Image Integrity Analysis with BlockChain Technology

Bachelor Thesis

Submitted By

Anuj Kr. Pathak
&
Sayan Shankhari

A thesis submitted to

Indian Institute of Information Technology Kalyani

for the partial fulfillment of the degree of

Bachelor of Engineering in Computer Science
in
Department of Computer Science and Information Technology

May, 2019

To my beloved parents and friends who have supported me and prayed for my success throughout my life.

Certificate

This is to certify that the thesis entitled **Image Integrity Analysis with BlockChain Technology** being submitted by undergraduate students **Anuj Kr. Pathak** (Id: 000000102) and **Sayan Shankhari** (Id: 00000121) in the Department of Computer Science and Information Technology, Indian Institute of Information Technology Kalyani, Nodia, 741235, India, for the award of **Bachelors of Technology** in **Computer Science & Engineering**, is an original research work carried by them under my supervision and guidance. The synopsis has fulfilled all the requirements as par the regulation of **IIIT Kalyani** and in my opinion, has reached the standards needed for submission. The works, techniques and the results presented have not been submitted to any other university or Institute for the award of any other degree or diploma.

——————————————-

(**Dr. Imon Mukherjee**)
Assistant Professor
Department of Computer Science and Information Systems
Indian Institute of Information Technology Kalyani
IIIT Kalyani Campus, West Bengal 741235, India. December 2018

Declaration

We hereby declare that the work being presented in this project report entitled, **"Image Integrity Analysis with BlockChain Technology"**, submitted to Indian Institute of Information Technology Kalyani in partial fulfilment for the award of the degree of Bachelor of Technology in Computer Science and Engineering during the period from July, 2018 to May, 2019 under the supervision of Dr. Imon Mukherjee, Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, does not contain any classified information.

Anuj Kumar Pathak
CSE-150__ :: 00000102
Computer Science and Engineering,
Indian Institute of Information
Technology Kalyani,
WEBEL IT Park, West Bengal
741235 India

Sayan Shankhari
IT-15026 :: 00000121
Information Technology,
Indian Institute of Information
Technology Kalyani,
WEBEL IT Park, West Bengal
741235 India

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(**Dr. Imon Mukherjee**)
Assistant Professor
Department of Computer Science and Information Systems
Indian Institute of Information Technology Kalyani
IIIT Kalyani Campus, West Bengal 741235, India
May 2019

Acknowledgments

First of all, We would like to take this opportunity to thank my supervisor Dr. Imon Mukherjee without whose effort this thesis would not have been possible. We are so grateful to him for working tirelessly after us, clearing our doubts whenever and wherever possible. We are most grateful to Department of Computer Science and Information Technology, Indian Institute of Information Technology Kalyani, West Bengal, 741235, India, for providing us this wonderful opportunity to complete our bachelor thesis. We would like to thank our friends for providing us help as and when required. We would like to thank our team mates for being a great motivators and great friends.

And last but the biggest of all, We want to thank our parents, for always believing in us and letting us do what we wanted, but keeping a continuous check that we never wandered off the track from my goal.

Anuj Kumar Pathak
CSE-150__ :: 00000102
Computer Science and Engineering,
Indian Institute of Information
Technology Kalyani,
WEBEL IT Park, West Bengal
741235 India

Sayan Shankhari
IT-15026 :: 00000121
Information Technology,
Indian Institute of Information
Technology Kalyani,
Webel IT Park, West Bengal
741235 India

Contents

List of Figures

Abstract

With the digitalization any historical record are available in the form of collection of bits. Any document can be stored in advanced electronic devices like computers, smart phones file systems as files. The most used or popular files are media files (image, audio, video). For the lack of true information and experience or bad ethics of mind the number of digital scams is getting higher in developing country like India. So the data files need to be protected somehow somewhere and also there should be a system that will veryfy the requested file. So we started with the simplest media format *i.e.* Image. Making a centrral database can be unreliable because the system can be breached and the data might be changed no matter how advanced and protective the protocols are. So we took the concept from advanced decentralized architecture of BlockChain mostly BitCoin. A purely peer-to-peer version of online data would allow online transactions to be sent directly from one node to another without going through a central authority. Digital signatures provide part of the solution. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone. To prevent the action against one of the biggest issues in India of Fake Media (Image, Sound) Scam. This project tries to find a way to prevent it with a new upcoming technology starting with image files.

Keywords: Peer-to-Peer network, Proof-of-work, Proof-of-Stake, Distributed Ledger, Concensus Mechanism, Hash Function, Video Formatting, Image File formats, Watermarking, WebServer.

Chapter 1

Introduction

This chapter presents the introduction of the thesis that includes the brief description of BlockChain, and the adopted approach to address the problems. This chapter also presents the scope of this thesis and the contributions of the thesis.

Image integrity is the task of checking if the image file's bits are changed or not at any point. A digital file might change by platforms. This would be fully online system. Nowadays with increasing people in digital world the editing softwares are getting smart. Most of the files that might be non-editable in some operating systems or file systems, but the so called hackers or masters of electronic devices have file systems that can easily access and change any file data. So in that way the media files can not be checked for integrity. And also while the images are shared in different platforms, the files' data might be changed according to their protocols for data compression or security.

The proposed system in this thesis will help us to detect those shared files and compare them not bit by bit, but context wise and signature wise. That means if one shares an image both our system as well as one of those online platforms and download them as separate file and check in our system, it should return truth values if data portion are same or atleast 95% same. The system we are introducing the social media like platform that have the capability to store images and show them in news feed. There is a option for every photo to download in user's computer or mobile devices and after sharing and getting back the person can check if the file is intact or not in our proposed system by the digital signature.

1.1 Scope of Discussion

This thesis focuses on building a hybrid architechture inspired by one of the major implementations of BlockChain i.e. BitCoin [6]. By the advancement of PHP (Hypertext PreProcessor) and JS (JavaScript) which are the basic building languages or platforms that can be run in any modern devices, it might be quite easy to make a peer-to-peer web-api (Application Programming Interface) running like bitcoin decentraized network as well as social media platform that might be a real Truth Machine. While comparing image files data we will first compress the data using our own protocol and compare by bit matching Euclid, or Deep CNN (Convolutional Neural Network).

1.2 Methodological Approach

The system that we are trying to build is capable of processing, storing and showing image files. Not only that, it also allows viewers to veryfy his own copy of the image. It is not easy to store and veryfy in a moment and it is about impossible to change the whole blockchain, because the miners have their own copy of the public ledger and their processed computation powers in their computers. The automated server should back up its data from both the Virtual miners as well as remote miners' computers. The studied and inspired technologies are discuussed in the thesis.

1.3 Thesis Contribution

The main contributions of this thesis includes

1. Proposes an user anonymity-preserving algorithm to be a part of Video Integrity Program.

2. Formally analyzes the security of the newly designed protocol as well as its performance.

3. The scheme, as compared to the existing schemes, not only authenticates the users but, also establishes a session key between the user and the System after successful mutual authentication.

4. The scheme provides many security and robustness features of user authentication and Block Processing scheme for BCTs.

5. No installation required to be a part of the system, except you want to be the miner.

1.4 Roadmap of the Thesis

The structure of the thesis is as follows:

1. The Chapter 1 is an introductory part which discusses the scope of the thesis, about the contribution of this thesis and the motivation for writing it.

2. The Chapter 2 provides the background of Image integrity security aspects of it and previous works.

3. The Chapter **??** introduces the proposed authentication framework after highlighting the motivations behind this work.

4. The implementation of Chapter 4, where an informal implementation of the proposed protocol has been discussed.

5. The Chapter 5 comprises of the conclusion and further work of the Project in future.

1.5 BlockChain

A BlockChain or **"The Truth Machine"** [3] can be broadly described as a peer-to-peer network of nodes that makes a collaborative effort in sensing certain specified chain of blocks of data around its periphery and thereby controls the surrounding environment. Accrrding to Wikipedia [8] a blockchain, originally block-chain, is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree). In BlockChains, each node consists of processing capability, it may contain multiple types of memory like program, data and memories, having a Web-Service transceiver, Client-side processors, and a power source. The nodes communicate with each other using web-services and self-organized. There are certain nodes called miners that veryfies each transactions or entry of data in the chain and are the most reliable personnel in the network who always have the updated copy of blocks of data.

Some of the underlying concepts of the BlockChain especially BitCoin (the most popular implementation of BlockChain) and some other important technology are briefed.

1.5.1 Peer-to-Peer Network

This is the internet protocol that connects different logged in users as a node which is having some computation power. The users who are only uploading and verifying may have computer or smartphone, but the verifiers or the miners must have to work on computer with sufficient amount of computation power.

1.5.2 Transactions

Everything in crypto-currency comes under transactions, i.e. someone is sending some amount of money to someone else at some time. So a basic or overall transaction data can be structured as,

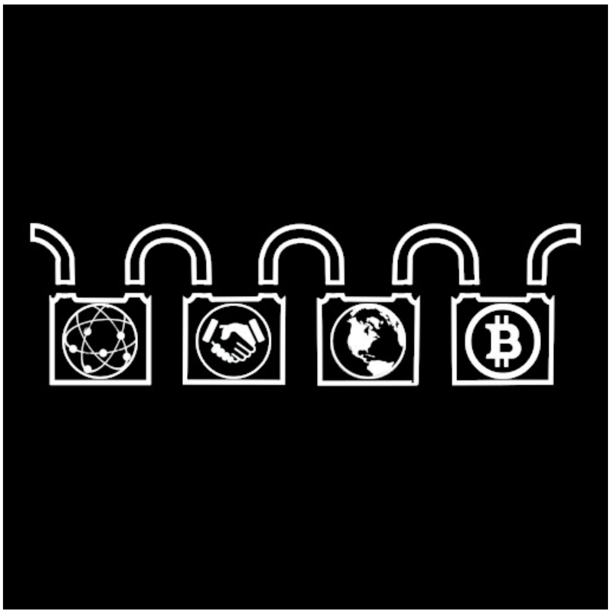

Figure 1.1: Overview of Block-Chain Technology (BCTs).

```
Transaction ::  {
     <Transaction_id>,
     <TimeStamp>,
     <Sender_Id>,
     <Receiver_Id>,
     <Amount_Unit>
}
```

This transaction details is send to every peer to verify. If they heard already about it, it is true, or it is false (same as women's un-manipulated gossip in village). If more than 50% population declare it true, then it is allowed to be in the Public Ledger.

1.5.3 Public Ledger

This is the publicly shared record of transactions kept as a list of blocks. At a particular point of time everybody (every node), who are connected to the network, should have a same copy of ledger in their own devices of what server has. Whenever a new person logs into the network aotomatically the server forces to update the ledger to the person's device. So basically the ledger is the chain file.

1.5.4 Chain of Blocks

It means list of blocks of data having some common part with previous and next block. This is like Single-way linked-list(every node consists of data and program memory address to the next node). Every block contains a number of transactions and many more things.

```
Block ::  {
     <Block_Id>,
     <TimeStamp>,
     <Merkle_Root>,
     <Verifier_Id>,
     <Nonce_Value>,
     <Previous_Hash>,
     <Current_Hash>,
     <Data ::  ANumberOfTransactions>
}
```

So it is kind of backword linked-list which is propagating by having the previous block's kind of identity (because there is a mild chance of collision i.e. multiple values' hashes are same) hash.

1.5.5 Timestamp

The time means absolute global date-time in the format:

```
TimeStamp ::  String ( <day_of_week_code ::  ddd>,
<month_code ::  mmm>, <day_of_month ::  dd>, <year ::
yyyy>, <time ::  hh:mm:ss>, <Distance from Mean-TimeLine
::  GMT+hhmm>, <Time Zone ::  Country_Name Standard Time>)
```

Example: **"Sat May 25 2019 20:45:04 GMT+0530 (India Standard Time)"**. The timestamp is one of the most needed to prove it later for verification of the record. It is used to create the current block's hash.

1.5.6 Hash

The job of a hash algorithm is to map any size of domain to a particular size of range. SHA-256 is one of the most popular hash algorithms which takes any length input and returns 256 bit output. All input/output operations can be transferred into strings.

1.5.7 Merkle Root Hash

Merkle tree is a complete binary tree which has the hash values of transaction data as leaf nodes. The tree propagation occures from leaves to root as tournament tree form. For any point,

parent hash = hash(child-1 hash + child-2 hash);

A little change in any data of any transaction will change the merkle root, and thus the block's hash and the complete chain. Because it is used to create the current block's hash.

1.5.8 Previous Hash

The previous block's hash. It is required to maintain the chain system because we can not create the next address and we don't know when it would be created, so it is better to store what we already have. It is used to create the current block's hash.

1.5.9 Nonce

It is the quantity of computation power used to solve a mathematical problem which is not so hard but not so easy either. Too easy solution will be easy to break and too hard solution will take so long time to create a block that the adversary with huge computation power have a chance to alter the data before creating and verifying a block. Rather a medium hard problem will be better. It is used to create the current block's hash.

This is to show the verifiers that the block creator have spent sufficient amount of computation power before creating the block and also to delay the process a little bit and this is called POW (Proof of Work). In bitcoin the problem is to find the first hash of given values which is having 'd' number of leading 'zero's where the 'd' represents the difficulty of the problem i.e. the more 'd' gets it will take longer to calculate. Typical value of 'd' is 32 bits and average delay for the whole block addition (create, verify then add) is about 10 minutes.

1.5.10 Consensus Mechanism

It is the contract or the protocol by which the blocks are verified and the winning blocks are added to everyone's ledger as well as the central server. The actual consensus algorithm is not published for security reasons. But by possible ways or reverse eengineering people have created different models. Some of those models are:

1. Probable bitcoin consensus mechanism

2. Paxos (Part-Time Parliament) consensus mechanism [4, 5]

3. Raft consensus mechanism [7]

Bitcoins one is probably the simplest one, but having bugs. The Paxos allows different types of sources and faults to come, it learns and fixes it. Raft does not allow any fault to happen.

The basic overall way in which consensus happen in bitcoin might be the following,

1. The transaction comes to server from client nodes;

2. Server stores that in file system database as unverified transaction;

3. At the point of interval of 10 minutes server broadcasts it to the miners network i.e. to every miner;

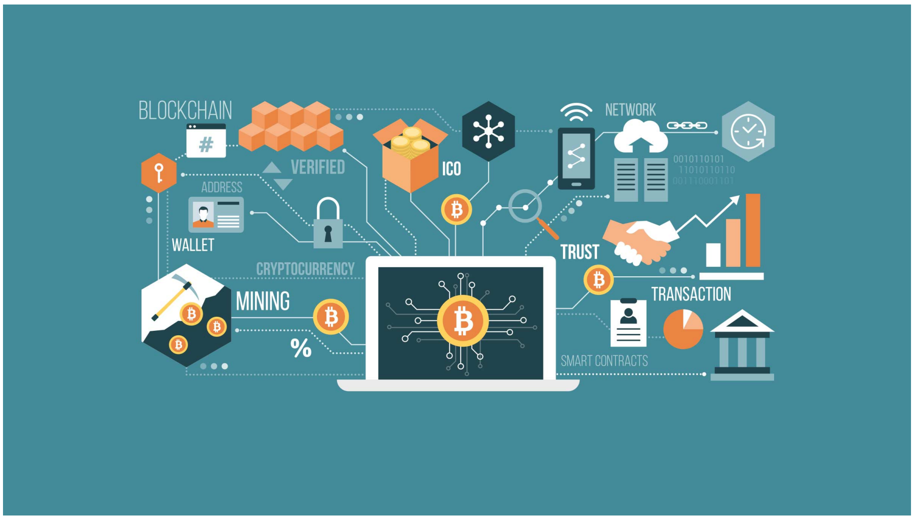

Figure 1.2: Overview of BitCoin Technology (BCTs).

4. Each miner individually verifies and adds correct transaction in a block. Each node holds its block creation and validates the new block as soon as it gets new block from network. Who's block is valid and introduced first to network will be added to everyone's chain and they will start making new block on top of it. Sometimes forks might be created for the networking distance between distant nodes, at that moment conflict will come. If a new introduced block's previous hash does not match with the last block's current hash the node requests to the network to get the missing blocks one by one until the blocks match, it validates it, delete the wrong blocks (make them orphan) and add missing blocks to own chain to resolve fork and maintain longest updated chain. So it is a race between miner nodes.

5. Server gets a copy from miners group and updates own copy and delete the added tansactions from file system.

1.5.11 Applications of BCTs

Block-Chain Technology provides one major advantage over conventional centralized database system: immunity from unexpected data changes or Hacks, which gives rise to numerous applications. Some of them include

- Crypto-currency: Creating and transferring digital money, Data Mining.
- Military applications: Secure and verified records of Every Military Events and documents.
- Structural health Monitoring

- e-Biding Systems
- Election System or e-Voting Systems
- Selling Records and other Commercial Applications
- Music Copyright Verification System
- Integrity Analysis of Media Files

1.5.12 Security and Integrity in BCTs

As the data is not sitting on a single data server so there is no security issue for Server Hacking. And also the hash-Chain with Cryptography makes it near to impossible to figure out or change previous data block in the blockchain. Before adding any data block with the help of consensus mechanism the blocks are verified with the digital signature of the node and some solution of nonce (the number of times the cllient application required to calculate the mathematical problem) and various other meta data. As in some interval the system is refreshing itself, if some error seems to be occured, it backs up itself by contacting the miners (the trusted nodes), compare their files and back up with the valid one. So no integrity problem is there, unless and until the internet works fine.

Chapter 2

Background

As we all know that in India the politics and the social media is a big thing for people to consider as an important part of life. But the problem is that some social media users does think of exploiting with the content either by downloading the video or recording on screen and uploading it to the social media platforms. Those new videos might be very sensitive and controversial and that becomes viral.

What is needed is an digital verification system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other the video files related information without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

The process of image integrity checking is old job and many algorithms have been made. The following is the overview of them.

1. Store the real images in some database. Match the testing image with the existing one bit by bit.

2. Like in previous case, store it and check the new one's hash with the existing one's hash.

3. Searching the image in different databases using object detection and context similarity.

4. Every digital image is created in an electronic device which is having some unique property or signature in the world. The softwares that has created the image are designed in such a way that they put the digital signature in the metadata field of the image, let's say EXIF data for JPEG images. Any software that knows the byte signatures can detect the random image is original or not by checking the signature and named context that lies inside of the image file.

5. By checking the image context with reality or possiblity, some images can be checked.

6. An expert or hacker not only seeks for the context or visual similarity, but an expert knows that secret figures (WaterMarking) secret messages (Steganography) can be embedded in the image file, because the main image is a 2d matrix of intensity values (list of integers for BnW, RGBA, CMYK, or other) at different pixel positions.

7. The modern approach is to use deep neural network that learns and tries to detect image integrity in about 90% efficiency and accuracy.

8. Image Forensics uses about all of the previous ones.

Related Works:

1. There are plenty of image storing and searching by image in the websites. In the list [,] they use the image searching by name, context.

2. Some of them uses object detection [,]

3. Some website programs use reverse image search like [,]

4. There are plenty of softwares like the photo eding softwares itself like Photoshop, GNU Image Manipulation Program, and other editors.

5. In [1] they used CNN for image comparing and many other that can be mentioned for well known face recognition problem.

6. In [2] they used a mechanism for video integrity analysis.

Chapter 3

Proposal

Using the concepts of Block-Chain we propose our algoorithms for Image Integrity Analysis D-APP system.

We are trying to build a web-based social media application like WhatsApp and that will be connected with the online server running blockchain in it. The process of verification of the media (text, image or video) for checking fakeness or scam in a nutshell will be as follows,

As mentioned earlier, the main jobs of our project to build a small social media platform that will store images, show them in news feed and not only that, someone can bring an image and perform search operation in the system to check if we know about the image or not, i.e. if the image is valid or not.

The system we are proposing is a hybrid of previous strict systems in terms of Permission (Permissioned/Non-permisionalized) and Centralization (Centralized/Decentralized) along with some extra mechanisms. And also we preserve the scalability of the implementation that it is not necessary to have a dedicated huge computation power for all.

3.1 Workflow

The following is the control flow of the overall processes,

1. An user will have to log into the system first to be a part of the system;

2. There are two types of users, some are normal users and rest are miners who have some extra powers as well as responsibilities.

3. Normal users are those who will come to share their images by upload or taken from their camera interface and also search images for checking. In their home page they can scroll and click an image for viewing, search signatures, download images to own devices.

Figure 3.1: Consensus mechanism

4. The miners are the creators of the blocks i.e. they will have computation power to generate the blocks, store the blocks in their own computers and broadcast them in the miners network, gossip among each others. They verify transactions before putting into a block.

3.2 System Design

In one side of the system, the normal user can request the system either to store an image or to verify. On storing request the server will help the client's device to resize the image in a particular dimention (image preprocessing) and generate the correct request format in terms of transaction (Tx). Then the server will create a copy of it making a symmetric key encryption over assymetric key encryption of the user and store in own file system database in unverified or new transaction file.

On a certain time interval (not so long, not so small, bitcoin takes 10 minutes, we will take it 5 minutes) the miners will be verifying these pool of unverified transactions and adding them in new blocks in parallel through consensus mechanism. Our consensus mechanism works as follows,

3.2.1 Consensus mechanism

1. At a time interval the server will broadcast the transactions or the miners will get them on request from the server database through XML-HTTP or SFTP protocol.

2. The miners either individually or by gossiping will verify the transaction data. Verifying by own will create the uniqueness, by gossiping maintains the correctness that prevent

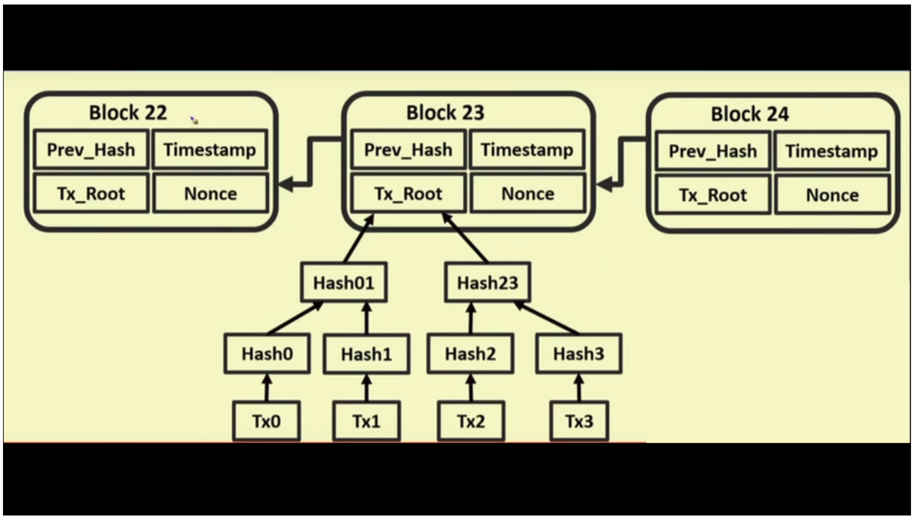

Figure 3.2: Block Creation

the fault from any side (server, other miners, itself).

3. After they delete the invalid transaction set they will start creating new block out of it.

4. After a node discovers or completes creating a new block, it broadcasts to the miner network and waits for confirmation and others blocks. After a node gets a new block, it holds it's own computation and validate the new block. If the new block is valid and the node have all previous blocks that should be linked in the chain by previous hash the node keeps it into own temporary space, and wait for others for confirmation, and if not valid then resume own computation. After all blocks say we have a valid block (either own or someone else's) to each other, they compare with what other nodes are having. The winner is one of the blocks which is having most of the criterias below,

   (a) Biggest in size $\Longrightarrow$ It is having maximum number of transactions

   (b) Bigger nonce $\Longrightarrow$ Most computation power is spent for it.

   (c) Lowest rewarded miner $\Longrightarrow$ To remove partiality and create balance and good understanding between miners.

   After 5 minutes the server requests the miners to get the decided block to be added in the chain. Server validates the block and adds it in the chain and remove the transactions that are either added or marked as invalid from the transaction pool.

5. On request of an image verification, first the image metadata is being checked (1st phase). If the metadata contains the our system generated signature, it will decrypt it and check in that particular block or the range of the blocks for the transaction that contains the image as well as the user's profile for it. Becacuse while creating downloadable file the system

encrypts and adds the metadata in the image file. If found, it generates the confirmation message and the links regarding it. And if not then the system requests for time and search the entire blockchain database for hashes (2nd phase). If found it does the same. And still if it not found, it runs a deep CNN for searching for contexts or by object for maximum possible proof (3rd phase). The result it gives now is the final result.

6. The process continues again and again recursively.

Chapter 4

Implementation

The setup we have is like this: Computer Name: Dell
Computer OS: Ubuntu 16.04 LTS
Runtime Platform:
Server: Localhost, Apache, MySQL, PHP, PHP-MyAdmin
Client: Browser (Chrome, FireFox), JavaScript-5, Active Internet Connection

The list of tools and important functions used:

- Image API
- File API
- SHA-256
- Base-64
- Image
- AES
- RSA

We have tried to make basic blockchain platform without video processing right now for a single node processing. The screenshot of the process is the following,

This picture shows how we have created the basic blockchain platform with the following tools:

- Image API
- SHA-256

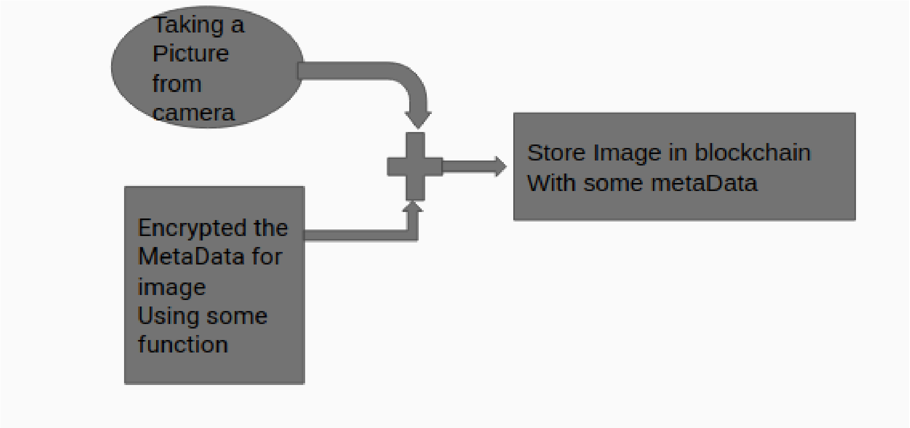
Taking a Picture from camera
Encrypted the MetaData for image Using some function
Store Image in blockchain With some metaData

Figure 4.1: Image Insertion

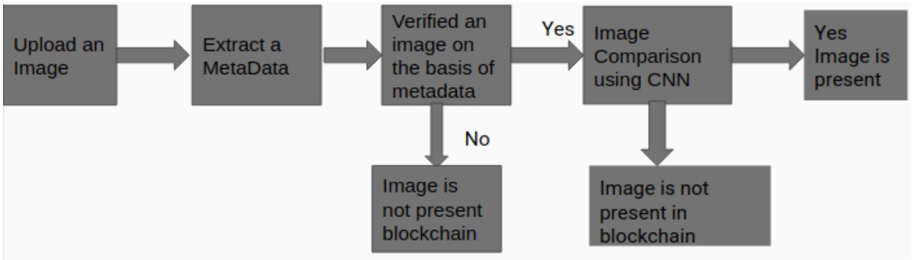
Upload an Image
Extract a MetaData
Verified an image on the basis of metadata
Yes
Image Comparison using CNN
Yes Image is present
No
Image is not present blockchain
Image is not present in blockchain

Figure 4.2: Image Verification

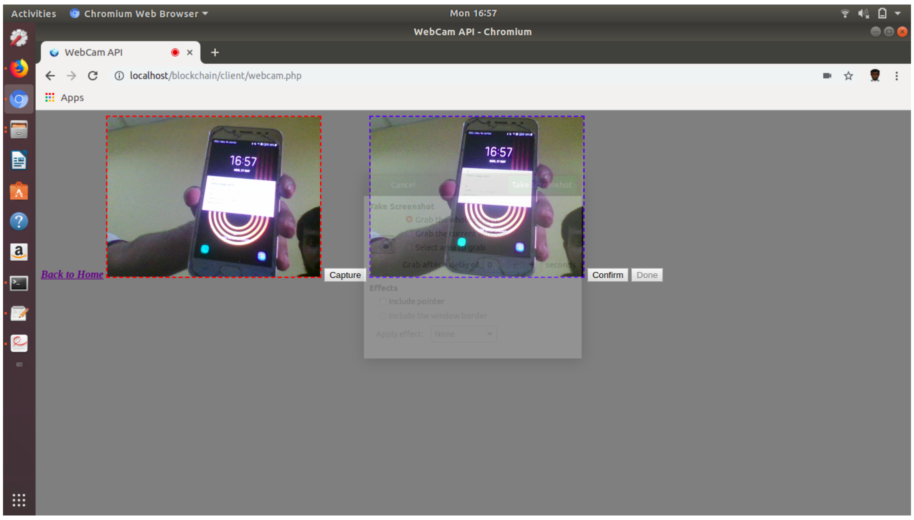

Figure 4.3: Image Verification

Figure 4.4: . Implementation Runtime in Terminal

There is a very simple concensus mechanism we have used to create the temporary (as the blocks are created and destroyed in the logical memory by Computer Program) private (as right now it is residing in our computer, not in a web server) ledger.

As we are implementing in Java, there are publicly functioning classes along with a class containing main function.
The classes we have are:
Block
Transaction
String Utility
Main

The String Utility contains some manipulation with the SHA-256 program, implemented by Java.

The Block contains the following: Id, Timestamp, Data, Prev hash, Nonce

For containing the hashes we are using List datatype in Java. Before adding any block we are checking 3 things, that are:
Timestamp: The time that it has been created
Previous Hash: parent for current block
Nonce: Consensus mechanism

while using CreateBlock() and MineBlock()
In the transaction class we have, Source Name (Hash): transaction from
Destination Name (Hash): transaction to

From main, we are creating transactions: calling Transaction class visiting transactions: receiving hashes
We have planned to include online platform the following way,

Figure 4.5: . System Model

In this model we are suggesting that there should be client-server system before involving into the blockchain system. So user have to login first with the hashed id and password. Then the person can apply for either Storing data into system or verify data from the system. The

Server will perform the task.

Figure 4.6: . Process Diagram with Hashes

Figure 4.7: . Verification Hash

Chapter 5

Conclusion and Further Work

This synopsis provides a detailed description of an Practical implementation of Online Biding system which provides Secure Key Exchange and agreement. We have implemented system for

- Capturing or uploading image;
- Show status in Home and Profile Page;
- Processing the image;
- Signing the image
- Create transactions
- Create block in server
- Adding to chain

5.0.0.1 Further Work

The next targets are to create a system that will standalone perform the following tasks

- making a peer-to-peer network for android that will perform the following tasks
- connecting to Online Verifier service
- sending and Receive messages containing data between peers
- improving Verification Service
- connecting in mobile application
- estimating risks and reconfigure concensus protocols

[6, 2] [8]

Bibliography

[1] Srikar Appalaraju and Vineet Chaoji. Image similarity using deep cnn and curriculum learning.

[2] Adam Hemlin Billström and Fabian Huss. Video integrity through blockchain technology. 2017.

[3] Michaeel J. Casey and Paul Vigna. *The Truth Machine: The Blockchain and the Future of Everything*. HarperCollins, 2018.

[4] Leslie Lamport. Paxos made simple. 2001.

[5] David Mazieres. Paxos made practical.

[6] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[7] Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm (extended version). 2014.

[8] Wikipedia. Blockchain. https://en.wikipedia.org/wiki/Blockchain, Oct 2014. Accessed on 2018-12-01.