
Video Integrity Analysis with BlockChain Technology

Bachelor Thesis

By

Anuj Kr. Pathak
Sayan Shankhari



A thesis submitted to

IIIT Kalyani

for the partial fulfillment of the degree of

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

December, 2018

Certificate

This is to certify that the thesis entitled **Video 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 per the regulation of **IIT 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, Ph.D)

Assistant Professor

Department of Computer Science and Information Systems

Indian Institute of Information Technology Kalyani

IIT Kalyani Campus, West Bengal 741235, India. December 2018

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

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 Kr. Pathak

Id. No.: 00000102

Sayan Shankhari

Id. No.: 00000121

December, 2018

Abstract

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 Video Scam. This project tries to find a way to prevent it with a new upcoming technology.

Keywords: Peer-to-Peer network, Proof-of-work, Proof-of-Stake, Distributed Ledger (Hyper-Ledger), Consensus Mechanism, Hash Function, Video Formatting, Video File formats, Water-marking, Android, WebServer.

Contents

1	Introduction	1
1.1	BlockChain	1
1.2	Applications of BCTs	1
1.3	Security and Integrity in BCTs	2
1.4	Thesis Contribution	2
1.5	Roadmap of the Thesis	3
2	Background	1
3	Details	1
3.1	Transaction	1
3.2	Timestamp	2
3.3	Proof of Work	2
3.4	Network	2
3.5	Incentive	2
3.6	Claiming Memory	3
3.7	Simplified Video Verification	3
3.8	Privacy	3
4	Proposal	1
5	Implementation	1
6	Conclusion and Further Work	6
7	Bibliography	7

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.

1.1 BlockChain

A BlockChain 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. 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-Server processors, and a power source. The nodes communicate with each other using web-services and self-organized.

1.2 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

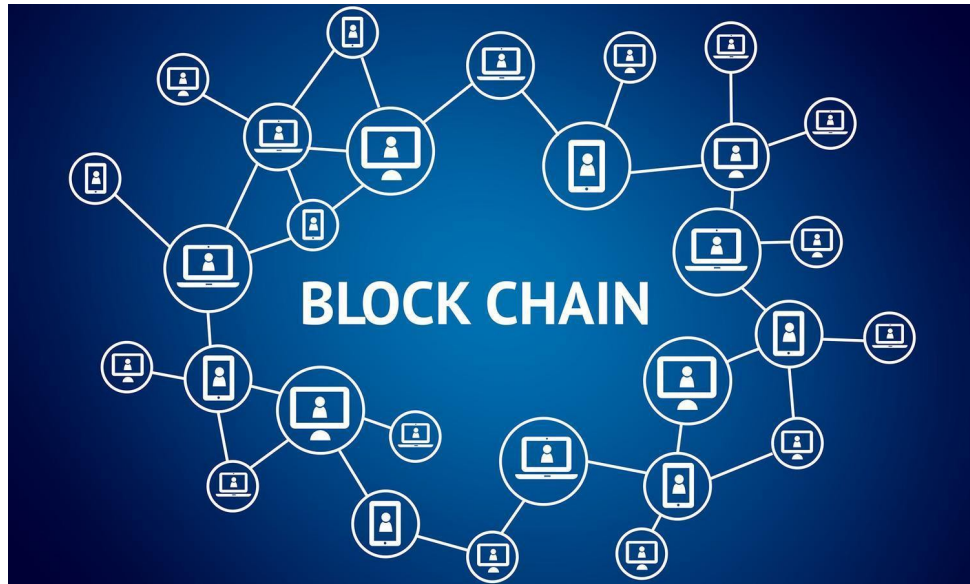


Figure 1.1: Overview of Block-Chain Technology (BCTs).

- Election System or e-Voting Systems
- Selling Records and other Commercial Applications
- Music Copyright Verification System
- Video Message integrity Testing

1.3 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.

1.4 Thesis Contribution

The main contributions of the 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.

1.5 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 BCTs, applications of BCTs, security aspects of BCTs.
3. The Chapter 3 discusses in details the basis of ECC, some definitions for security and some mathematically intractable problems.
4. The Chapter 4 introduces the proposed authentication framework after highlighting the motivations behind this work.
5. The implementation of Chapter 5, where an informal implementation of the proposed protocol has been discussed.
6. The Chapter 6 comprises of the conclusion and further work of the Project in future.

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.

Chapter 3

Details

The process underlying the block-chain technology are the following:

- Creating Account and Be a Part of the Network with an hash id
- Creating own Data block
- Request the BCT System to add it by sending it to the open Network in secure or encrypted way
- The system then send the block to other nodes for verification
- The other nodes can verify it with solving some nonce and report to system
- After being verified the System will add the new block to the chain and update it to every peer's copy of the hyper-ledger
- Data mining is useful when to check again and again for the integrity of data and the performance of the system

The following is the overview of the BitCoin implementation using concepts of blockchain,

3.1 Transaction

We define a block data masked with chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.

3.2 Timestamp

The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.

3.3 Proof of Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-work system. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash. For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.

3.4 Network

The steps to run the network are as follows:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

3.5 Incentive

By convention, the first transaction in a block is a special transaction (the block is called genesis block) that starts a new block owned by the creator of the block. This adds an incentive for nodes

to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended.

3.6 Claiming Memory

Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree, with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.

3.7 Simplified Video Verification

It is possible to verify video without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he/she can get by querying network nodes until he's/she's convinced he/she has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.

3.8 Privacy

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.

Chapter 4

Proposal

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

We are trying to build a Mobile 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,

- User logs into the account with hashed id
- User shares media into the social platform by capturing or texting
- System checks for match with previous hash if the content is already present in the network
- System stores the data as a block into the blockchain, along with creating the cryptographic hashes
- User sends video for verification
- System hashes it and looks into the blocks for same video
- If found, System decodes the file and search for the metadata containing the source and Watermarks
- System process the result
- System publishes the expected result
- User finds it in his/her mobile client app

Chapter 5

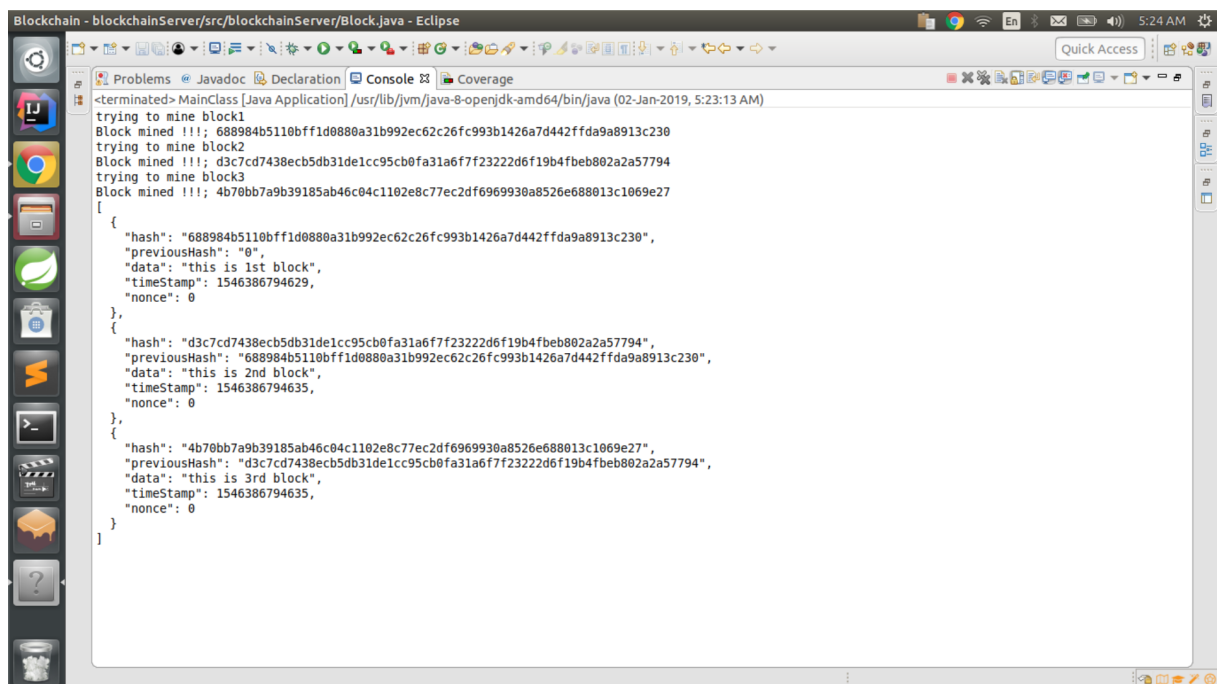
Implementation

The setup we have is like this: Computer Name: Lenovo

Computer OS: Ubuntu 16.04 LTS

Runtime Platform: Java 1.0.0_131

We have tried to make basic blockchain platform without video processing right now. The screenshot of the process is the following, This picture shows how we have created the basic



```
<terminated> MainClass [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (02-Jan-2019, 5:23:13 AM)
trying to mine block1
Block mined !!!; 688984b5110bff1d0880a31b992ec62c26fc993b1426a7d442ffda9a8913c230
trying to mine block2
Block mined !!!; d3c7cd7438ecb5db31de1cc95cb0fa31a6f7f23222d6f19b4fbeb802a2a57794
trying to mine block3
Block mined !!!; 4b70bb7a9b39185ab46c04c1102e8c77ec2df6969930a8526e688013c1069e27
{
  "hash": "688984b5110bff1d0880a31b992ec62c26fc993b1426a7d442ffda9a8913c230",
  "previousHash": "0",
  "data": "this is 1st block",
  "timestamp": 1546386794629,
  "nonce": 0
},
{
  "hash": "d3c7cd7438ecb5db31de1cc95cb0fa31a6f7f23222d6f19b4fbeb802a2a57794",
  "previousHash": "688984b5110bff1d0880a31b992ec62c26fc993b1426a7d442ffda9a8913c230",
  "data": "this is 2nd block",
  "timestamp": 1546386794635,
  "nonce": 0
},
{
  "hash": "4b70bb7a9b39185ab46c04c1102e8c77ec2df6969930a8526e688013c1069e27",
  "previousHash": "d3c7cd7438ecb5db31de1cc95cb0fa31a6f7f23222d6f19b4fbeb802a2a57794",
  "data": "this is 3rd block",
  "timestamp": 1546386794635,
  "nonce": 0
}
}
```

Figure 5.1: . Implementation Runtime in Terminal

blockchain platform with the following java tools:

- Image API
- SHA-256 API

There is a very simple consensus 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,

System Model

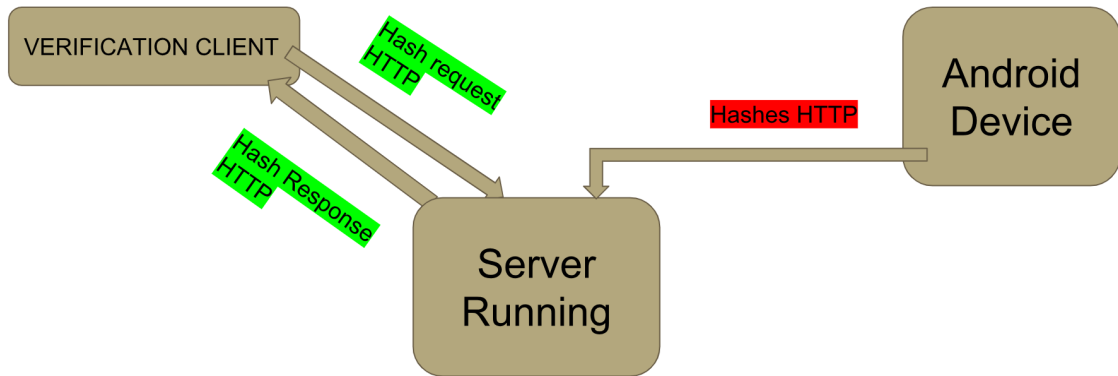


Figure 5.2: . System Model

Process diagram of adding hashes into blockchain

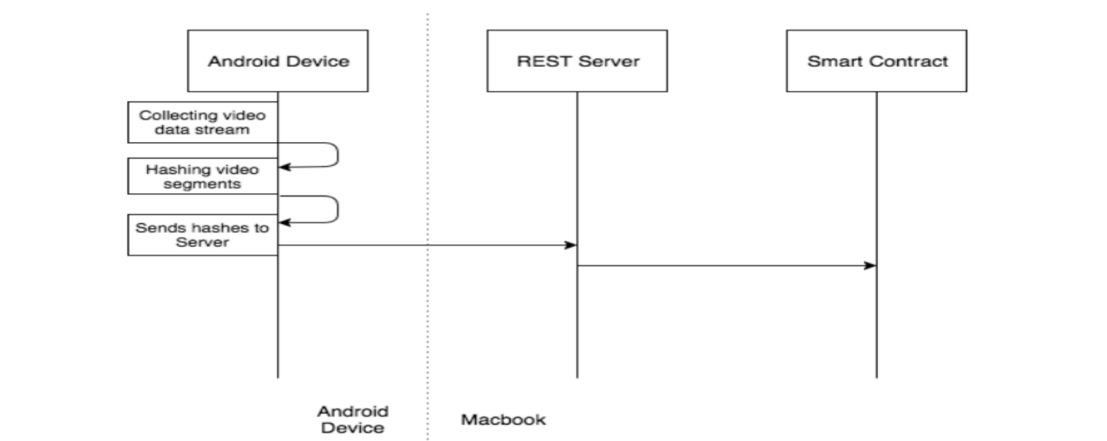


Figure 5.3: . Process Diagram with Hashes

Verification of hash

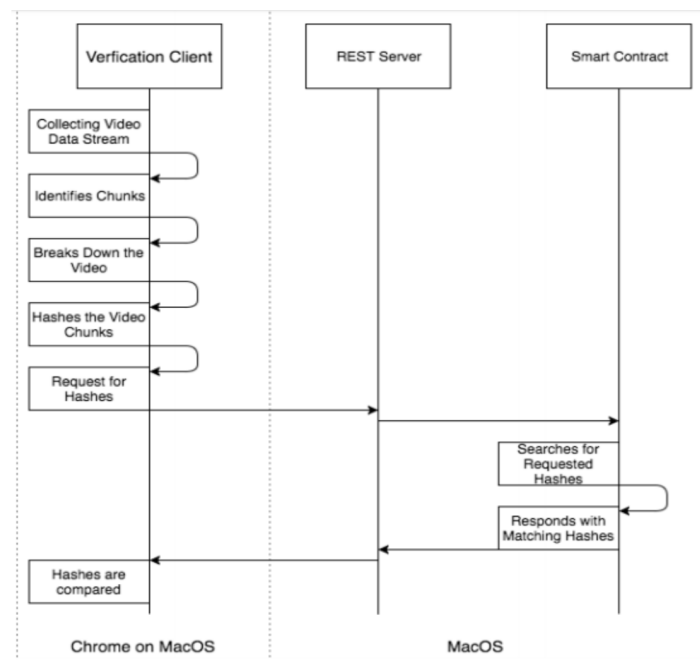


Figure 5.4: . Verification Hash

Chapter 6

Conclusion and Further Work

This synopsis provides a detailed description of an Practical implementation of Online Biding system which provides User Anonymity and Secure Key Exchange and agreement.

Further Work

The next target is to make,

- a software for android that will standalone perform the following tasks
 - Capturing the image or video
 - Processing the video
 - Signing into video
 - Connect to Online Verifier service
 - Send and Receive messages containing Text, Image or Video
- along with the Online Verification Service which will
 - Connect to android app on request
 - Take data, verify and publish result

Chapter 7

Bibliography

We are very thankful to the following contents with their creators which definitely helps to grow our project,

- Official Site: <https://www.blockchain.com/>
- Wiki: <https://en.wikipedia.org/wiki/Blockchain>
- Inspired by: "Video Integrity through Blockchain Technology" -by ADAM HEMLIN BILLSTRÅUM and FABIAN HUSS
- Inspired by: "Bitcoin: A Peer-to-Peer Electronic Cash System" -by Satoshi Nakamoto (The Inventor of BlockChain and BitCoin)