USER IDENTIFICATION IN BLOCKCHAIN BY SPATIO-TEMPORAL FINGERPRINTS

Thesis Submitted By:
Muhammad Tehseen Tahir - 01-243171-029

Supervised By:
Dr. Muhammad Muzammal

A dissertation submitted to the Department of Computer Science, Bahria University, Islamabad as a partial fulfillment of the requirements for the award of the degree of Masters in Computer Science

Session (2017-2019)

Bahria University
Discovering Knowledge

MS-13

Thesis Completion Certificate

Scholar's Name: Muhammad Tehseen Tahir Registration No.31970

Programme of Study: Master of Science in Computer Science (MSCS)

Thesis Title: User Identification in Blockchain By Spatio-Temporal Fingerprints

---

It is to certify that the above student's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for Evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at _____ that is within the permissible limit set by the HEC for the MS/ MPhil degree thesis. I have also found the thesis in a format recognized by the BU for the MS/MPhil thesis.

Principal Supervisor's Signature: Dr. Muhammad Muzammal

Date: 17th July, 2019 **Name:** _____

Bahria University
Discovering Knowledge

MS-14A

Author's Declaration

I, Muhammad Tehseen Tahir hereby state that my MS thesis titled "User Identification in Blockchain By Spatio-Temporal Fingerprints" is my own work and has not been submitted previously by me for taking any degree from this university (Bahria University, Islamabad)

or anywhere else in the country/world.

At any time if my statement is found to be incorrect even after my Graduate the university has the right to withdraw/cancel my MS degree.

Name of scholar: Muhammad Tehseen Tahir
Date: 17th July, 2019

Bahria University
Discovering Knowledge

MS-14B

Plagiarism Undertaking

I, solemnly declare that research work presented in the thesis titled "User Identification in Blockchain By Spatio-Temporal Fingerprints" is solely my research work with no significant contribution from any other person. Small contribution / help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Bahria University towards plagiarism. Therefore I as an Author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred / cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the university reserves the right to withdraw / revoke my MS degree and that HEC and the University has the right to publish my name on the HEC / University website on which names of students are placed who submitted plagiarized thesis.

Student / Author's Sign: _____

Name of the Student: Muhammad Tehseen Tahir

Abstract

User identification using the spatio-temporal geometries or trajectories has always been the interest of many researchers and is a very specific topic in fields related to data. Nowadays large spatio-temporal data are collected using different techniques including the smartphone’ s GPS. However, one of the major concerned issue regarding the popularity of GPS-based devices and systems is large scalability of the personal location information (that is often highly dimensional) generated by them and the sharing of that massive data with applications or identity systems. Traditional user verification techniques usually separately consider spatial and temporal approaches. Although there have also some work that has been done to integrate both the spatial and temporal information for user identity prediction but most of them suffer from the overfitting problem because of the large number of spatio-temporal trajectory patterns. Blockchain technology recently introduced in several areas after the successful working in the domain of crypto-currencies. Advent of blockchain can help resolving the concern of large scalability of mobility data by its reliable storage capacity, immutability and decentralized trustless data processing features. We consider a spatio-temporal Blockchain that registers both time and location attributes of the users.

In this research, we propose a novel approach to uniquely identify an individual by using its spatio-temporal fingerprints which are stored in blockchain. Fingerprint’s defines the spatial data points or the spatial trajectory of that respective individual. Proving this research, we developed a prototype blockchain system using the Hyperledger Fabric in which user spatio-temporal fingerprints are recorded on the basis of user mobile GPS data. And then we verify the user using the previously recorded data in blockchain and user-given location data to the system. Furthermore, we generate a new private key for the individual after the guaranteed verification steps satisfying our set threshold.

Acknowledgments

BISMILLAH HIR REHMAN NIR RAHIM,
In the name of Allah, the Most Gracious and the Most Merciful. I would like to thank Allah Almighty who gave me the strength and courage to complete this research work. I would also like to express my special thanks to my mentor and supervisor Dr. Muhammad Muzammal for his valuable and constructive suggestions during the planning and development of this research. I also like to thank Dr. Samabia Tehsin for keeping my progress on schedule. I extend my gratitude to all teachers and my parents for constant guidance and moral support.

MUHAMMAD TEHSEEN TAHIR
Bahria University Islamabad, Pakistan

July 2019

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

GPS	Global Positing System
API	Application Programming Interface
SDK	Software Development Kit
CURD	Create Update Read Delete
DLT	Distributed Ledger Technology
WSL	Windows Subsystem Linux
POI	Point of Interests
JS	JavaScript
Lat	Latitude
Log	Longitude
DB	Database

Chapter 1

Introduction

1.1 Introduction

Many organizations have been working on the collection of location-based data for identification and verification of objects such as devices, wireless systems or some individual (person). In the mean time engineers, data analysts scientists in many fields have also been capturing large complex datasets, such as terabyte's of data is being received on daily basis from spaceborne instruments, temporal and spectral-resolution remote IoT systems, and other hand-held devices such as smartphone's [2]. Due to the large scalability and incessant nature of such location data is the major concern. The concern is regarding to the storage capacity and the security of such data.

Currently, organizations are storing this type of data in traditional centralized databases which are trusted but authenticated to single party. Due to the advent and recent interest of many organizations in blockchain covers the issue of large scalability of this data, and the security about loss of this data by its decentralized nature. In distributed ledgers (blockchain) the recorded information like spatial trajectory neither can be modified nor deleted. Previously the research in spatial and temporal data models and registration systems has mostly been completed independently. Spatial database research provides more focus on supporting the modeling and then extracting of geometries that are linked with objects in a database. Temporal (time information) databases have focused on providing the knowledge recorded in a database about the present state of the real world to include the past [3]. And hence, when there is an integration of space and time, they are actually dealing with geometries that makes changes over the time [3]. Generally, mobile produced data comprises of the historical information of a user' s visiting sequence

(history) that is generated by the GPS technology, which is significantly more precise positioning technique [4]. The mobile data includes the detailed version of the user' s visited locations/history and comparable time-stamps. The work that is going on user' s visiting movement with mobile data, oftenly location prediction, can potentially benefit in many areas, like in mobile advertising, disaster predication and identification [5].
Also, the traditional location prediction techniques on mobile data uses the spatial trajectory patterns. For example, spatio-temporal data grows at a much bigger rate as compared to the transactions data that is currently supported by financial Blockchain systems. Furthermore, the proof-of-work or smart contract for spatio-temporal data also requires proof-of-location processing. Blockchain systems over the traditional databases value proposition is the data integrity through cryptographically signed history. Large enterprises and service providers such as Google requires spatio-temporal data analytics for providing continuous services in a given time and location. Formulating this research problem we should also be worried about that a spatio-temporal Blockchain system building should consider the fact of secure data storage with the efficient query processing simultaneously such that the database like analytics are also possible on the Blockchain.

1.2 Blockchain

Blockchain as a paradigm-shift technology that is first used in the crypto-currencies and has recently using and backing into a variety of fields and application domains. The Blockchain technology has showed its applicability for business solutions in sectors including financing, healthcare, education and others [6]. For example, blockchain can be used to identify our car. Our car can only be started once we tap the right pattern. We can also use this technology to activate our smartphone, our smartphone will only be used or in functional state if we type the right PIN code. These both work can protect our privacy and ownership. The problem with previously used forms of smart property is that the key is usually is in the form of physical, such as the SIM card or car key, and usually it cannot be easily copied/transferred. The Blockchain reduces the risks of these by allowing developers to replicate and then replace a lost protocol.

1.2.1 Technology behind Blockchain

Blockchain is essentially a distributed database that is secured by design and shared across nodes geographically [7] and can store any type of record(s). Users, that are acting as the node's can only edit the parts of the blockchain they own, which makes this technology highly secure. But at the same time anyone with access to it can see the data, so it is also

highly transparent. Previously blockchain (chain of blocks) is being used for registering the transactions or enables moving crypto currencies or digital coins from one individual to another individual. It is a continuously growing list of records and is decentralized, means which is completely open to anyone. And no single node owns the data that is present in a Blockchain and the logic of a Blockchain makes it infeasible to modify or cheat data once it's published on a public Blockchain [8]. Blockchain firstly introduced in Bitcoin, a crypto-currency known as digital currency which moves over the internet from one individual to another. Private key is generally used to transfer the digital coins and if it is lost is impossible to retrieve and hence all the crypto' s are then to be lost. We will discuss later in this report how to re-generate the private key. As per a recent review, it has been observed that currently blockchain is used to handle the ledger for a $10 billion-dollar currency [9].
The working of blockchain depends on the nodes that are connected to this network and some of the preliminaries that are core techniques of blockchain or hyperledger. Blockchain keeps track of all the transactions that are recorded from the start. All the nodes which in this case computers are connected to this network keeps the track of all the transactions and together they approve and reject the new transaction data coming into the blockchain. No data can be altered or modified in the blockchain, once the data is recorded to the ledger, it can not be deleted or modified. This is the main reason blockchain differ from the traditional databases.

As shown in the figure 1.1, all the nodes connected to a blockchain and forms a peer to peer network.

1.2.2 Distributed Ledger

The data in the Blockchain is stored in a distributed ledger, which means it is distributed among all the nodes or users that are in different areas, and removing a intermediate authority to keep track that the system is under control. The information or data in the distributed ledger stored using cryptography and only accessible using keys.
There are two different types of distributed ledgers. One is permission-less and other one is permissioned blockchain.In permissionless blockchain, no permission for accessing and using is required to be a part of the existing Blockchain network and to contribute in it. As stated about permissionless in many areas that anyone and anything can become part of a permissionless blockchain.
On the other hand, permissioned blockchains is a complete opposite of permissionless blockchain. As blockchains idea is to open the network to everyone, but a permissioned blockchain is effectively the opposite. Permissioned blockchains are also know as the

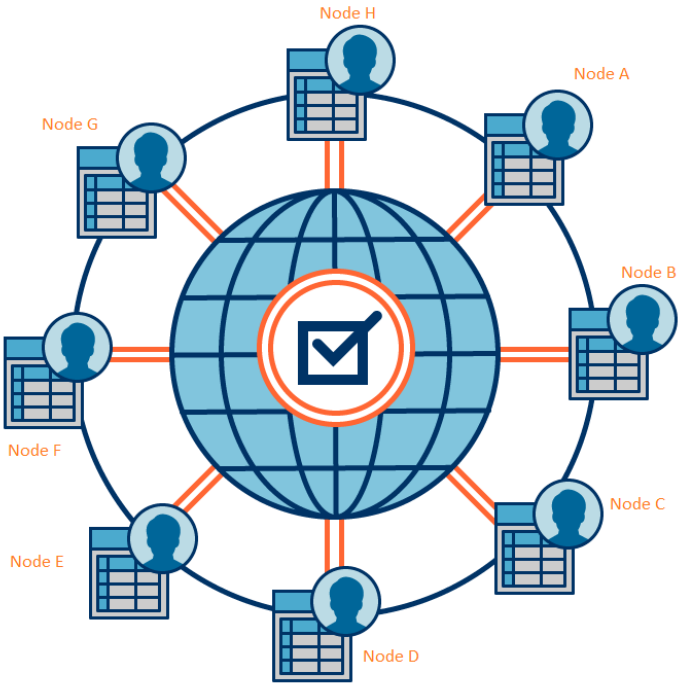

Figure 1.1: Sample Blockchain

private blockchains.In permissioned blockchains we need a permission to join or access a blockchain network [10]. As a result, the owner of a permissioned blockchain has the right to decide who should and should not uses of become the part of its network. The owner can also control the network’ s structure, and to issue the software updates. Generally, the owners control and take care of everything that takes place on their Blockchain [11].

1.2.3 Blocks in Blockchain

Blocks are records or data, which together forms a blockchain [12]. As shown in the figure 1.2, each block typically consists of a cryptographic hash of the previous block, a timestamp and the data/ information. The first node or block in this network is known as genesis block. By design, as described earlier a blockchain is restricted to modification or tempering of the data. It is open to everyone, distributed ledger that can register transactions between multiple parties efficiently, in a verifiable and permanent way means once the information is registered, it can not be deleted. It lives with the Blockchain system.

Blockchains are the perfect example of a distributed computing system. This technique makes distributed ledgers potentially suitable for recording or storing of events,

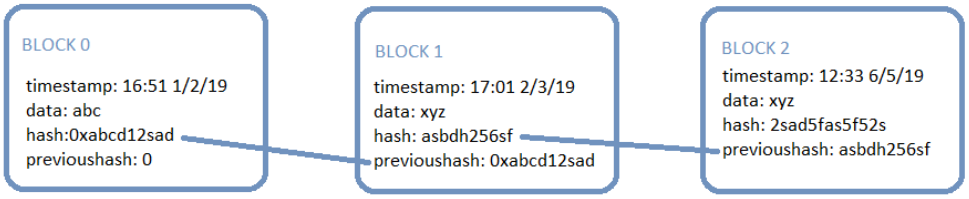

Figure 1.2: Chain of Blocks

medical records, and other records management activities, such as identity management, transaction processing, documenting provenance, food traceability or voting.

1.3 Spatio-Temporal Fingerprints

Definition: 1.3 *"'Sptio-temporal data belongs to both space and time or to space–time."'*

Spatio-temporal is the space data with respect to time. It shows the geometry changes over time. As for tracking of moving objects, which remains at a single position within the given time or a database of wireless communication systems, which may exist only for a short time-span within a geographic region. Spatio-temporal fingerprints are generally used to identify some object in some geo-spatial region within some time-window.

1.4 Motivation and Problem Description

Currently, the user identification over looking at the previously stored data is not very reliable. As the data can be tampered and cheated, so we can't guarantee the identification of the concerned user. The other main reason of this data (as this is in the form of data stream) requires a large storage space. Also typical user credentials are also not enough as it can still be cheated by using others phone's and email's. To overcome these issue and provide a successful identification of user we are using the blockchain technology. Once the data, as in our case GPS locations and the time-stamp is stored in the Blockchain it can not be tampered or cheated. And no centralized authority can hold it. We will identify a user by considering its spatio-temporal fingerprints, which is latitude, longitude and the timestamp. All this data will be coming from user's smart phone and is recorded to the Blockchain. The user can be identified by provided the previous history of visits with the time and then the system will identify the respective user by looking over the stored data in Blockchain. The data stored in Blockchain will also be hidden from the user and the system just provides a simple interface which can only be used for some queries.

1.5 Objectives and Research Contribution

Considering that smartphones are rapidly involves into multipurpose devices that can access and can be used for a wide range of services, there is a general issue about how the positioning information is stored, managed and processed. Examples includes:

a) Information on the specific locations of individuals/users at some defined times.

b) Moving patterns of individuals (specific paths or routes at given time windows and their frequency).

c) Personal points of interest (POI) (frequent visits to specific shops, clubs, or institutions).

In this work, we present an implicit mobile user identification approach as a prototype system. The system can continuously and efficiently verify a mobile user using their trajectory data (spatio-temporal) within some given time-window. The system provides the advantage as the facts that a mobile processing device is personal and there are many low-cost user identification capable sensors are present in today' s smartphone systems for some other tasks and functions.
We assume the processing of spatio-temporal data in a Blockchain and propose the system to find the optimal window which guarantees the verification of some individual. The idea is to find out to prove this formally that how much time-window with certain number of location readings is enough to guarantee for the successful identification of the person. And how much locations user/person need to address to the system. We consider a record that is registered in a Blockchain system also known as a transaction in terms of Blockchain has a typical the following attributes: timestamp, longitude, latitude, and hashed key which is an account identifier. Thus, we ensures the Blockchain for efficient user identification anonymously.

The proposed system ensures the efficiency and a reliable identification of user without the involvement of some intermediate medium which are considered to be as the trusted third parties. Our contributions are as follows:

- We gather the GPS mobile information and store this information in the form of spatio-temporal fingerprints and then being conveniently embedded in our proposed blockchain system (which is basically the history of the specific location of individuals at different times windows).

Notation	Meaning
ℓ	Recorded location
r	Radius for a specific range
t	Transaction timestamp
t1	Start interval time; first timestamp
t2	End interval time; last timestamp

Table 1.1: Table with its useful notations

- We propose a novel user verification system to capture the spatio-temporal patterns of user visits. It considers not only the spatial historical trajectory, but also the temporal periodic patterns. For example, "'listing of all the objects at the location 'ℓ' at time 't', or list all objects that moved in a radius 'r' for location 'ℓ' at time interval ['t1'; 't2']"'.
- We can ensure the identification by querying the user multiple time's to ask the history of different visits in different times
- Our proposed prototype system will find the similarity between the user provided data and stored information in the Blockchain.
- Our main goal in this research is to find that optimal window which successfully do the identification of some individual.
- Furthermore, after the successful identification of person using their spatio-temporal fingerprint we will generate a new private key in the event of a loss of private key of that respective person.

1.6 Thesis Organization

This thesis is about the successful identification of the user using a Blockchain based system which records the user mobility data as a spatio-temporal fingerprint. The rest of the this dissertation is organized as follows:
chapter 2 presents a brief description of the previously done related work, in which some of the companies main Blockchain products are also described.
chapter 3 provides the system architecture overview and the Hyperledger Composer framework.
In the next chapter 4, the detailed methodology is described which consists of the environment setup, processing logic flow and the algorithms created to work for our system.

Furthermore in chapter 5, some of the experiments taken to proof our work and we showed the results of our system. And at the end we have defined the conclusion of our work which is followed by some of the future work.

Chapter 2

Literature Review

Modelling the idea when talking about the spatio-temporal work owes very much to scientists researching the physics of space and time in the start of the twentieth century, And huge volume of data that consists of both space and time are being collected by the smartphones or devices using some of the location based API’ s or services. However, some device may not provide that much accuracy which requires to identify some individuals with high rate of accuracy, and then this does not report the correct location and this is due to the privacy concerns. The validity of location is programed by the techniques that are very similar to the location proofs. [13]. A number of studies [13, 14] have taken in to consideration of secure privacy-aware location validity. The literature study can broadly be distinguished into that are based on infrastructure and that are independent from infrastructure location-proof studies. Infrastructure-based location history verification systems works as the presence of trusted access points or sensors, traditionally the Wi-Fi points, or some other short-range communication medium. The location recorded by such mechanisms is although correct, but it has also some of the limitations. For example, access points or sensors are hard to scale because of the limited coverage. The study in [13], considers wireless proofing using the spatial and temporal properties of wireless systems whereas SecureRun utilizes Wi-Fi points to get the proof of location PoL.

In another work, Zheng et al.[5] proposed a supervised learning techniques to detect the person’s motion sequences from their historical GPS data points. The authors modeled various individual’ s location trajectories to mine the interesting locations and travel sequences with GPS logs. Gao and Huiji in [15] introduced social networks to predict a people’s next location by the recent location of their closest friend, which did not relies on the user’ s own location history. They proposed a social-historical system to study about

the social-historical patterns of check-in behavior for location predictions.
The research in spatio-temporal data has always been the interest of many people and so Luca and James in [16] proposed a series of spatio-temporal methods for user identification by considering of GPS mobility data. The main focus of their work is to identify or track the users from their movement patterns. They uses three popular datasets for experimenting the uniqueness of GPS information. Moreover they provided a detailed analysis of the unfairof power of speed, the direction and total distance of movement. De Montjoye et al [17] also presented the work, where the researchers are able to identify users from a lesser subset of their location records taken from mobile phone service sensors. They mentioned that, in data set where the location of some user or person is registering hourly or after some time interval and with a special resolution that is similar to that given of information of sensors, "'four spatio-temporal points are enough to uniquely identify 95% of the individuals."' They changed the data spatially and temporally to find a way for the uniqueness of human mobility data points given their resolution and the available from the outside information. This way of doing cocludes that the uniqueness of mobility traces decays is approximately as the "'1/10"' power of their resolution.

In another existing and relate-able work to our research, Feng Tian a researcher from Austria [18], developed a technique of traceability in supply chain domain. He mentioned in his work that previously all of the systems were centralized which are asymmetric and are not transparent that could result in the trust issues, like as fraud, cheating, tampering the data and cheating with the information. And the other motivation to his work is the single point of failure of such systems. So he developed a blockchain based solution in which he created a food supply chain tracking system that lies on "'HACCP (Hazard Analysis and Critical Control Points)"' and IoT, which provides an information for all the supply-chain stakeholders with clear visibility, transparency, reliability and security. Also he introduced a new technology called "'BigchainDB"' to fill the gap in the de-centralized systems and to query some large scale data.

2.1 Introduction to Proof of Location

FOAM, a company working on the blockchain technology happens to building the spatial protocols, procedures and applications that bring the geospatial data to distributed ledgers (Blockchains). Recently they posted a blog 'The FOAM Proof of Location' [19], in which they described their work about allowing the users and autonomous agents to secretly record authenticated location data at the time of their choosing, and then afterwards

they reveal their personal information as per their interest, by presenting a fraud-proof of location claim. They introduced a solution that maintains the Byzantine consensus throughout a distributed network with the need to sync of clocks.

2.2 Crypto-Spatial Coordinate (CSC)

FOAM CTO, Kristoffer Josefsson blog post [20] described that CSC allows any smart contract to make an irreversible claim to an address (data) stored in the blockchain and a related location on the map. The blog post described, that CSC are Ethereum smart contracts with related addresses that are positioning in a physical space that can be verified from the chain. This methods is actually allow for physical addresses that are in running environment to have the corresponding smart contract address that can be used for distributed ledgers. They have introduced a method that is based on the geohash technique as a base for their development because of its conceptual simplicity. Another use of the geohash technique is that it is used as the public domains. This not means that the geohash do not have the limitations. This also allows for changing this in future releases if they find it to be very restricted. Instead of using the word hash, the geohash is not actually a hash in cryptography. This is despitea self-similar, space-registering coordinate system.

The CSC standard can be used by any of the smart contract to make a demand to, or for referening a location in some physical environment. When used across different use cases, the CSC considers smart contract registering activities to take it to a spatial dimension. The CSC acting as an index for spatial locations that works for any kind of transaction inside the blockchain. Since geohashes are naturally forms a hierarchy, it also means that a contract that points towards a building, and a contract reference to sensor based devices placed within that premises, and that builds a spatial relationship.

2.3 The Spatial Index [1]

FOAM, a company based on Blockchain work also introduced the "‘The Spatial Index"’ which is a general purpose Blockchain based explorer for visualization. It considered to be the interface for any decentralized system, that has the need for visualizing the smart contracts on a map which is on the visual explorer. They also provided a detialed information about their work and the technologies that are used behind to achieve their goal.

2.4 Analysis and Limitations

After analyzing the above mentioned works and the researches findings that are very much related to our proposed goal there are some of the things that they have are not previously done. No work has been done using the Blockchain based user identification previously. They have used different techniques to follow the user identification or predication, some do not provide much accuracy due to the privacy of such data and some systems are unable to handle the large data as the this type of data with both the temporal and spatial is highly scalable and needs a lot of storing space. Limitations that they have faced are just as described above the handling of large data as the data is continuously growing as the movement of smart phone’ s. And it is on a single place, only a single authority handles it so if the system shut downs all the information will be lost and impossible to retrieve. And for that we have used Blockchain which is exactly the opposite of these systems/techniques as it is decentralized, many entities have the data and not a single line of data item can be tempered.

Chapter 3

System Architecture

In this section, we will give a detailed overview of the architecture, components, modules and interfaces for making of this prototype system and their working. We have used Hyperledger Composer for the development of this prototype system. Hyperledger Composer uses the Hyperlegder Fabric which runs by the Linux foundation. The architecture for the developed blockchain based application comprises of different modules. The process need to accomplish the final output is defined below.

- **Internet and GPS access for coordinates:** This is the first step of the application when it triggers to record an entry into the blockchain. This step will check for the smartphone's internet connection and the GPS for getting the coordinates of the user current location.
- **Smartphone's Local Storage:** Due to the unavailability of smartphone's internet, we will store the GPS coordinates (lat,log) and the system time (in our case smartphone's date and time) in the local db. And once the internet is available we will sync this data to the distributed ledger using an API call.
- **API:** We have used the REST API to connect the smarphone to the distributed ledger, in our case with the hyperlegder composer. This API core task is to get the GPS location and and send it over to the distributed ledger to store this data. This API is acting as a middle-ware to connect the smartphone with the core hyperlegder architecture.
- **Supported/Execution Runtimes:** Hyperledger Composer currently supporting three pluggable runtimes implementations:

  - Hyperledger Fabric: Which is actually used to store the state in the "‘distributed ledgers"’ (Blockchain systems).
  - Web: This runs inside a web page (browser), and is generally used as a interface (playground). The state happens to store in the browser local database.
  - Embedded: This is the execution which happens inside a "‘Node.js"’ process, and the core task is for unit testing business logic. The state of this is stored in an "‘in-memory key-value"’ store.

- **"‘Connection Profiles:"’** are used all over the Hyperledger Composer to specify how to coordinate with the execution runtime. Some different configuration types for execution runtime like the CP for a Hyperledger Fabric runtime that will hold the "‘TCP/IP"’ addresses and ports for Fabric peers and also holds the cryptographic certificates. These connection setups are also the part of "‘Business Network"’ cards.

- **"‘JavaScript SDK:"’** The Hyperledger Composer "‘JavaScript SDK"’ is a collection of "‘Node.js"’ APIs: that actually helps in creation of the applications for managing and interacting with the created business networks. The APIs are further divided into two npm modules:

  - "‘Composer-client:"’ that is used to submit the records/data to a business network and to perform the CURD operations on assets and participants.
  - "‘Composer-admin:"’ that is used to manage business networks which includes the installation, startup and upgradation.

- **"‘Command Line Interface:"’** The composer command line provides an ease to the developers and administrators to create, deploy and managing the business-network definitions.

- **"‘REST Server:"’** The Hyperledger Composer REST server initially provides an Open REST API for a business network. The REST server that is underlying on the "‘LoopBack"’ make the change to composer model for a business network into an open API definition, and at runtime implements CURD support for assets and participants for allowing the records be registered for processing and retrieving.

  - LoopBack Connector: The Hyperledger Composer LoopBack Connector is generally used by the Composer REST Server, but it is also be used without the Rest Server by integration of tools that support the LoopBack implicitly.

- **"'Playground Web User Interface: [21]"'** is a web user interface to define and test the distributed ledger ("'business network"'). It allows the solutions analysts to quickly import templates and prototypes of business logics that will execute on the Web or Hyperledger Fabric runtime.

The diagram that supports the above defined modules is shown below (figure 3.1):

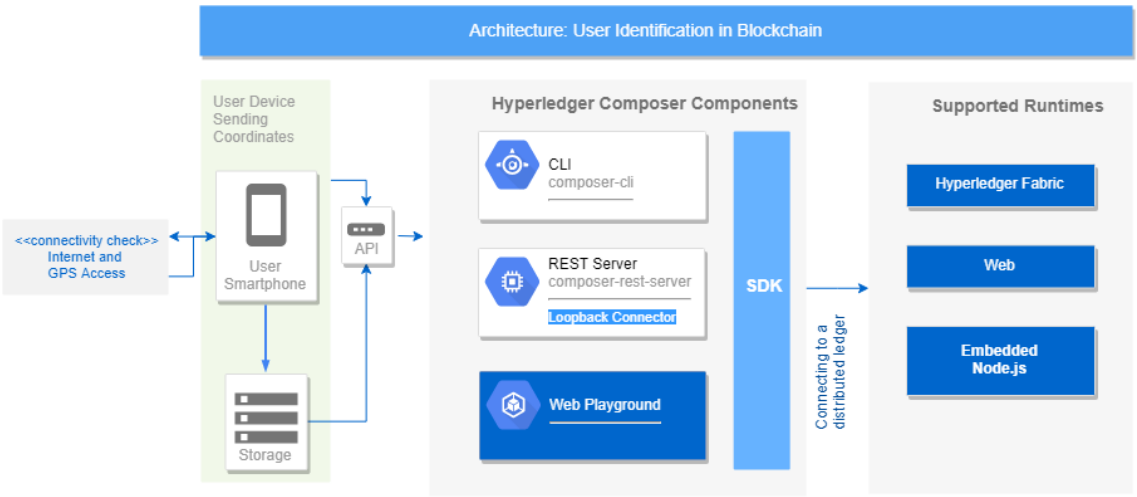

Figure 3.1: Architecture of the System

As our core development to achieve this research outcome is by using Hyperleger Composer, we will discuss in detail the framework of Hyperlegder Composer.

3.1 Hyperledger Composer Framework

Hyperledger Composer [22] is an extensive, open-source development platform developed by the Linux foundation. It is a framework for developing the small usecase blockchain applications. Hyperledger Composer works on above of the current Hyperledger Fabric Blockchain infrastructure and runtime, and this can allow the hyperledger or blockchain consensus protocol to verify that the data recorded is valid and according to the rules defined by the owners or participants.

As shown in figure 3.2, there are different components that lined together to create a simple blockchain application and to deploy it. Each component holds some responsibilities or we can say each got a separate role. We will define each of the module one by one.

Model File (.cto): Model file define all the stakeholders or entities present in the "'business network"'. The major three components in this module are the 'assets', 'participants', 'transactions'. Every component consists of the set of variables and their own data types. We can also create the relationships between each entity.

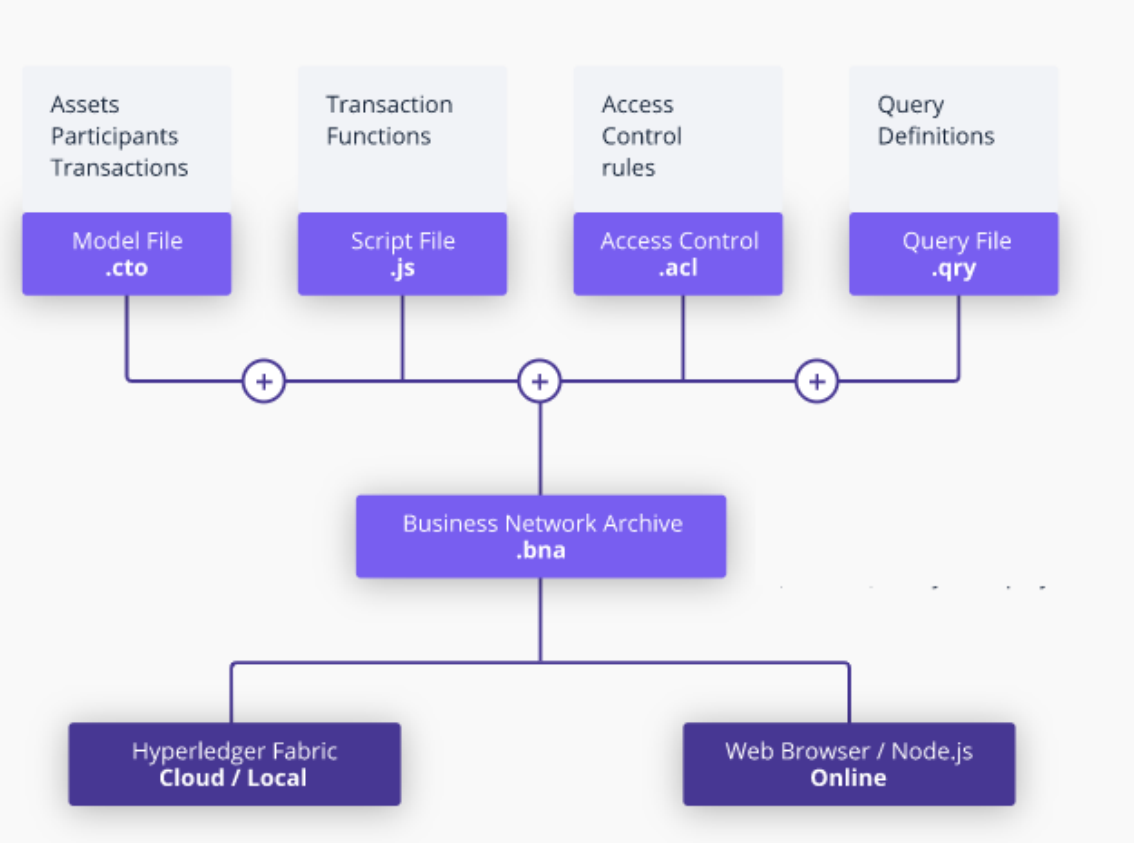

Figure 3.2: Blockchain (Hyperledger Composer) Architecture

Script File (.js): This module handles all the transactions or data recorded in the system and is usally refereed as transaction processor functions. The main logic of the use case is described here as in our case we define the identity management how to compare the data entriers and how to identify a user looking over its previous recorded data.
Access Control File (.acl): The user roles or stakeholder roles are defined in this level. In this file all the conditions should be written and in our case we only registering the data that is coming from the smart phone's GPS.
Query file (.qry): In this file we define the generic queries and this module take care of all the query related stuff such as filtering the results of assets, participants and even transactions. This module work as an SQL but in a litter way.
Business Network Archive (.bna): This module complies up all the above mentioned components and it will be later deployed as the business network in the underlying environment, hyperledger fabric.

As mentioned above about the three main components of Model File, here are some of the details about them:

- **Assests:** Assests are the information about the things that are being registering in

our system. In our case the assests would be the spatio-temporal fingerprints which is in the form of GPS location and the timestamp.

- **Participants:** Participants are directly communicating with the system, as in our case the Individual persons who is using the system for identity verification would be the participant in our blockchain system.
- **Transactions:** Every new entry in the system considered to be a transaction and in our case whenever a new location or spatio-temporal fingerprint reading in recorded in the system it is considered to be a transaction.

3.2 High Level Design

This section describes in further the detailed elements discussed in the architecture. Figure 3.3 shows the the client side and blockchain side components. In this section we have divided the client side components and blockchain side components to give the detailed overview of the working for a blockchain based system.

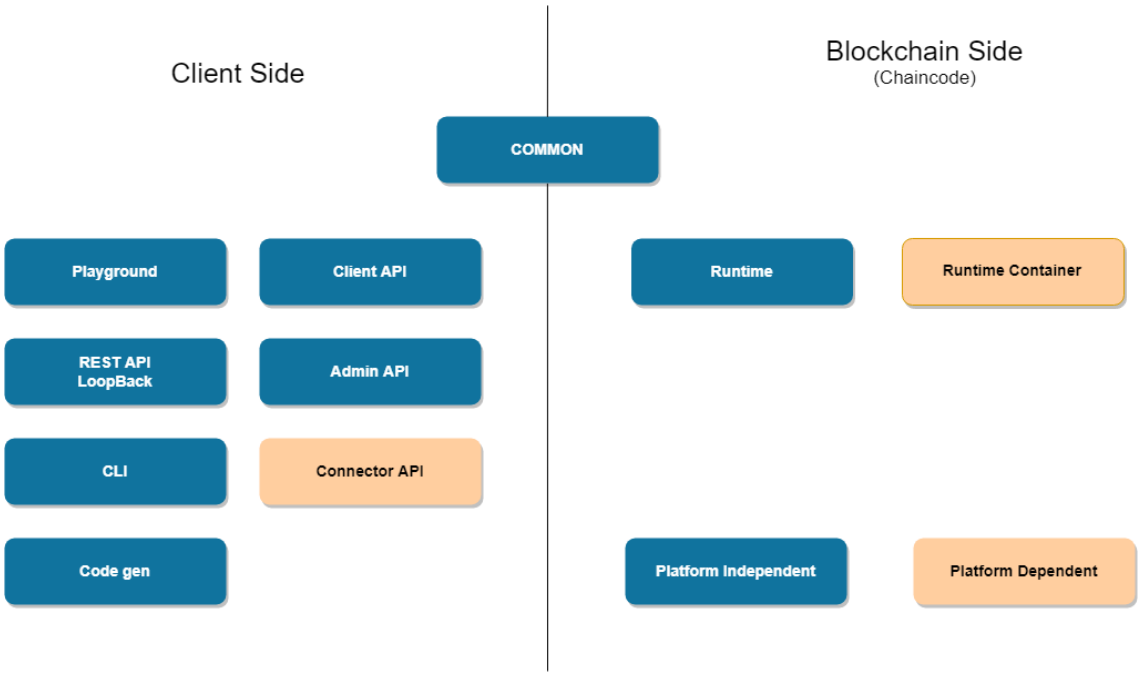

Figure 3.3: Hyperledger Composer Interface

3.2.1 Client Side Components

The majority of the components are client side components, and provide functionality for developing solutions with hyperledger composer.

- **Playground:** Composer-playground is used for developing, configuration and testing business networks in a browser. It provides a web based interface to interect with the hyperledger composer network and to test the scripts of blockchain. We can see the blocks and the block details using this playground and also can access specific block. Below is the image 3.4 that supports the above description of Hyperlegder Composer Playground.

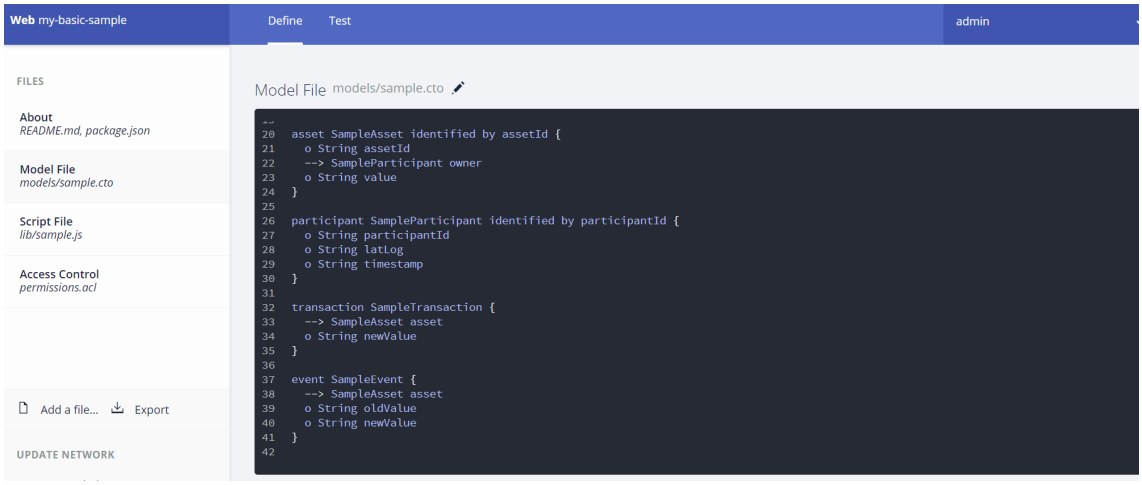

Figure 3.4: Hyperledger Composer Playgorund (Web Interface)

- **REST API generation using LoopBack:** Loopback connector composer is used for building REST APIs for a business network. LoopBack [7] is a framework for exposing backend systems such as databases via REST API. This connector exposes a deployed business network to LoopBack so it can generate a REST API for the assets, participants, and transactions in that business network.

- **Hyperledger Composer-CLI** Hyperledger Composer-CLI is used for the automation and scripting, it provides a CLI application composer that exposes functionality in the composer-admin and composer-client modules to scripting languages, through this we are able to do the following:

  - Deploy, update and upgrade the business network.
  - CURD operations for assests, participants and registries

  - Submit transactions
  - Issue and revoke identities

- **Client API:** Composer client is used for building client applications. Provides APIs for working with a deployed business network and performs the client side data validation before serializing requests and sending them to the runtime running on the target Blockchain platform.

- **Admin API:** Composer-admin is used for building administrative or operational applications. This provides APIs for managing business networks, the first time deployment to the Blockchain platform (deploying the chaincode/smart contract), updating of a deployed business network definition. Also used for undeploying a deployed business network.

- **Editor Plugins:** Provides ease to user to use their interest of editors, we have used VSCode for this system.

3.2.2 Blockchain Side Components

The rest of the components are Blockchain components and provide functionality for running a deployed business network:

- **Composer Runtime:** Composer runtime provides the following features:
  - Management of the deployed business network.
  - Persistence of resources (assets, participants, transactions) into registries.
  - Access Control enforcement.
  - Execution of user developed transaction processor functions.

- **Composer Runtime Container:** This is the chaincode/smart contract implementation, that provides a platform specific set of services to the platform independent runtime code:
  - Loading and execution of JS core runtime
  - Routing of API calls from client into core runtime.
  - Logging
  - Data persistent using the world state
  - Identifying the participant/ certificate used to submit the transaction.

3.2.3 Composer Common Module

Composer common module performs the following tasks:

- Logging APIs used by the rest of Hyperledger Composer.
- Parsing and validating the APIs for the parts of a business network definition – model files, access control lists and transaction processor functions.
- APIs for creating and loading business network definitions/archives.
- Connection profile manager API, for describing connections to Blockchain platforms.
- Connector API, for building connectors which connect to Blockchain platforms.
- Code generation for JSON Schema and Loopback models

3.3 Loopback Connector Composer

Loopback [23] is a framework for exposing for backend systems such as databases via Rest API. The hyperledger composer loopback connector exposes a deployed business network to loopback so it can generate a REST API for the 'assets', 'participants' and 'transactions' in that "'business network"'.
The composer-rest-server module provides an easy to use CLI application for users who don't need to understand Loopback to create a REST API.

Chapter 4

Methodology

In this chapter, we will give a detailed overview of the methodology used to achieve this research outcome. We described the implementation steps and techniques here that is used to gain our research objective. First we will describe the pre-requisites to install for setup a blockchain system and then we will list down the commands, algorithms and some of the sample block designs.

4.1 Environment Setup

Some of the main pre-requisites that we need to setup the Blockchain system are mentioned below with the detailed version and flavor description, all the below mentioned languages and tools are need to be install to up-running a complete Blockchain system:

- **Operating Systems: Ubuntu Linux 16.04 LTS (64-bit):** To use Linux, we have used the windows Linux subsystem. We just need to activate the windows Linux system and we can will be able to use Linux (Ubuntu) in our current operating system.
- **Docker Engine - Version 17.03 or plus :** Docker provides the developers an environment to develop and run applications. Docker quickly gather all the components and gets the code tested and deployed in the production.
- **Docker-Compose - Version 1.8 or higher:** Compose is a tool for defining and running multi-container Docker applications.
- **Node: 8.9 or higher:** Node is an open-source, JS run-time environment that executes the code written in JS outside the browser.

- **npm: v5.0 or greater:** npm is a package-manager for the JavaScript programming language.
- **git: 2.9.0 or plus:** Git is a distributed technology mainly used for version-control for track down the changes happens in the source code during the development stages.
- **Python: v2.7 or greater:** Python is a high-level programming language.
- **VS Code:** Visual Studio Code is a Microsoft code editor.

4.2 Implementation Details

We implement a prototype identification for users to know retrieve their identity. The implementation is done using the Hyperledger Composer which uses the Hyperledger Fabric as its underlying technology. Hyperledger Composer provides an easy to use interface (playground) to develop and test the Blockchain applications. It is an open-source development toolset and a framework currently powered by the Linux foundation. Composer allow us to define the data models and to implement the business logics. Currently, composer only supports the JavaScript as the main programming language to develop. We additionally need the Docker to set-up this Blockchain network, as we are creating the distributed application locally. Docker makes easy to run and test the application on the local environment. Below are the steps (pre-requisites) that we need to follow and to install for to up and running this Blockchain environment (Hyperlegder Composer) [24]:

- After logging into the terminal window of WSL, we need to install the commands mentioned in 4.1. The commands mentioned below are used to clone the environment of Composer and to install the pre-requisites for the Composer in the Linux environment.

→ *curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh*
→ *chmod u+x prereqs-ubuntu.sh*
→ *./prereqs-ubuntu.sh*

Table 4.1: Commands to setup Hyperledger Composer

- After installing some of the pre-requiesties of Hyperlegder Composer, we need to install an extension of Hyperlegder composer on VS Code.

- Once the extension is install, we need to some more installations using the same WSL (Ubuntu) terminal window. Some of the CLI tools to provide the essential operations like running a REST server, creating application assets and applications are mentioned below in 4.2.

→ *sudo npm install –unsafe-perm –verbose -g composer-cli@0.20*
→ *sudo npm install –unsafe-perm –verbose -g composer-rest-server@0.20*
→ *sudo npm install –unsafe-perm –verbose -g generator-hyperledger-composer@0.20*
→ *sudo npm install –unsafe-perm –verbose -g yo*

Table 4.2: CLI Tools to Install

- After the installation of CLI tools, we need a web interface to interact with the transactions and assets. Composer provides a web playground UI and to install this we need the following command to install the playground.

→ *sudo npm install –unsafe-perm –verbose -g composer-playground@0.20*

Table 4.3: Composer Playground UI Install

- Then, we need to install the Hyperledger Fabric to give us the local runtime for deployment of our business network. The commands used to install Fabric are provided in the below box 4.4.

→ *mkdir /fabric-dev-servers*
→ *curl -O https://raw.githubusercontent.com/hyperledger/composer-tools/master/packages/fabric-dev-servers.tar.gz tar*
→ *cd /fabric-dev-servers*
→ *./downloadFabric.sh*

Table 4.4: Hyperledger Fabric Installation

- At the end we need to generate a Peer Admin Card and start the web app.

All of the above installation guide is followed using this [25].

4.3 Processing Logic Flow

The following sequence of steps shows the system processing logic flow:

→ *./startFabric.sh*
→ *./createPeerAdminCard.sh*
→ *composer-playground*

Table 4.5: Starting Application

- When the Blockchain system up and running, we have to register/record the GPS coordinates into the composer assets. This considered to be a transaction. These records are coming from the user smartphone device. Whenever a new record as a transaction submitted to the Blockchain, the system will also enter the timestamp to the transaction block.
- All the transactions of one single object are recorded against some user or smartphone ID. This is used to retrieve all of the blocks related to some specific ID when needs to identify.
- Once the transactions are registered in the system, users can anytime from anywhere can locate or identify themselves by using or query the system. The system can provide a interface to provide or query information to match the similarity between the feeded record in the system and provided record to the system.
- System can ask for the location and time of the user, when user wants to identify themselves. User after providing the location and time to system will be able to identify or verify. System ensures the identification after checking of location at the set timestamp.

4.4 Algorithms

Blockchain applications can be best describes in terms of the Assets, Participants and the Transactions, that are stored inside a network.

- **Assets:** Assets could be anything of value that can be registered or shared over the network. In our case of user identification system the assets are the spatio-temporal fingerprints which comprises of the GPS coordinates and the timestamps. We here defined the data-structure comprises of the following information. The below mentioned data structure needs to be build in the .cto file of composer.
  - **ownerId:** Unique ID of a user. We can identify the key values (fingerprints) of a user by using this identifier.
  - **participantId:** This is the block ID, identifier for the some specific block.

Data Structure
-> String ownerId
-> String participantId
-> String lat
-> String log
-> DateTime timestamp

Table 4.6: Block Definition

  - **lat:** Stores the latitude coordinate.
  - **log:** Stores the longitude coordinate.
  - **timestamp:** Binds the time and date with the block when ever a transaction happens.

- **Participants:** are considered to be the actual stakeholders of the system. These are the real actors which needs to store the information in the network.
- **Transactions:** This describes the actions that are performed on the asset by the participant. Every new block entry considered to be a new transaction in the system.

This identification system key roles are to register the spatio-temporal fingerprints and provides a decision every time a user wants to identify the system. The main algorithms used to fulfill these scenarios are mentioned below.

Algorithm 1 Fingerprint Creation

Data: Owner Id, Fingerprint Description
Result: Creates a Fingerprint with appropriate values in Identification System
if *fingerprint exists* **then**
  return
else
  set the fingerprint attributes with the description
  push the fingerprint to Blockchain
  push the current time to Blockchain
end

Fingerprint creation algorithm takes two input for storing in the blockchain system, one is the OwnerID and the other is fingerprint description, which is in our case is the GPS locations with concatenation of the timestamp. If the same information is present already in the system ,the new information will not be stored and if the information against some ID is new. It will be stored in the system.

Algorithm 2 User Identification

Data: Location Information, Time and Date
Result: Returns a result (Successful or Unsuccessful)
if *location not exists in time* **then**
  return
else
  retrieve all locations against the owner ID
   check the time against the locations with the input time
   if *location and time matches* **then**
    return successful
     push the result to logs
  else
    return unsuccessful
     push the result to logs
  end
end

The algorithm 2 provides the basic logic flow of this identification system. It takes location information and date-time as an input and after matching the queried window with the Blockchain system, it returns the blocks or inputs as successful or unsuccessful.

Algorithm 3 Displays User Information History

Data: Owner Id, Fingerprint Description
Result: Returns a all loaction history with timestamp
if *user history exists* **then**
  return user location and history within the query time
else
  return
end

Algorithm 3 displays the user visited location information with the time of visits. This algorithm takes OwnerID, and timestamp as as input and returns the user visit history from Blockchain. Algorithm also checking if the ownerID or users exists in the Blockchain system or not.

Chapter 5

Experiments and Results

In this section, we will show some of the outcomes of our research work. First we will show the basic blocks of this identification and then we will accumulate our test results basis on some test cases. We have registered some blocks against a user ID, which is the owner ID in terms of the Blockchain data structure. Every time a block enters in the network it is considered as a transaction in the system. Some of the blocks against some owner ID is shown in the below figure 5.1.

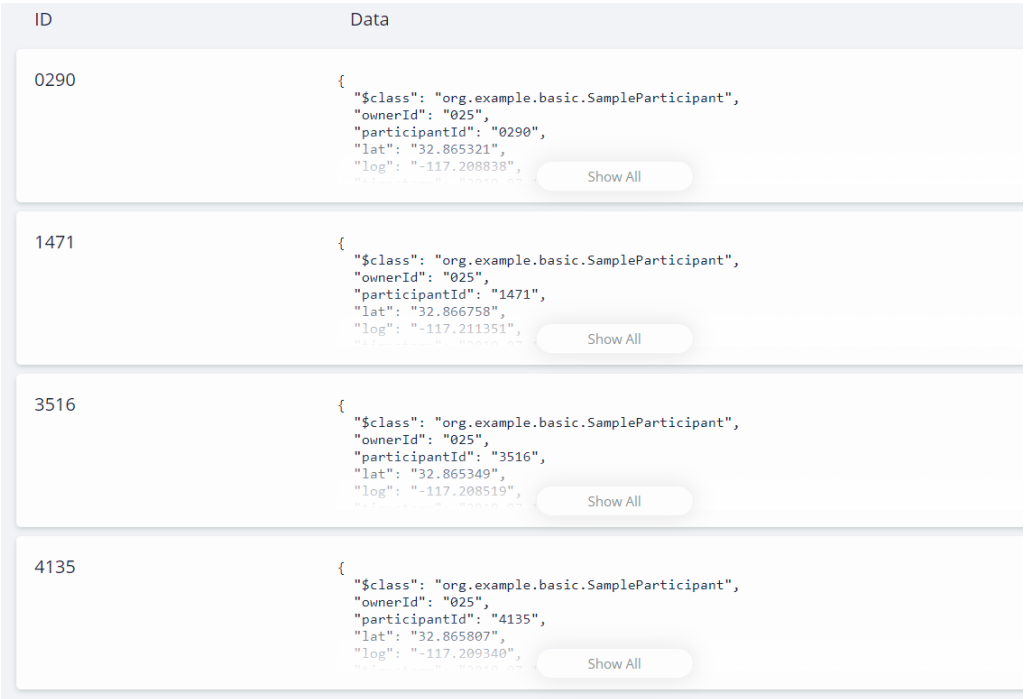

Figure 5.1: Blocks against Owner ID: 025

Each block shows the complete description of the block and the fingerprint of a user. The figure 5.2 shows the complete block in our user identification system.

Figure 5.2: Block 0290 against Owner ID: 025

As a test case we have provided multiple inputs to the system to check the retrieval rate of the system. System successfully matched the coordinates registered in the system and the coordinates provided to the system. As a prototype system we have to enter the exact coordinates to be matched to give the output to the user. The matching engine retrieves the output in our case blocks within no second. The input provided to the system should be accurate enough to match the coordinates and with in the time provided.

Analysis As for the experiments and collecting results to show the accuracy and the efficiency of our prototype system, we have collected the results as shown in the below table 5.1:

OwnerID	Query No.	Fingerprints	Query Time	System Performance (CPU)
025	1	100+	<2 sec	80%
025	2	100+	<2 sec	79%
025	3	100+	<2 sec	84%
026	1	20	<1 sec	50%
026	2	20	<1 sec	49%
026	3	20	<1 sec	23%
027	1	10	<1 sec	26%
027	2	10	<1 sec	27%
027	3	10	<1 sec	27%

Table 5.1: Test runs and Results

We have recorded the information of three smart phone’ s GPS coordinates against their respective ID’s. And we have collected the GPS locations that are recorded after the every change in the coordinates. The query time and the system performance can also be seen in the table. The total number of records are also shown in the table. We have also calculated the end result of the system by providing the coordinates to the system that can be seen in table 5.2.

System queried the user to enter the details of the coordinates and the timestamp and after matching or precessing the provided information with the system’s information, it provides the end results as shown in the table 5.2.

Query No.	Processing Time (Seconds)	Result
1	<3 sec	Successful
2	<5 sec	Unsuccessful
3	<5 sec	Successful

Table 5.2: End Results

Chapter 6

Conclusions & Perspectives

Blockchain technology by design provides many features as the integrity, transparency, authenticity, and audit-ability and it is also provides larger security as comparable to other storing systems. And thus this makes it possible the greatest choice for maintaining and identifying a the user due to the large data set that is in the form of GPS readings with the its timestamp. Blockchain provides lesser conflicts through increased trust and distributed environment. This research work User Identification using spatio temporal fingerprints is a Blockchain based technique to successful identify a user by using its smart phones location readings. We provided the prototype of user identification model based on Hyperledger Composer and evaluated its performance. The experimented system registers the readings of mobile GPS coordinates and stores them in the Blockchain system as a fingerprint and then system successfully able to identify the user by querying the location and time. The problem that we have faced in during the development is to run the environment setup in a local machine. It is very complex to setup in a local machine and takes a lot of time to initiate the network. Hyperlegder composer provides much ease to test and run the application using it's interface that is Playground. The prototype has shown acceptable overhead in terms throughput and resource utilization with the scope of optimization for full-scale end to end application. We have run multiple system performance tests and calculated the end results to achieve this research goal.

6.1 Potential Future Directions

There is always a room for improvement and therefore, a few possible future directions for enhancement in this work are mentioned below. The technique that can be useful in future and and performance of this prototype system can be enhanced by:

- Designing a more accurate system, this can be done by using a lighter tool and hyperlegder so the query time required to process a single verification will take less than the current.
- Less querying data for a system to successfully identify a user using spatio-temporal data and provide more accuracy with respect to GPS data collection.
- Ensuring the systems accuracy by asking the multiple time-windows of user visits in different days. And generating a private key after using some algorithm to retrieve all of the lost digital currency.

Bibliography

[1] Ryan John King. The Spatial Index (A general purpose visual blockchain explorer). https://blog.foam.space/the-spatial-index-9793f42c46c8, 2018. [Online; accessed 05-June-2018]. Cited on pp. iii and 11.

[2] Paolo Compieta, Sergio Di Martino, Michela Bertolotto, Filomena Ferrucci, and Tahar Kechadi. Exploratory spatio-temporal data mining and visualization. *Journal of Visual Languages & Computing*, 18(3):255–279, 2007. Cited on p. 1.

[3] Martin Erwig, Ralf Hartmut Gu, Markus Schneider, Michalis Vazirgiannis, et al. Spatio-temporal data types: An approach to modeling and querying moving objects in databases. *GeoInformatica*, 3(3):269–296, 1999. Cited on p. 1.

[4] Claudio Bettini, X Sean Wang, and Sushil Jajodia. Protecting privacy against location-based personal identification. In *Workshop on Secure Data Management*, pages 185–199. Springer, 2005. Cited on p. 2.

[5] Huiji Gao, Jiliang Tang, and Huan Liu. Mobile location prediction in spatio-temporal context. In *Nokia mobile data challenge workshop*, volume 41, pages 1–4, 2012. Cited on pp. 2 and 9.

[6] Ben Cresitello-Dittmar. Application of the blockchain for authentication and verification of identity. *Scientific reports*, 1:1–9, 2016. Cited on p. 2.

[7] Cuneyt Gurcan Akcora, Yulia R Gel, and Murat Kantarcioglu. Blockchain: A graph primer. *arXiv preprint arXiv:1708.08749*, 2017. Cited on pp. 2 and 18.

[8] Vijay Kandy and Shane Loomb De Montjoye. Decibellive: A decentralized noise pollution monitoring and incentive platform. *Scientific reports*, 3:1–8, 2017. Cited on p. 3.

[9] Jeff Herbert and Alan Litchfield. A novel method for decentralised peer-to-peer software license validation using cryptocurrency blockchain technology. In *Proceedings*

of the 38th Australasian Computer Science Conference (ACSC 2015), volume 27, page 30, 2015. Cited on p. 3.

[10] Thomas Hardjono, Ned Smith, and Alex Sandy Pentland. Anonymous identities for permissioned blockchains. 2016. Cited on p. 4.

[11] Vincent Gramoli. On the danger of private blockchains. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers (DCCL16)*, 2016. Cited on p. 4.

[12] Bastian Fredriksson. A distributed public key infrastructure for the web backed by a blockchain, 2017. Cited on p. 4.

[13] Chitra Javali, Girish Revadigar, Kasper B Rasmussen, Wen Hu, and Sanjay Jha. I am alice, i was in wonderland: secure location proof generation and verification protocol. In *2016 IEEE 41st conference on local computer networks (LCN)*, pages 477–485. IEEE, 2016. Cited on p. 9.

[14] Xinlei Wang, Amit Pande, Jindan Zhu, and Prasant Mohapatra. Stamp: enabling privacy-preserving location proofs for mobile users. *IEEE/ACM transactions on networking*, 24(6):3276–3289, 2016. Cited on p. 9.

[15] Huiji Gao, Jiliang Tang, and Huan Liu. Exploring social-historical ties on location-based social networks. In *Sixth International AAAI Conference on Weblogs and Social Media*, 2012. Cited on p. 9.

[16] Luca Rossi, James Walker, and Mirco Musolesi. Spatio-temporal techniques for user identification by means of gps mobility data. *EPJ Data Science*, 4(1):11, 2015. Cited on p. 10.

[17] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376, 2013. Cited on p. 10.

[18] Feng Tian. A supply chain traceability system for food safety based on haccp, blockchain & internet of things. In *2017 International Conference on Service Systems and Service Management*, pages 1–6. IEEE, 2017. Cited on p. 10.

[19] Ryan John King. "Introduction to Proof of Location". https://blog.foam.space/introduction-to-proof-of-location-6b4c77928022, 2018. [Online; accessed 02-June-2018]. Cited on p. 10.

[20] Kristoffer Josefsson. "Crypto-Spatial Coordinates". https://blog.foam.space/crypto-spatial-coordinates-fe0527816506, 2018. [Online; accessed 05-June-2018]. Cited on p. 11.

[21] Hyperledger Composer. "Typical Hyperledger Composer Solution Architecture (Playground)". https://hyperledger.github.io/composer/v0.19/introduction/solution-architecture, 2019. [Online; accessed 19-Feb-2019]. Cited on p. 15.

[22] Hyperledger Composer. "Build Blockchain applications and business networks your way". https://hyperledger.github.io/composer/latest/, 2019. [Online; accessed 19-Feb-2019]. Cited on p. 15.

[23] IBM company. LoopBack 4: Build Amazing APIs. https://loopback.io/, 2018-2019. [Online; accessed 02-April-2019]. Cited on p. 20.

[24] Hyperlegder Composer. "Installing the development environment". https://hyperledger.github.io/composer/latest/installing/development-tools.html, 2019. [Online; accessed 20-Feb-2019]. Cited on p. 22.

[25] Eddie Kago. "Getting Started with Hyperledger Composer on Windows 10". https://medium.com/kago/tutorial-to-install-hyperledger-composer-on-windows, 2018. [Online; accessed 08-March-2019]. Cited on p. 23.

Muhammad Tehseen Tahir - Thesis

ORIGINALITY REPORT

17%	9%	9%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"Web Information Systems Engineering – WISE 2018", Springer Nature America, Inc, 2018 Publication	2%
2	shyamtechno.blogspot.com Internet Source	2%
3	Submitted to Higher Education Commission Pakistan Student Paper	1%
4	Auqib Hamid Lone, Roohie Naaz Mir. "Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer", Digital Investigation, 2019 Publication	1%
5	"Secure Data Management", Springer Nature, 2005 Publication	1%
6	Submitted to The University of Memphis Student Paper	<1%
7	Submitted to National College of Ireland Student Paper	<1%