	19
11:45:24	11:45:12	11:45:22	10	12
12:05:04	12:04:50	12:05:01	11	14
12:23:30	12:23:15	12:23:25	10	15
12:50:23	12:50:11	12:50:21	10	12
13:11:35	13:11:20	13:11:33	13	15
13:28:01	13:27:45	13:27:58	13	16
13:40:27	13:40:14	13:40:24	10	13
13:50:20	13:50:10	13:50:17	07	10
13:57:52	13:57:34	13:57:49	15	18
14:30:55	14:30:44	14:30:52	08	11
14:45:56	14:45:42	14:45:54	12	14
14:55:50	14:55:38	14:55:48	10	12
15:13:20	15:13:10	15:13:18	08	10
15:29:36	15:29:26	15:29:34	08	10
15:47:51	15:47:39	15:47:48	09	12
16:18:49	16:18:36	16:18:46	10	13
16:35:04	16:34:50	16:35:01	11	14
16:49:39	16:49:27	16:49:37	10	12
17:10:53	17:10:38	17:10:50	12	15
17:30:40	17:30:24	17:30:37	13	16
17:45:34	17:45:17	17:45:30	13	17
17:55:43	17:55:24	17:55:40	16	19
18:09:27	18:09:13	18:09:24	11	14
18:20:34	18:20:19	18:20:31	12	15
18:40:21	18:40:05	18:40:17	12	16
20:17:53	20:17:33	20:17:50	17	20
20:33:47	20:33:32	20:33:44	12	15
20:45:48	20:45:34	20:45:45	11	14
20:55:53	20:55:40	20:55:50	10	13
		Average and St Dev.	12.03(±2.88)	15.13(±3.22)

5.4 Results from the Collected Data

On the data tables for each of the prototypes, 400 transactions time had been recorded for 10 days. For two prototypes total of 800 transactions' time data had been recorded for a total of 20 days within the time of around one month. According to our metrics, we have calculated the Read Latency and Transaction Latency for each of the transactions, and that data is shown on the table in seconds. From all these data we figured out two other metrics Read Throughput and Transaction Throughput.

Read throughput means in one second how much data can be read from the IOTA Tangle.

Read Throughput = Total read operations/ Total time for these read operations in the second = 800/10209 = 0.078 RPS (Read Per Second)

Transaction throughput means the rate at which valid transactions are committed by the IOTA network.

Transaction Throughput =Total committed transactions/ Total time for these all transactions in seconds =800/8057 = 0.099 TPS (Transactions Per Second)

Now we will see the average time taken to complete a successful transaction. According to our metric on the table, it is shown as transaction latency.

Average or Mean Transaction Latency = Total Transaction Latency time for both of the prototypes/ Total number of transactions =8057/800 =10.07 Second/Transaction

This time we will see the time it took to finally show or visualize one successful transaction data on our program or app.

Average or Mean Read Latency= Total Read Latency time for both of the proto-

types/ Total number of transactions = 10209/800 = 12.76 Seconds

From our experiments, we can finally say that from starting a transaction to show it on the user's end IOTA Tangle averagely takes 12.76 Seconds. From the Average Transaction Latency and Average Read Latency value, we see there is a difference between them. This is the average time delay to display transaction data on the receiving end. Average display time delay = (12.76-10.07) Seconds =2.69 Seconds

One most important finding of this experiment was among all these transactions not a single transaction had been failed on this network rather some transactions took a little more time to be successful.

5.5 Graphical Representation of the Results

We have calculated the standard deviation for both of the latencies and to show the real data distribution we have created histograms.

5.5.1 Read Latency

For average Read Latency the standard deviation value is 3.20. From the graph, for this value we can see that the Bell curve for standard deviation is not so wide which means the read latency values for most of the transactions are not spread so widely. Most of the values are very close to the average value of Read Latency which is 12.76 s/transaction.

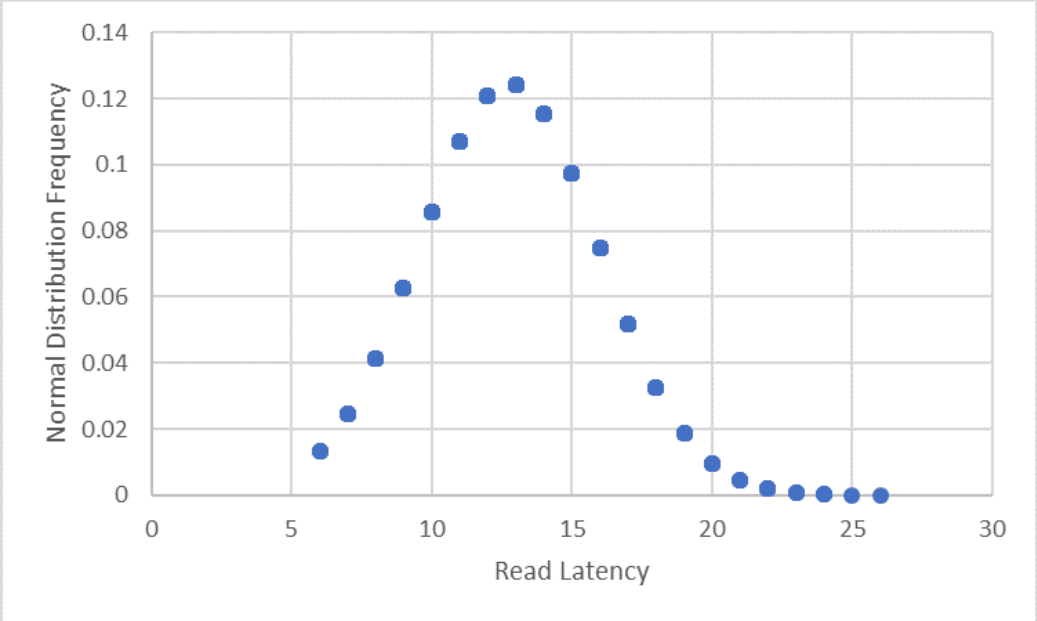

Figure 5.2: Read Latency Bell Curve

If we distribute the whole read latency data on a histogram we can see that the

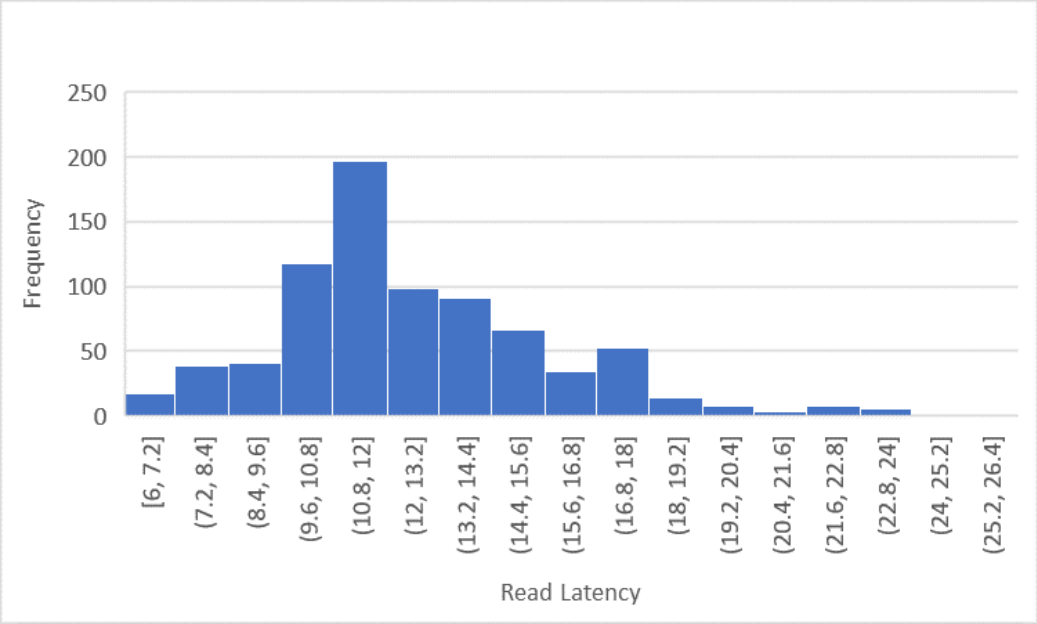

Figure 5.3: Read Latency Histogram

maximum frequency of Read Latency time is within the range of 10.8 and 12.

5.5.2 Transaction Latency

For average Transaction Latency, the standard deviation value is 2.82. For this value the Bell curve looks almost the same as Read Latency. From the graph, we can see that the Bell curve for standard deviation is not so wide in this case as well which means the Transaction Latency values for most of the transactions are not spread so widely. Most of the values are very close to the average value of Transaction Latency which is 10.07 s/transaction.

If we distribute the whole Transaction latency data on a histogram we can see that the maximum frequency of Transaction Latency time is within the range of 9.5 and 10.6.

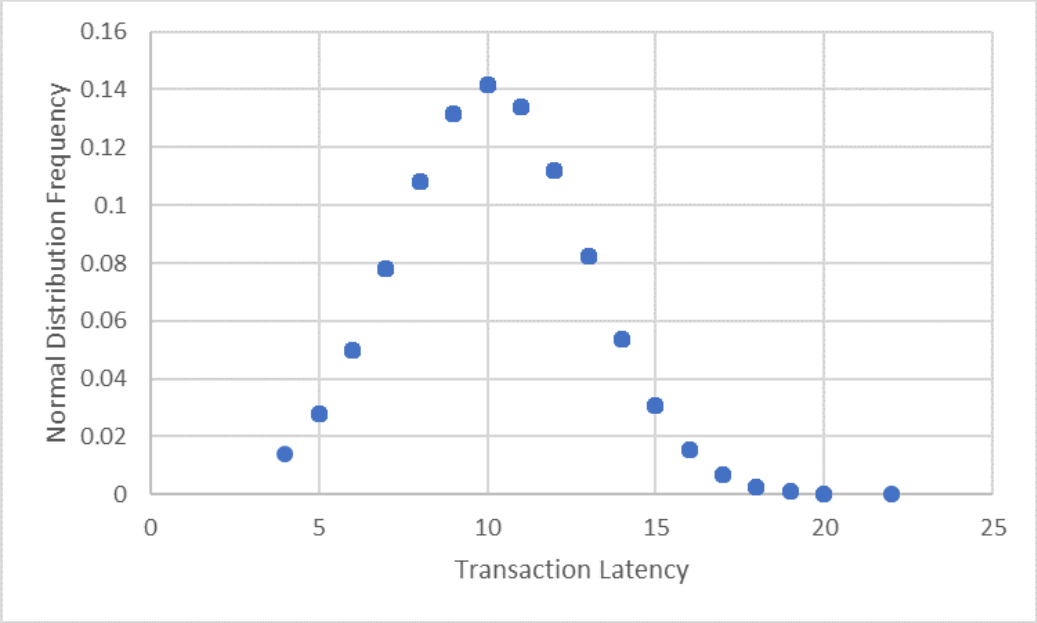

Figure 5.4: Transaction Latency Bell Curve

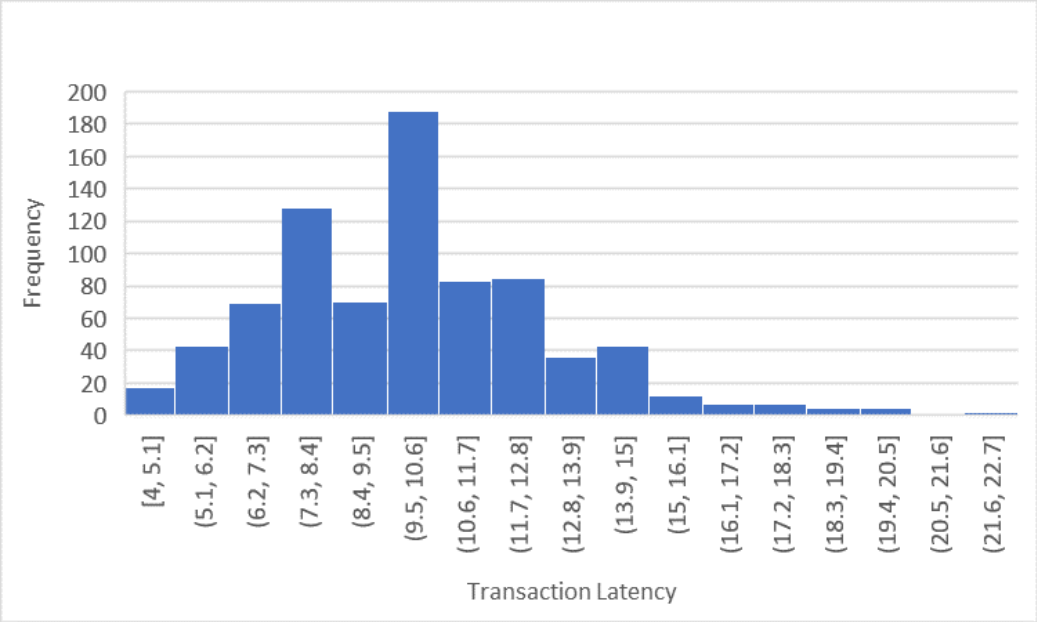

Figure 5.5: Transaction Latency Histogram

5.6 Limitations

IOTA Tangle is in a continuous process of developing of their network architecture. Still, this network is not a mature network like any new invention. IOTA doesn't have support for all types of smart devices to communicate with its network directly. We had tried to use IOTA Tangle to integrate with Samsung Galaxy Smartwatch 3. Our purpose was to create a prototype where health data would be shared with physicians through a secure IOTA Tangle network. But it was not possible because of hardware and software compatibility issues. IOTA's programmable library for IoT is mostly Raspberry Pi focused or the program can be written on similar types of architecture-based computers. When we started this research work MAM was stable and was in a full working model to send and receive communication data. From the launch of IOTA's communication protocol, IOTA Foundation mentioned that their communication protocol will go through different changes over time. Very recently IOTA's communication protocol MAM has fully been phased out. MAM was created based on a controversial ternary data structure where the data is represented by using these three numbers -1, 0, 1. MAM has its new version which is now known as IOTA Streams. It is still in Alpha or it is in the first stage. They call it Chrysalis or IOTA 1.5. This new communication protocol's data structure is changed to a binary system instead of using the previous ternary system. MAM's JavaScript communication library is not working anymore because of the change in the data structure and the previous transaction data history has also been archived which is known as Snapshot in IOTA. In this Chrysalis phase 1 of Stream, there is a limitation. Messages can be read once only. IOTA Stream is now written in the programming language Rust

instead of JavaScript. Upcoming change with IOTA Stream is Chrysalis Phase 2 which is expected to be a production-ready design for the industry instead of the current experimental version. Recently I have tried to use their new version of the communication protocol as well. But the communication library they have written in Rust have some issues to be used with Rasberry Pi or any ARM-based processor. Some packages are facing compatibility issues. At the time of writing this thesis, their GitHub Rust development repository was not complete. Even though we had been able to take a substantial amount of transaction data while MAM was working, still it would be possible to take some more transactional data if there would be no issues with their sudden change in data structure and up-gradation of their communication library in a different programming language.

Chapter 6

Conclusions and Future Work

6.1 Future work

Nodes are a significant part of the IOTA network because all the transactions that happen in the IOTA network are validated by the nodes following a consensus mechanism. In our data sending and receiving program a link of a public IOTA node had been hardcoded to get connected with the most updated ledger of the network. As Raspberry Pi was not set up as a local full node, it can not connect with the neighbour nodes to keep itself updated. The full node requires more memory to store the big number of transaction data. By running a program that is called Hornet, public full nodes remain connected with other global nodes to be consistent with the most recent update on the network. For an extension of this work, a local full node of our own can be set up to do the proof of work fast to validate the transaction quickly. Then we can calculate the transaction validation time efficiency of that local node as well. As mining in blockchain or DLT is a big debate these days for their

energy consumption which harms the environment, it can be figured out what is the exact energy consumption level for a full local validator node and whether an efficient full local node can be set up on a low architecture device like Raspberry Pi. In our use case prototypes, only one device is working to send or receive the data from the Tangle. Not only this but also in the future, there can be another approach to build a more complicated prototype where one prototype can try to negotiate with another prototype through the IOTA Tangle network and can be incentivized for their services through the IOTA cryptocurrency and how efficiently the network can do it that can be figured out.

6.2 Conclusions

Data integrity and security of IoT devices can be achieved in many ways. From our literature review, we had figured out Blockchain or DLT based solutions are in huge focus because of their immutable distributed data recording system to reduce the security threats for IoT devices. But it is a big question what sort of DLT can be integrated with IoT. As DLT is mainly built for cryptocurrency, only a few DLTs that have smart contract features can build applications on top of it. Not all of these DLTs are designed for IoT devices integration, moreover, they need transaction fees which is a big impediment for IoT devices for billions of their regular communication. Communication in IOTA Tangle is free and it is the first generation DLT of its type. Only a few research papers talk about some theoretical use cases of sensor devices and IOTA integration. IOTA needs huge use case experiments to find out its usability. Time is a big factor in the communication world. In our established cloud-based

communication world, everything happens almost in a blink of an eye. If DLTs integration with IoT takes unreasonable time for a specific communication service, the technology might not flourish in the long run even if it ensures better security. That's why the main goal of this research was to test the feasibility of integrating IOTA as a DLT with IoT and test its transactional efficiency. From this research, we have figured out the feasibility of integration, efficiency, and its current limitations. Distributed ledger technologies are ideal tools to facilitate IoT infrastructure where devices can communicate, work together, and also make and receive payments for their services. Even though the IOTA Tangle has not yet been massively deployed for smart devices, still in the immature development stage but it has the potential to create a new dimension to ensure the security of IoT devices. This growing field of research will have a direct impact on our daily lives; already industrial companies worldwide are joining the research effort to develop the viability of such Distributed Ledger Technologies. Our research can elevate the use of this new technology for IoT devices. That's why through this research we had tried to build a real-time example of a prototype of the IOTA Tangle and IoT to scale up the security of the IoT devices, at the same time find out the potential of IOTA Tangle.

Bibliography

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] D. Giusto, A. Iera, G. Morabito, and L. Atzori, *The internet of things: 20th Tyrrhenian workshop on digital communications*. Springer Science & Business Media, 2010.

[3] X. Xu, C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A. B. Tran, and S. Chen, "The blockchain as a software connector," in *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, IEEE, 2016, pp. 182–191.

[4] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba, "Blockchain technology innovations," in *2017 IEEE technology & engineering management conference (TEMSCON)*, IEEE, 2017, pp. 137–141.

[5] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A taxonomy of blockchain-based systems for architecture design," in *2017 IEEE international conference on software architecture (ICSA)*, IEEE, 2017, pp. 243–252.

[6] G. W. Peters and E. Panayi, "Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money," in *Banking beyond banks and money*, Springer, 2016, pp. 239–278.

[7] M. Mainelli and M. Smith, "Sharing ledgers for sharing economies: An exploration of mutual distributed ledgers (aka blockchain technology)," *Journal of financial perspectives*, vol. 3, no. 3, 2015.

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

[9] C. Cachin *et al.*, "Architecture of the hyperledger blockchain fabric," in *Workshop on distributed cryptocurrencies and consensus ledgers*, Chicago, IL, vol. 310, 2016.

[10] 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 thirteenth EuroSys conference*, 2018, pp. 1–15.

[11] M. Raskin and D. Yermack, "Digital currencies, decentralized ledgers and the future of central banking," in *Research handbook on central banking*, Edward Elgar Publishing, 2018.

[12] I. Weber, V. Gramoli, A. Ponomarev, M. Staples, R. Holz, A. B. Tran, and P. Rimba, "On availability for blockchain-based systems," in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, IEEE, 2017, pp. 64–73.

[13] S. Thompson, P. Lamela Seijas, and D. Adams, "Scripting smart contracts for distributed ledger technology," 2016.

[14] W. Egbertsen, G. Hardeman, M. van den Hoven, G. van der Kolk, and A. van Rijsewijk, "Replacing paper contracts with ethereum smart contracts," *Semantic Scholar*, vol. 35, 2016.

[15] M. Alharby and A. Van Moorsel, "Blockchain-based smart contracts: A systematic mapping study," *arXiv preprint arXiv:1710.06372*, 2017.

[16] N. M. Kumar and P. K. Mallick, "Blockchain technology for security issues and challenges in iot," *Procedia Computer Science*, vol. 132, pp. 1815–1823, 2018.

[17] H. Damghani, L. Damghani, H. Hosseinian, and R. Sharifi, "Classification of attacks on iot," in *4th International Conference on Combinatorics, Cryptography, Computer Science and Computation*, 2019.

[18] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[19] D. Diaz-Sanchez, A. Marín-Lopez, F. A. Mendoza, P. A. Cabarcos, and R. S. Sherratt, "Tls/pki challenges and certificate pinning techniques for iot and m2m secure communications," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3502–3531, 2019.

[20] A. Bahga and V. K. Madisetti, "Blockchain platform for industrial internet of things," *Journal of Software Engineering and Applications*, vol. 9, no. 10, pp. 533–546, 2016.

[21] M. Conoscenti, A. Vetro, and J. C. De Martin, "Blockchain for the internet of things: A systematic literature review," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, IEEE, 2016, pp. 1–6.

[22] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," *arXiv preprint arXiv:1506.03471*, 2015.

[23] Y. Zhang and J. Wen, "An iot electric business model based on the protocol of bitcoin," in *2015 18th international conference on intelligence in next generation networks*, IEEE, 2015, pp. 184–191.

[24] D. Wörner and T. von Bomhard, "When your sensor earns money: Exchanging data for cash with bitcoin," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 2014, pp. 295–298.

[25] L. Axon, "Privacy-awareness in blockchain-based pki," *Cdt technical paper series*, vol. 21, p. 15, 2015.

[26] C. Fromknecht, D. Velicanu, and S. Yakoubov, "Certcoin: A namecoin based decentralized authentication system," *Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep*, vol. 6, pp. 46–56, 2014.

[27] I. Friese, J. Heuer, and N. Kong, "Challenges from the identities of things: Introduction of the identities of things discussion group within kantara initiative," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, IEEE, 2014, pp. 1–4.

[28] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, vol. 4, pp. 2292–2303, 2016.

[29] P. Brody and V. Pureswaran, "Device democracy: Saving the future of the internet of things," *IBM, September*, vol. 1, no. 1, p. 15, 2014.

[30] C. H. Lee and K.-H. Kim, "Implementation of iot system using block chain with authentication and data protection," in *2018 International Conference on Information Networking (ICOIN)*, IEEE, 2018, pp. 936–940.

[31] *Slock.it—blockchain + iot*, `Available:https://slock.it/faq.md`, 2020.

[32] *Ethereum frontier*, `Available:https://www.ethereum.org/`, 2020.

[33] X. Liang, J. Zhao, S. Shetty, and D. Li, "Towards data assurance and resilience in iot using blockchain," in *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*, IEEE, 2017, pp. 261–266.

[34] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*, IEEE, 2017, pp. 618–623.

[35] S. King, "Primecoin: Cryptocurrency with prime number proof-of-work," *July 7th*, vol. 1, no. 6, 2013.

[36] M. A. Khan and K. Salah, "Iot security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[37] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 254–269.

[38] M. Dhawan, "Analyzing safety of smart contracts," in *Proceedings of the Conference: Network and Distributed System Security Symposium, San Diego, CA, USA*, 2017, pp. 16–17.

[39] L. Brent, A. Jurisevic, M. Kong, E. Liu, F. Gauthier, V. Gramoli, R. Holz, and B. Scholz, "Vandal: A scalable security analysis framework for smart contracts," *arXiv preprint arXiv:1809.03981*, 2018.

[40] P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Buenzli, and M. Vechev, "Securify: Practical security analysis of smart contracts," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 67–82.

[41] S. Amani, M. Bégel, M. Bortin, and M. Staples, "Towards verifying ethereum smart contract bytecode in isabelle/hol," in *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*, 2018, pp. 66–77.

[42] E. Albert, P. Gordillo, B. Livshits, A. Rubio, and I. Sergey, "Ethir: A framework for high-level analysis of ethereum bytecode," in *International symposium on automated technology for verification and analysis*, Springer, 2018, pp. 513–520.

[43] W. F. Silvano and R. Marcelino, "Iota tangle: A cryptocurrency to communicate internet-of-things data," *Future Generation Computer Systems*, vol. 112, pp. 307–319, 2020.

[44] B. Shabandri and P. Maheshwari, "Enhancing iot security and privacy using distributed ledgers with iota and the tangle," in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, 2019, pp. 1069–1075.

[45] E. Vieira, P. C. Bartolomeu, S. M. Hosseini, and J. Ferreira, "Iotapass: Enabling public transport payments with iota," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, IEEE, 2020, pp. 1–6.

[46] R. Nakanishi, Y. Zhang, M. Sasabe, and S. Kasahara, "Iota-based access control framework for the internet of things," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, IEEE, 2020, pp. 87–95.

[47] S. K. Pinjala and K. M. Sivalingam, "Dcaci: A decentralized lightweight capability based access control framework using iota for internet of things," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, IEEE, 2019, pp. 13–18.

[48] M. Bhandary, M. Parmar, and D. Ambawade, "Securing logs of a system-an iota tangle use case," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, IEEE, 2020, pp. 697–702.

[49] K. Ashton *et al.*, "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[50] X. Huang, P. Craig, H. Lin, and Z. Yan, "Seciot: A security framework for the internet of things," *Security and communication networks*, vol. 9, no. 16, pp. 3083–3094, 2016.

[51] 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), Tech. Rep*, vol. 1, no. 9, 2014.

[52] M. Ben-Daya, E. Hassini, and Z. Bahroun, "Internet of things and supply chain management: A literature review," *International Journal of Production Research*, vol. 57, no. 15-16, pp. 4719–4742, 2019.

[53] K. E. Skouby and P. Lynggaard, "Smart home and smart city solutions enabled by 5g, iot, aai and cot services," in *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, IEEE, 2014, pp. 874–878.

[54] K. Su, J. Li, and H. Fu, "Smart city and the applications," in *2011 international conference on electronics, communications and control (ICECC)*, IEEE, 2011, pp. 1028–1031.

[55] Q. Gou, L. Yan, Y. Liu, and Y. Li, "Construction and strategies in iot security system," in *2013 IEEE international conference on green computing and communications and IEEE internet of things and IEEE cyber, physical and social computing*, IEEE, 2013, pp. 1129–1132.

[56] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, "Iot security: Ongoing challenges and research opportunities," in *2014 IEEE 7th international conference on service-oriented computing and applications*, IEEE, 2014, pp. 230–234.

[57] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[58] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: A survey," *Future generation computer systems*, vol. 56, pp. 684–700, 2016.

[59] D. Rhodes, "The future is saas, the future is in a cloud," *Int'l. In-House Counsel J.*, vol. 3, p. 1, 2009.

[60] D. Beimborn, T. Miletzki, and S. Wenzel, "Platform as a service (paas)," *Business & Information Systems Engineering*, vol. 3, no. 6, pp. 381–384, 2011.

[61] D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman, and D. Woods, "Cloud-trust—a security assessment model for infrastructure as a service (iaas) clouds," *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 523–536, 2015.

[62] M. Rauchs, A. Glidden, B. Gordon, G. C. Pieters, M. Recanatini, F. Rostand, K. Vagneur, and B. Z. Zhang, "Distributed ledger technology systems: A conceptual framework," *Available at SSRN 3230013*, 2018.

[63] A. Sunyaev, "Distributed ledger technology," in *Internet Computing*, Springer, 2020, pp. 265–299.

[64] C. V. Helliar, L. Crawford, L. Rocca, C. Teodori, and M. Veneziani, "Permissionless and permissioned blockchain diffusion," *International Journal of Information Management*, vol. 54, p. 102 136, 2020.

[65] M. C. Ballandies, M. M. Dapp, and E. Pournaras, "Decrypting distributed ledger design—taxonomy, classification and blockchain community evaluation," *Cluster Computing*, pp. 1–22, 2021.

[66] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business & Information Systems Engineering*, vol. 59, no. 3, pp. 183–187, 2017.

[67] A. G. DAG, "Directed acyclic graph," 2013.

[68] L. Baird, M. Harmon, and P. Madsen, "Hedera: A public hashgraph network & governing council," *White Paper*, vol. 1, 2019.

[69] S. Zaman, M. R. Khandaker, R. T. Khan, F. Tariq, and K.-K. Wong, "Thinking out of the blocks: Holochain for distributed security in iot healthcare," *arXiv preprint arXiv:2103.01322*, 2021.

[70] A. Deshpande, K. Stewart, L. Lepetit, and S. Gunashekar, "Distributed ledger technologies/blockchain: Challenges, opportunities and the prospects for standards," *Overview report The British Standards Institution (BSI)*, vol. 40, p. 40, 2017.

[71] B. Farahani, F. Firouzi, and M. Luecking, "The convergence of iot and distributed ledger technologies (dlt): Opportunities, challenges, and solutions," *Journal of Network and Computer Applications*, vol. 177, p. 102 936, 2021.

[72] M. Westerlund and N. Kratzke, "Towards distributed clouds: A review about the evolution of centralized cloud computing, distributed ledger technologies, and a foresight on unifying opportunities and security implications," in *2018 International Conference on High Performance Computing & Simulation (HPCS)*, IEEE, 2018, pp. 655–663.

[73] M. Bhandary, M. Parmar, and D. Ambawade, "A blockchain solution based on directed acyclic graph for iot data security using iota tangle," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, 2020, pp. 827–832.

[74] C. Dods, N. P. Smart, and M. Stam, "Hash based digital signature schemes," in *IMA International Conference on Cryptography and Coding*, Springer, 2005, pp. 96–115.

[75] J. Fleck Jr and E. Canfield, "A random walk procedure for improving the computational efficiency of the implicit monte carlo method for nonlinear radiation transport," *Journal of Computational Physics*, vol. 54, no. 3, pp. 508–523, 1984.

[76] D. Stucchi, R. Susella, P. Fragneto, and B. Rossi, "Secure and effective implementation of an iota light node using stm32," in *Proceedings of the 2nd Workshop on Blockchain-enabled Networked Sensor*, 2019, pp. 28–29.

[77] L. Vigneri and W. Welz, "On the fairness of distributed ledger technologies for the internet of things," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, IEEE, 2020, pp. 1–3.

[78] N. Petrov, D. Dobrilovic, M. Kavalić, and S. Stanisavljev, "Examples of raspberry pi usage in internet of things," in *Proceedings of 6th International Conference on Applied Internet and Information Technologies AIIT2016*, 2016, pp. 03–04.

[79] W. Gay, "Dht11 sensor," in *Advanced Raspberry Pi*, Springer, 2018, pp. 399–418.

[80] E. J. Morgan, *Hc-sr04 ultrasonic sensor*, 2014.

[81] A. Chaudhuri, "Internet of things data protection and privacy in the era of the general data protection regulation," *Journal of Data Protection & Privacy*, vol. 1, no. 1, pp. 64–75, 2016.

Appendices

DHT11 Prototype Timestamp Data is given below.

Table 2: DHT 11 Data Taken Every 2 Minutes Day 2

RRT	SDST	DACT	TL	RL
08:47:37	08:47:22	08:47:33	11	15
08:49:35	08:49:23	08:49:33	10	12
08:51:36	08:51:23	08:51:33	10	13
08:53:36	08:53:22	08:53:34	12	14
08:55:40	08:55:22	08:55:36	14	18
08:57:44	08:57:24	08:57:39	15	20
08:59:42	08:59:20	08:59:38	18	22
09:01:39	09:01:22	09:01:36	14	17
09:03:36	09:03:23	09:03:33	10	13
09:05:44	09:05:24	09:05:32	08	10
09:07:33	09:07:21	09:07:30	09	12
09:09:33	09:09:22	09:09:29	07	11
09:11:37	09:11:23	09:11:34	11	14
09:13:37	09:13:22	09:13:34	12	15
09:15:39	09:15:23	09:15:35	12	16
09:17:36	09:17:24	09:17:33	09	12
09:19:31	09:19:23	09:19:29	06	08
09:21:33	09:21:23	09:21:31	08	10
09:23:34	09:23:22	09:23:31	09	12
09:25:35	09:25:22	09:25:31	09	13
22:10:57	22:10:47	22:10:54	07	10
22:12:57	22:12:48	22:12:54	06	09
22:14:59	22:14:46	22:14:56	10	13
22:17:05	22:16:48	22:17:01	13	17
22:19:03	22:18:48	22:18:59	11	15
22:21:02	22:20:48	22:20:58	10	14
22:23:02	22:22:47	22:22:59	12	15
22:25:03	22:24:47	22:25:00	13	16
22:27:04	22:26:45	22:26:59	14	19
22:29:02	22:28:47	22:28:59	12	15
22:31:00	22:30:47	22:30:57	10	13
22:33:03	22:32:46	22:32:59	13	17
22:35:04	22:34:48	22:35:00	12	16
22:37:03	22:36:48	22:37:00	12	15
22:39:02	22:38:48	22:38:59	11	14
22:41:00	22:40:47	22:40:57	10	13
22:42:59	22:42:47	22:42:56	09	12
22:45:01	22:44:47	22:44:57	10	14
22:46:58	22:46:46	22:46:56	10	12
22:49:01	22:48:47	22:48:58	11	14
		Average and St Dev.	10.75(±2.47)	14(±2.93)

Table 3: DHT 11 Data Taken Every 5 Minutes Day 3

RRT	SDST	DACT	TL	RL
08:27:27	08:27:15	08:27:24	09	12
08:32:30	08:32:15	08:32:27	12	15
09:37:32	09:37:15	09:37:29	14	17
08:42:33	08:42:14	08:42:27	13	19
08:47:32	08:47:15	08:47:28	13	17
08:52:35	08:52:16	08:52:32	16	19
08:57:34	08:57:16	08:57:30	14	18
09:02:28	09:02:15	09:02:25	10	13
09:07:29	09:07:15	09:07:26	11	14
09:12:28	09:12:15	09:12:25	10	13
09:17:24	09:17:14	09:17:22	08	10
09:22:28	09:22:14	09:22:26	12	14
09:27:32	09:27:15	09:27:29	14	17
09:32:28	09:32:13	09:32:24	11	15
09:37:28	09:37:15	09:37:24	09	13
09:42:29	09:42:15	09:42:26	11	14
09:47:29	09:47:14	09:47:26	12	15
09:52:28	09:52:13	09:52:26	13	15
09:57:26	09:57:14	09:57:24	10	12
10:02:28	10:02:14	10:02:24	10	14
20:14:23	20:14:07	20:14:19	12	16
20:19:26	20:19:08	20:19:22	14	18
20:24:29	20:24:08	20:24:25	17	21
20:29:32	20:29:08	20:29:28	20	24
20:34:32	20:34:06	20:34:28	22	26
20:39:28	20:39:06	20:39:23	17	22
20:44:30	20:44:07	20:44:25	18	23
20:49:29	20:49:07	20:49:26	19	22
20:54:32	20:54:07	20:54:27	20	25
20:59:26	20:59:07	20:59:23	16	19
21:04:24	21:04:07	21:04:24	14	17
21:09:20	21:09:08	21:09:17	09	12
21:14:17	21:14:06	21:14:14	08	11
21:19:23	21:19:08	21:19:20	12	15
21:24:22	21:24:08	21:24:19	11	14
21:29:23	21:29:07	21:29:19	12	16
21:34:25	21:34:08	21:34:20	12	17
21:39:17	21:39:07	21:39:14	07	10
21:44:23	21:44:08	21:44:19	11	15
21:49:24	21:49:07	21:49:21	14	17
		Average and St Dev.	12.93(±3.55)	16.4(±4)

Table 4: DHT 11 Data Taken Every 5 Minutes Day 4

RRT	SDST	DACT	TL	RL
09:25:26	09:25:12	09:25:22	10	14
09:30:30	09:30:13	09:30:25	12	17
09:35:30	09:35:12	09:35:26	14	18
09:40:25	09:40:12	09:40:22	10	13
09:45:27	09:45:12	09:45:23	11	15
09:50:29	09:50:11	09:50:24	13	18
09:55:24	09:55:12	09:55:22	10	12
10:00:22	10:00:11	10:00:19	08	11
10:05:24	10:05:12	10:05:20	08	12
10:10:24	10:10:12	10:10:21	09	12
10:15:26	10:15:13	10:15:23	10	13
10:20:26	10:20:12	10:20:23	11	14
10:25:20	10:25:12	10:25:18	06	08
10:30:18	10:30:12	10:30:16	04	06
10:35:20	10:35:11	10:35:18	07	09
10:40:22	10:40:11	10:40:19	08	11
10:45:22	10:45:12	10:45:18	06	10
10:50:26	10:50:11	10:50:22	11	15
10:55:25	10:55:12	10:55:22	10	13
11:00:22	11:00:12	11:00:19	07	10
20:23:58	20:23:45	20:23:55	10	13
20:28:56	20:28:44	20:28:53	09	12
20:33:52	20:33:44	20:33:49	05	08
20:38:53	20:38:44	20:38:50	06	09
20:43:55	20:43:43	20:43:51	08	12
20:48:58	20:48:45	20:48:55	10	13
20:53:52	20:53:44	20:53:49	05	08
20:58:54	20:58:45	20:58:51	06	09
20:03:53	20:03:43	20:03:50	07	10
20:08:53	20:08:44	20:08:50	06	09
20:13:53	20:13:43	20:13:49	06	10
20:18:54	20:18:43	20:18:50	07	11
20:23:54	20:23:44	20:23:51	07	10
20:28:57	20:28:45	20:28:53	08	12
20:33:59	20:33:44	20:33:55	11	15
20:38:57	20:38:44	20:38:54	10	13
20:43:56	20:43:44	20:43:52	08	12
20:48:53	20:48:43	20:48:49	06	10
20:53:56	20:53:44	20:53:52	08	12
20:58:56	20:58:43	20:58:53	10	13
		Average and St Dev.	8.45(±2.32)	11.8(±2.70)

Table 5: DHT 11 Data Taken Every 10 Minutes Day 5

RRT	SDST	DACT	TL	RL
08:09:40	08:09:24	08:09:36	12	16
08:19:41	08:19:24	08:19:35	11	17
08:29: 36	08:29:23	08:29:32	09	13
08:39:36	08:39:24	08:39:32	08	12
08:49:35	08:49:24	08:49:31	07	11
08:59:37	08:59:23	08:59:33	10	14
09:09:35	09:09:23	09:09:33	10	12
09:19:34	09:19:22	09:19:30	08	12
09:29:43	09:29:24	09:29:37	13	19
09:39:42	09:39:24	09:39:38	14	18
09:49:40	09:49:23	09:49:36	13	17
09:59:39	09:59:23	09:59:35	12	16
10:09:35	10:09:23	10:09:31	08	12
10:19:33	10:19:22	10:19:30	08	11
10:29:36	10:29:24	10:29:33	09	12
10:39:37	10:39:24	10:39:33	09	13
10:49:43	10:49:23	10:49:30	07	10
10:59:36	10:59:22	10:59:33	11	14
10:09:36	10:09:24	10:09:34	10	12
10:19:37	10:19:24	10:19:33	09	13
21:17:26	21:17:14	21:17:22	08	12
21:27:26	21:27:15	21:27:24	09	11
21:37:28	21:37:15	21:37:26	11	13
21:47:28	21:47:15	21:47:25	10	13
21:57:29	21:57:14	21:57:26	12	15
22:07:25	22:07:13	22:07:21	08	12
22:17:23	22:17:13	22:17:20	07	10
22:27:22	22:27:13	22:27:19	06	09
22:37:23	22:37:13	22:37:21	08	10
22:47:21	22:47:13	22:47:19	06	08
22:57:26	22:57:14	22:57:24	10	12
23:07:29	23:07:16	23:07:26	10	13
23:17:26	23:17:16	23:17:24	08	10
23:27:25	23:27:14	23:27:23	09	11
23:37:29	23:37:14	23:37:26	12	15
23:47:31	23:47:14	23:47:27	13	17
23:57:27	23:57:14	23:57:24	10	13
00:07:25	00:07:13	00:07:23	10	12
00:17:28	00:17:13	00:17:24	11	15
00:27:25	00:27:14	00:27:22	08	11
		Average and St Dev.	9.6(±2.01)	12.9(±2.52)

Table 6: DHT 11 Data Taken Every 10 Minutes Day 6

RRT	SDST	DACT	TL	RL
08:47:27	08:47:17	08:47:25	08	10
08:57:28	08:57:17	08:57:24	07	11
09:07:27	09:07:17	09:07:25	08	10
09:17:28	09:17:18	09:17:28	10	12
09:27:33	09:27:19	09:27:30	11	14
09:37:29	09:37:17	09:37:27	10	12
09:47:31	09:47:18	09:47:28	10	13
09:57:25	09:57:17	09:57:23	06	08
10:07:24	10:07:17	10:07:22	05	07
10:17:27	10:17:17	10:17:25	08	10
10:27:28	10:27:17	10:27:26	09	11
10:37:32	10:37:20	10:37:30	10	12
10:47:30	10:47:17	10:47:27	10	13
10:57:31	10:57:17	10:57:28	11	14
10:07:29	10:07:17	10:07:27	10	12
10:17:29	10:17:18	10:17:27	09	11
10:27:28	10:27:18	10:27:26	08	10
10:37:28	10:37:16	10:37:25	09	12
10:47:27	10:47:17	10:47:24	07	10
10:57:30	10:57:17	10:57:26	09	13
20:12:54	20:12:42	22:12:52	10	12
20:22:51	20:22:43	22:22:48	05	08
20:32:51	20:32:42	22:32:49	07	09
20:42:52	20:42:42	22:42:50	08	10
20:52:56	20:52:42	22:52:52	10	14
21:02:55	21:02:43	23:02:51	08	12
21:12:55	21:12:45	21:12:52	07	10
21:22:55	21:22:42	21:22:52	10	13
21:32:54	21:32:42	21:32:49	07	12
21:42:56	21:42:42	21:42:53	11	14
21:52:55	21:52:42	21:52:51	09	13
22:02:55	22:02:43	21:02:51	08	12
22:12:55	22:12:42	21:12:51	09	13
22:22:54	22:22:42	21:22:51	09	12
22:32:52	22:32:42	21:32:49	07	10
22:42:54	22:42:43	21:42:52	09	11
22:52:56	22:52:43	21:52:53	10	13
23:02:52	23:02:42	21:02:50	08	10
23:12:56	23:12:41	23:12:53	12	15
23:22:53	23:22:42	23:22:50	08	11
		Average and St Dev.	8.68(±1.61)	11.48(±1.81)

Table 7: DHT 11 Data Taken Every 15 Minutes Day 7

RRT	SDST	DACT	TL	RL
07:40:33	07:40:23	07:40:30	07	10
07:55:35	07:55:23	07:55:33	10	12
08:10:39	08:10:25	08:10:36	11	14
08:25:35	08:25:22	08:25:32	10	13
08:40:35	08:40:22	08:40:32	10	13
10:55:33	08:55:23	10:55:31	08	10
10:10:38	09:10:23	10:10:33	10	15
10:25:37	09:25:23	10:25:35	12	14
10:40:39	09:40:23	10:40:35	12	16
10:55:37	09:55:23	10:55:34	11	14
11:10:38	10:10:24	11:10:34	10	15
11:25:36	10:25:23	11:25:33	10	13
11:40:37	10:40:23	11:40:34	11	14
11:55:36	10:55:24	11:55:33	09	12
11:10:34	11:10:24	11:10:32	08	10
11:25:36	11:25:23	11:25:33	10	13
11:40:37	11:40:23	11:40:34	11	14
11:55:35	11:55:24	11:55:32	08	11
12:10:38	12:10:24	12:10:34	10	14
12:25:35	12:25:23	12:25:33	10	12
20:13:22	20:13:05	20:13:17	12	17
20:28:19	20:28:05	20:28:16	11	14
20:43:16	20:43:07	20:43:14	07	09
20:58:15	20:58:05	20:58:13	08	10
21:13:13	21:13:05	21:13:11	06	08
21:28:15	21:28:06	21:28:13	07	09
21:43:15	21:43:05	21:43:13	08	10
21:58:19	21:58:05	21:58:16	11	14
22:13:20	22:13:05	22:13:17	12	15
22:28:17	22:28:05	22:28:15	10	12
22:43:17	22:43:04	22:43:14	10	13
22:58:18	22:58:04	22:58:14	10	14
23:13:17	23:13:05	23:13:13	08	12
23:28:20	23:28:05	23:28:15	10	15
23:43:17	23:43:05	23:43:14	09	12
23:58:18	23:58:05	23:58:14	09	13
00:13:16	00:13:06	00:13:14	08	10
00:28:16	00:28:05	00:28:12	07	11
00:43:19	00:43:05	00:43:15	10	14
00:58:16	00:58:04	00:58:11	07	12
		Average and St Dev.	9.45(±1.62)	12.58(±2.09)

Table 8: DHT 11 Data Taken Every 15 Minutes Day 8

RRT	SDST	DACT	TL	RL
07:32:27	07:32:15	07:32:25	10	12
07:47:27	07:47:16	07:47:25	9	11
08:02:25	08:02:14	08:02:24	10	11
08:17:25	08:17:15	08:17:23	8	10
08:32:26	08:32:13	08:32:23	10	13
08:47:29	08:47:15	08:47:26	11	14
09:02:30	09:02:15	09:02:27	12	15
09:17:28	09:17:16	09:17:26	10	12
09:32:27	09:32:15	09:32:25	10	12
09:47:27	09:47:14	09:47:26	12	13
10:02:22	10:02:12	10:02:20	8	10
10:17:28	10:17:14	10:17:26	12	14
10:32:23	10:32:13	10:32:21	8	10
10:47:24	10:47:15	10:47:22	7	9
11:02:23	11:02:15	11:02:22	7	8
11:17:26	11:17:16	11:17:23	7	10
11:32:24	11:32:14	11:32:22	8	10
11:47:27	11:47:15	11:47:25	10	12
12:02:25	12:02:14	12:02:24	10	11
12:17:25	12:17:15	12:17:23	8	10
12:32:22	12:32:13	12:32:20	7	9
12:47:23	12:47:15	12:47:21	6	8
13:02:22	13:02:15	13:02:21	6	7
13:17:22	13:17:12	13:17:20	8	10
13:32:28	13:32:15	13:32:25	10	13
13:47:30	13:47:16	13:47:27	11	14
14:02:30	14:02:15	14:02:28	13	15
14:17:25	14:17:15	14:17:23	8	10
14:32:27	14:32:15	14:32:25	10	12
14:47:27	14:47:14	14:47:25	11	13
15:02:25	15:02:14	15:02:24	10	11
15:17:24	15:17:14	15:17:22	8	10
15:32:24	15:32:15	15:32:22	7	9
15:47:22	15:47:14	15:47:20	6	8
16:02:26	16:02:14	16:02:23	9	12
16:17:27	16:17:14	16:17:25	11	13
16:32:24	16:32:14	16:32:22	8	10
16:47:25	16:47:14	16:47:23	9	11
17:02:24	17:02:15	17:02:22	7	9
17:17:24	17:17:14	17:17:22	8	10
		Average and St Dev.	9(±1.83)	11.03(±1.99)

Table 9: DHT 11 Data Taken Every 5 Minutes Day 9

RRT	SDST	DACT	TL	RL
09:16:37	09:16:29	09:16:36	7	8
09:21:39	09:21:30	09:21:38	8	9
09:26:36	09:26:28	09:26:34	6	8
09:31:41	09:31:29	09:31:39	10	12
09:36:39	09:36:28	09:36:37	9	11
09:41:41	09:41:28	09:41:38	10	13
09:46:42	09:46:28	09:46:39	11	14
09:51:40	09:51:28	09:51:38	10	12
09:56:37	09:56:27	09:56:35	8	10
10:01:35	10:01:27	10:01:33	6	8
10:06:35	10:06:28	10:06:34	6	7
10:11:35	10:11:29	10:11:34	5	6
10:16:38	10:16:30	10:16:37	7	8
10:21:37	10:21:28	10:21:35	7	9
10:26:40	10:26:28	10:26:38	10	12
10:31:38	10:31:28	10:31:36	8	10
10:36:42	10:36:27	10:36:39	12	15
10:41:40	10:41:28	10:41:38	10	12
10:46:38	10:46:28	10:46:35	7	10
10:51:40	10:51:29	10:51:39	10	11
10:56:39	10:56:29	10:56:37	8	10
11:01:41	11:01:28	11:01:40	12	13
11:06:37	11:06:29	11:06:36	7	8
11:11:35	11:11:27	11:11:33	6	8
11:16:37	11:16:28	11:16:35	7	9
11:21:35	11:21:28	11:21:33	5	7
11:26:35	11:26:28	11:26:34	6	7
11:31:37	11:31:27	11:31:34	7	10
11:36:38	11:36:29	11:36:37	8	9
11:41:39	11:41:28	11:41:38	10	11
11:46:39	11:46:27	11:46:37	10	12
11:51:38	11:51:28	11:51:37	9	10
11:56:36	11:56:28	11:56:35	7	8
12:01:40	12:01:28	12:01:38	10	12
12:06:39	12:06:29	12:06:37	8	10
12:11:37	12:11:27	12:11:35	8	10
12:16:39	12:16:28	12:16:38	10	11
12:21:41	12:21:28	12:21:40	12	13
12:26:39	12:26:28	12:26:36	8	11
12:31:37	12:31:27	12:31:35	8	10
		Average and St Dev.	8.33(±1.90)	10.10(±2.10)

Table 10: DHT 11 Data Taken Every 5 Minutes Day 10

RRT	SDST	DACT	TL	RL
15:42:33	15:42:22	15:42:31	9	11
15:47:33	15:47:23	15:47:31	8	10
15:52:33	15:52:21	15:52:31	10	12
15:57:30	15:57:20	15:57:28	8	10
16:02:32	16:02:20	16:02:30	10	12
16:07:28	16:07:20	16:07:26	6	8
16:12:30	16:12:21	16:12:28	7	9
16:17:35	16:17:21	16:17:32	11	14
16:22:36	16:22:21	16:22:33	12	15
16:27:37	16:27:22	16:27:34	12	15
16:32:35	16:32:21	16:32:33	12	14
16:37:35	16:37:22	16:37:32	10	13
16:42:34	16:42:22	16:42:32	10	12
16:47:31	16:47:21	16:47:29	8	10
16:52:30	16:52:20	16:52:28	8	10
16:57:28	16:57:20	16:57:26	6	8
17:02:27	17:02:20	17:02:26	6	7
17:07:26	17:07:20	17:07:25	5	6
17:12:29	17:12:20	17:12:27	7	9
17:17:30	17:17:20	17:17:28	8	10
17:22:31	17:22:21	17:22:28	7	10
17:27:30	17:27:21	17:27:28	7	9
17:32:33	17:32:21	17:32:31	10	12
17:37:34	17:37:21	17:37:31	10	13
17:42:31	17:42:21	17:42:28	7	10
17:47:33	17:47:21	17:47:31	10	12
17:52:32	17:52:21	17:52:30	9	11
17:57:32	17:57:22	17:57:30	8	10
18:02:29	18:02:21	18:02:27	6	8
18:07:27	18:07:20	18:07:26	6	7
18:12:31	18:12:21	18:12:29	8	10
18:17:32	18:17:21	18:17:30	9	11
18:22:32	18:22:20	18:22:30	10	12
18:27:34	18:27:21	18:27:32	11	13
18:32:33	18:32:21	18:32:31	10	12
18:37:32	18:37:21	18:37:30	9	11
18:42:31	18:42:21	18:42:29	8	10
18:47:28	18:47:20	18:47:26	6	8
18:52:34	18:52:22	18:52:32	10	12
18:57:33	18:57:21	18:57:31	10	12
		Average and St Dev.	8.6(±1.88)	10.70(±2.16)

HC-SR04 Prototype Timestamp Data is given below.

Table 2: HC-SR04 Data for Day 2

RRT	SDST	DACT	TL	RL
09:12:44	09:12:32	09:12:41	09	12
09:20:28	09:20:13	09:20:24	11	15
09:38:37	09:38:22	09:38:34	12	15
09:55:24	09:55:10	09:55:22	12	14
10:05:55	10:05:43	10:05:51	08	12
10:20:22	10:20:09	10:20:19	10	13
10:45:43	10:45:25	10:45:41	16	18
10:57:30	10:57:14	10:57:28	14	16
11:10:52	11:10:35	11:10:50	15	17
11:17:59	11:17:40	11:17:55	15	19
11:32:48	11:32:34	11:32:46	12	14
11:47:36	11:47:21	11:47:32	11	15
12:05:36	12:05:19	12:05:34	15	17
12:20:16	12:20:02	12:20:14	12	14
12:40:24	12:40:09	12:40:22	13	15
15:16:40	15:16:24	15:16:39	15	16
15:30:34	15:30:21	15:30:33	12	13
15:40:42	15:40:32	15:40:40	08	10
15:51:05	15:50:47	15:51:02	15	18
15:57:40	15:57:20	15:57:37	17	20
16:07:37	16:07:15	16:07:34	19	22
16:18:32	16:18:11	16:18:29	18	21
16:33:35	16:33:12	16:33:32	20	23
16:44:43	16:44:19	16:44:41	22	24
16:56:24	16:56:07	16:56:22	15	17
19:40:58	19:40:42	19:40:56	14	16
19:48:42	19:48:27	19:48:39	12	15
19:58:16	19:58:02	19:58:13	11	14
20:10:54	20:10:39	20:10:51	12	15
20:19:27	20:19:13	20:19:25	12	14
20:31:29	20:31:17	20:31:27	10	12
20:39:21	20:39:10	20:39:17	07	11
20:43:33	20:43:17	20:43:30	13	16
20:56:58	20:56:41	20:56:54	13	17
21:07:37	21:07:19	21:07:34	15	18
21:19:10	21:18:56	21:19:08	12	14
21:29:38	21:29:23	21:29:35	12	15
21:42:16	21:42:03	21:42:13	10	13
21:55:27	21:55:14	21:55:25	11	13
22:10:19	22:10:07	22:10:17	10	12
		Average and St Dev.	13(±3.24)	15.63(±3.21)

Table 3: HC-SR04 Data for Day 3

RRT	SDST	DACT	TL	RL
07:39:26	07:39:12	07:39:23	11	14
07:47:32	07:47:19	07:47:29	10	13
07:59:14	07:59:04	07:59:12	08	10
08:08:39	08:08:27	08:08:37	10	12
08:08:34	08:08:23	08:08:33	10	11
08:18:27	08:18:13	08:18:24	11	14
08:25:30	08:25:17	08:25:28	11	13
08:34:24	08:34:09	08:34:21	12	15
08:43:23	08:43:07	08:43:20	13	16
08:50:40	08:50:26	08:50:36	10	14
08:59:05	08:58:52	08:59:02	10	13
09:21:48	09:21:34	09:21:44	10	14
09:42:40	09:42:23	09:42:36	13	17
09:57:49	09:57:38	09:57:47	09	11
10:10:34	10:10:21	10:10:31	10	13
10:19:35	10:19:21	10:19:32	11	14
10:28:40	10:28:25	10:28:37	12	15
13:14:30	13:14:17	13:14:27	10	13
13:27:45	13:27:32	13:27:43	11	13
13:40:47	13:40:35	13:40:45	10	12
13:54:00	13:53:49	13:53:56	07	11
14:05:21	14:05:11	14:05:18	07	10
14:22:26	14:22:18	14:22:24	06	08
14:33:46	14:33:37	14:33:44	07	09
14:45:47	14:45:32	14:45:44	12	15
14:51:29	14:51:16	14:51:26	10	13
15:04:03	15:03:46	15:04:00	14	17
15:16:51	15:16:38	15:16:48	10	13
15:30:44	15:30:32	15:30:40	08	12
15:39:06	15:38:56	15:39:04	08	10
15:47:50	15:47:43	15:47:48	05	07
15:58:17	15:58:05	15:58:15	10	12
20:12:29	20:12:18	20:12:27	09	11
20:21:25	20:21:15	20:21:23	08	10
20:34:23	20:34:11	20:34:21	10	12
20:45:35	20:45:21	20:45:32	11	14
20:53:16	20:53:04	20:53:14	10	12
21:05:04	21:04:49	21:05:00	11	15
21:18:08	21:17:57	21:18:06	09	11
21:25:18	21:25:06	21:25:16	10	12
		Average and St Dev.	9.85(±1.89)	12.53(±2.23)

Table 4: HC-SR04 Data for Day 4

RRT	SDST	DACT	TL	RL
09:17:12	09:17:03	09:17:10	07	09
09:23:22	09:23:06	09:23:16	10	12
09:29:29	09:29:18	09:29:27	09	11
09:46:07	09:45:57	09:46:05	08	10
09:54:32	09:54:21	09:54:29	08	11
10:08:17	10:08:05	10:08:15	10	12
10:17:41	10:17:28	10:17:39	11	13
10:26:37	10:26:22	10:26:34	12	15
10:32:50	10:32:34	10:32:47	13	16
10:41:25	10:41:11	10:41:23	12	14
10:50:53	10:50:38	10:50:49	11	15
10:59:42	10:59:28	10:59:40	12	14
11:13:58	11:13:45	11:13:55	10	13
11:25:08	11:24:56	11:25:06	10	12
11:36:06	11:35:47	11:36:03	16	19
11:43:17	11:42:56	11:43:14	18	21
11:52:05	11:51:43	11:52:02	19	22
12:02:56	12:02:38	12:02:54	16	18
15:42:50	15:42:33	15:42:48	15	17
15:56:17	15:56:01	15:56:14	13	16
16:13:21	16:13:05	16:13:19	14	16
16:19:58	16:19:43	16:19:55	12	15
16:27:56	16:27:44	16:27:54	10	12
16:35:01	16:34:51	16:34:59	08	10
16:39:31	16:39:20	16:39:29	09	11
16:47:27	16:47:17	16:47:25	08	10
16:52:43	16:52:30	16:52:40	10	13
21:27:45	21:27:31	21:27:42	11	14
21:43:34	21:43:22	21:43:32	10	12
21:54:44	21:54:34	21:54:42	08	10
22:12:40	22:12:25	22:12:37	12	15
22:21:13	22:21:05	22:21:10	05	08
22:29:53	22:29:42	22:29:50	08	11
22:35:50	22:35:36	22:35:47	11	14
22:41:45	22:41:33	22:41:43	10	12
22:50:33	22:50:20	22:50:30	10	13
22:59:33	22:59:24	22:59:31	07	09
23:08:33	23:08:21	23:08:31	10	12
23:15:43	23:15:34	23:15:40	06	09
23:24:24	23:24:14	23:24:21	07	10
		Average and St Dev.	10.65(±3.11)	13.15(±3.24)

Table 5: HC-SR04 Data for Day 5

RRT	SDST	DACT	TL	RL
08:31:25	08:31:02	08:31:22	20	23
08:43:40	08:43:18	08:43:36	18	22
08:51:55	08:51:35	08:51:53	18	20
08:59:31	08:59:12	08:59:27	15	19
09:10:01	09:09:43	09:09:57	14	18
09:18:13	09:17:54	09:18:10	16	19
09:26:43	09:26:23	09:26:40	17	20
09:57:08	09:56:46	09:57:05	19	22
10:10:44	10:10:23	10:10:40	17	21
10:19:47	10:19:30	10:19:44	14	17
10:32:12	10:31:56	10:32:09	13	16
10:43:46	10:43:32	10:43:44	12	14
10:53:11	10:52:57	10:53:08	11	14
11:05:51	11:05:36	11:05:48	12	15
11:16:36	11:16:20	11:16:31	11	16
11:25:22	11:25:08	11:25:18	10	14
11:35:38	11:35:24	11:35:36	12	14
11:44:36	11:44:21	11:44:32	11	15
11:53:29	11:53:16	11:53:26	10	13
12:07:20	12:07:08	12:07:17	09	12
12:12:63	12:12:49	12:13:00	11	14
12:26:13	12:25:58	12:26:10	12	15
12:34:53	12:34:40	12:34:50	10	13
12:45:33	12:45:23	12:45:30	07	10
12:58:21	12:58:04	12:58:20	16	17
17:22:03	17:21:45	17:22:00	15	18
17:29:51	17:29:35	17:29:48	13	16
17:47:46	17:47:30	17:47:44	14	16
17:56:51	17:56:34	17:56:47	13	17
18:18:58	18:18:45	18:18:55	10	13
18:25:08	18:24:56	18:25:05	09	12
18:33:50	18:33:36	18:33:47	11	14
18:48:10	18:47:57	18:48:07	10	13
18:56:13	18:56:01	18:56:10	09	12
19:15:17	19:15:07	19:15:15	08	10
19:24:43	19:24:31	19:24:40	09	12
19:36:24	19:36:12	19:36:20	08	12
19:45:24	19:45:10	19:45:21	11	14
19:51:49	19:51:41	19:51:47	06	08
20:01:16	20:01:06	20:01:13	07	10
		Average and St Dev.	12.2(±3.50)	15.25(±3.59)

Table 6: HC-SR04 Data for Day 6

RRT	SDST	DACT	TL	RL
07:20:44	07:20:33	07:20:40	07	11
07:34:32	07:34:22	07:34:30	08	10
07:42:35	07:42:21	07:42:32	11	14
07:51:25	07:51:12	07:51:22	10	13
08:03:60	08:03:45	08:03:57	12	15
08:13:10	08:12:57	08:13:07	10	13
08:20:36	08:20:22	08:20:33	11	14
08:29:44	08:29:32	08:29:42	10	12
08:36:53	08:36:40	08:36:50	10	13
08:50:58	08:50:44	08:50:55	11	14
08:58:46	08:58:36	08:58:44	08	10
09:10:22	09:10:15	09:10:20	05	07
09:18:06	09:17:58	09:19:04	06	08
09:30:58	09:30:50	09:30:56	06	08
09:37:17	09:37:07	09:37:15	08	10
09:49:55	09:49:44	09:49:52	08	11
09:58:14	09:58:04	09:58:12	08	10
10:10:23	10:10:13	10:10:20	07	10
10:22:10	10:21:58	10:22:07	09	12
10:32:18	10:32:08	10:32:16	08	10
10:41:09	10:41:01	10:41:06	05	08
10:48:29	10:48:22	10:48:27	05	07
10:58:57	10:58:48	10:58:56	08	09
11:11:47	11:11:37	11:11:45	08	10
16:37:36	16:37:25	16:37:32	07	11
16:46:36	16:46:23	16:46:33	10	13
16:55:46	16:55:34	16:55:43	09	12
17:12:58	17:12:43	17:12:55	12	15
17:20:32	17:20:18	17:20:29	11	14
17:28:34	17:28:22	17:28:32	10	12
17:37:45	17:37:31	17:37:42	11	14
17:45:52	17:45:40	17:45:50	10	12
17:54:38	17:54:24	17:54:35	11	14
18:30:58	18:30:47	18:30:55	08	11
18:41:19	18:41:09	18:41:17	08	10
18:50:68	18:50:53	18:51:06	13	15
18:59:45	18:59:32	18:59:42	10	13
19:12:38	19:12:22	19:12:34	12	16
19:22:42	19:22:29	19:22:39	10	13
19:32:51	19:32:37	19:32:48	11	14
		Average and St Dev.	9.05.2(±2.07)	11.7(±2.36)

Table 7: HC-SR04 Data for Day 7

RRT	SDST	DACT	TL	RL
08:37:38	08:37:20	08:37:36	16	18
08:46:51	08:46:35	08:46:47	12	16
08:54:57	08:54:40	08:54:52	12	17
09:11:06	09:10:50	09:11:03	13	16
09:20:10	09:19:56	09:20:07	11	14
09:27:59	09:27:44	09:27:57	13	15
09:41:05	09:40:51	09:41:04	13	14
09:49:16	09:49:03	09:49:13	10	13
09:58:31	09:58:17	09:58:28	11	14
10:09:41	10:09:29	10:09:39	10	12
10:18:18	10:18:01	10:18:15	14	17
10:30:50	10:30:31	10:30:47	16	19
10:42:60	10:42:40	10:42:57	17	20
10:51:46	10:51:32	10:51:43	11	14
11:07:19	11:07:04	11:07:16	12	15
11:20:36	11:20:20	11:20:32	12	16
11:31:50	11:31:33	11:31:47	14	17
11:40:28	11:40:15	11:40:25	10	13
11:48:24	11:48:12	11:48:22	10	12
11:57:45	11:57:31	11:57:42	11	14
12:12:38	12:12:25	12:12:35	10	13
12:25:17	12:25:07	12:25:15	08	10
15:33:35	15:33:20	15:33:31	11	15
15:49:02	15:48:44	15:48:59	15	18
15:58:09	15:57:49	15:58:07	18	20
16:13:44	16:13:30	16:13:41	11	14
16:24:58	16:24:43	16:24:55	12	15
16:35:35	16:35:23	16:35:33	10	12
16:45:06	16:44:56	16:45:04	08	10
16:54:47	16:54:31	16:54:45	14	16
17:20:49	17:20:35	17:20:47	12	14
17:32:49	17:32:36	17:32:46	10	13
17:46:31	17:46:19	17:46:29	10	12
17:56:05	17:55:49	17:56:01	12	16
18:10:12	18:10:02	18:10:10	08	10
18:21:25	18:21:15	18:21:24	09	10
18:33:14	18:33:03	18:33:10	07	11
18:44:39	18:44:30	18:44:37	07	09
18:58:06	18:57:52	18:58:03	11	14
19:11:32	19:11:20	19:11:30	10	12
		Average and St Dev.	11.53(±2.57)	14.25(±2.77)

Table 8: HC-SR04 Data for Day 8

RRT	SDST	DACT	TL	RL
08:21:18	08:21:12	08:21:17	5	6
08:30:23	08:30:15	08:30:21	6	8
08:38:10	08:38:02	08:38:08	6	8
08:48:43	08:48:33	08:48:41	8	10
09:03:54	09:03:45	09:03:52	7	9
09:12:27	09:12:20	09:12:26	6	7
09:20:30	09:20:17	09:20:27	10	13
09:32:15	09:32:05	09:32:13	8	10
09:45:59	09:45:45	09:45:56	11	14
09:52:22	09:52:10	09:52:20	10	12
10:05:22	10:05:09	10:05:19	10	13
10:12:18	10:12:08	10:12:16	8	10
10:22:56	10:22:45	10:22:53	8	11
10:30:05	10:29:55	10:30:03	8	10
10:36:22	10:36:12	10:36:19	7	10
10:48:21	10:48:09	10:48:19	10	12
10:56:20	10:56:10	10:56:18	8	10
11:13:42	11:13:34	11:13:40	6	8
11:26:40	11:26:28	11:26:38	10	12
14:20:49	14:20:40	14:20:48	8	9
14:33:42	14:33:30	14:33:40	10	12
14:47:35	14:47:22	14:47:33	11	13
14:56:28	14:56:17	14:56:26	9	11
15:07:33	15:07:25	15:07:31	6	8
15:15:15	15:15:03	15:15:13	10	12
15:28:38	15:28:21	15:28:36	15	17
15:42:50	15:42:35	15:42:47	12	15
15:54:03	15:53:51	15:54:01	10	12
17:03:58	17:03:42	17:03:56	14	16
17:21:45	17:21:31	17:21:43	12	14
17:35:05	17:34:47	17:35:02	15	18
17:46:09	17:45:56	17:46:06	10	13
17:56:26	17:56:14	17:56:24	10	12
18:10:22	18:10:12	18:10:20	8	10
18:22:22	18:22:10	18:22:20	10	12
18:36:52	18:36:43	18:36:51	8	9
18:48:42	18:48:27	18:48:40	13	15
18:57:53	18:57:41	18:57:51	10	12
19:18:24	19:18:11	19:18:23	12	13
19:39:22	19:39:12	19:39:20	8	10
		Average and St Dev.	9.33(±2.45)	11.40(±2.68)

Table 9: HC-SR04 Data for Day 9

RRT	SDST	DACT	TL	RL
08:29:12	08:29:03	08:29:10	7	9
08:44:35	08:44:25	08:44:33	8	10
08:57:24	08:57:12	08:57:22	10	12
09:19:19	09:19:05	09:19:17	12	14
09:31:46	09:31:33	09:31:43	10	13
09:47:30	09:47:21	09:47:29	8	9
09:55:56	09:55:46	09:55:54	8	10
10:07:00	10:06:49	10:06:58	9	11
10:17:31	10:17:21	10:17:29	8	10
10:30:35	10:30:23	10:30:33	10	12
10:42:42	10:42:29	10:42:39	10	13
10:54:32	10:54:22	10:54:30	8	10
12:14:24	12:14:15	12:14:22	7	9
12:28:30	12:28:18	12:28:28	10	12
12:46:50	12:46:39	12:46:48	9	11
12:55:55	12:55:45	12:55:53	8	10
14:12:48	14:12:36	14:12:46	10	12
14:28:48	14:28:34	14:28:46	12	14
14:37:30	14:37:17	14:37:27	10	13
14:49:43	14:49:31	14:49:41	10	12
16:02:20	16:02:06	16:02:18	12	14
16:14:23	16:14:11	16:14:21	10	12
16:27:23	16:27:10	16:27:20	10	13
16:39:25	16:39:16	16:39:23	7	9
16:49:50	16:49:38	16:49:48	10	12
17:04:52	17:04:42	17:04:50	8	10
17:13:25	17:13:14	17:13:23	9	11
17:27:25	17:27:13	17:27:23	10	12
17:40:58	17:40:48	17:40:56	8	10
17:52:48	17:52:35	17:52:45	10	13
20:24:01	20:23:52	20:24:00	8	9
20:35:49	20:35:41	20:35:48	7	8
20:48:70	20:48:55	20:48:67	12	15
20:57:22	20:57:10	20:57:20	10	12
21:10:50	21:10:34	21:10:47	13	16
21:22:59	21:22:45	21:22:57	12	14
21:30:20	21:30:08	21:30:17	9	12
21:42:24	21:42:16	21:42:22	6	8
21:52:28	21:52:18	21:52:26	8	10
22:07:17	22:07:10	22:07:15	5	7
		Average and St Dev.	9.2(±1.79)	11.33(±2.06)

Table 10: HC-SR04 Data for Day 10

RRT	SDST	DACT	TL	RL
10:09:38	10:09:28	10:09:36	8	10
10:18:29	10:18:22	10:18:28	6	7
10:29:53	10:29:44	10:29:51	7	9
10:42:23	10:42:10	10:42:20	10	13
10:54:42	10:54:32	10:54:40	8	10
11:25:48	11:25:37	11:25:46	9	11
11:38:01	11:37:49	11:37:59	10	12
11:49:01	11:48:51	11:48:59	8	10
11:56:33	11:56:23	11:56:31	8	10
12:18:49	12:18:39	12:18:48	9	10
12:37:43	12:37:34	12:37:41	7	9
12:49:26	12:49:18	12:49:25	7	8
14:27:39	14:27:30	14:27:37	7	9
14:38:53	14:38:44	14:38:52	8	9
14:46:38	14:46:27	14:46:36	9	11
14:57:10	14:56:57	14:56:67	10	13
15:06:05	15:05:53	15:05:63	10	12
15:29:53	15:29:39	15:29:51	12	14
15:40:58	15:40:43	15:40:55	12	15
15:51:35	15:51:19	15:51:32	13	16
16:12:22	16:12:08	16:12:20	12	14
16:23:28	16:23:17	16:23:26	9	11
16:33:15	16:33:05	16:33:13	8	10
16:44:13	16:44:01	16:44:11	10	12
16:53:40	16:53:29	16:53:38	9	11
19:03:47	19:03:37	19:03:45	8	10
19:10:59	19:10:47	19:10:57	10	12
19:25:59	19:25:50	19:25:57	7	9
19:34:33	19:34:25	19:34:31	6	8
19:46:41	19:46:29	19:46:39	10	12
19:53:50	19:53:38	19:53:48	10	12
20:09:28	20:09:17	20:09:26	9	11
20:17:30	20:17:20	20:17:28	8	10
20:26:48	20:26:35	20:26:45	10	13
20:38:32	20:38:22	20:38:30	8	10
20:49:26	20:49:15	20:49:24	9	11
20:57:44	20:57:32	20:57:42	10	12
21:15:50	21:15:40	21:15:48	8	10
21:25:28	21:25:16	21:25:26	10	12
21:36:13	21:36:02	21:36:11	9	11
		Average and St Dev.	8.95(±1.62)	10.98(±1.91)