

# **ENHANCING CHAIN OF CUSTODY MANAGEMENT THROUGH BLOCKCHAIN TECHNOLOGY**

**A PROJECT REPORT**

*Submitted by*

<b>KESAVAN A</b>	<b>2021016</b>
<b>NARESHKUMAR E</b>	<b>2021024</b>
<b>SURYA R</b>	<b>2021042</b>
<b>UMESH S</b>	<b>2021046</b>

*In partial fulfillment for the award of the degree  
of*

**BACHELOR OF ENGINEERING**

*in*

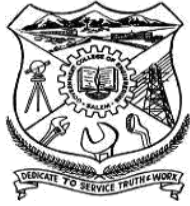
**COMPUTER SCIENCE AND ENGINEERING**



**GOVERNMENT COLLEGE OF ENGINEERING  
SALEM – 636011**

**ANNA UNIVERSITY: CHENNAI- 600025**

**MAY 2024**



# **ENHANCING CHAIN OF CUSTODY MANAGEMENT THROUGH BLOCKCHAIN TECHNOLOGY**



## **A PROJECT REPORT**

*Submitted by*

<b>KESAVAN A</b>	<b>2021016</b>
<b>NARESHKUMAR E</b>	<b>2021024</b>
<b>SURYA R</b>	<b>2021042</b>
<b>UMESH S</b>	<b>2021046</b>

*in partial fulfillment for the award of the degree  
of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**GOVERNMENT COLLEGE OF ENGINEERING, SALEM**

*(An Autonomous Institution Affiliated to Anna University, Chennai, NACC Accredited)*

**ANNA UNIVERSITY: CHENNAI - 600 025**

**MAY 2024**

# **GOVERNMENT COLLEGE OF ENGINEERING, SALEM**

*(An Autonomous Institution Affiliated to Anna University, Chennai, NACC Accredited)*

**ANNA UNIVERSITY: CHENNAI 600 025**

## **BONAFIDE CERTIFICATE**

Certified that this project report “**ENHANCING CHAIN OF CUSTODY MANAGEMENT THROUGH BLOCKCHAIN TECHNOLOGY**” is the Bonafide work of “**KESAVAN A (2021016), NARESHKUMAR E (2021024), SURYA R (2021042), UMESH S (2021046)**” who carried out the project work under my supervision during the academic year 2023-2024.

### **SIGNATURE**

**Dr. A.M. KALPANA M.E., Ph.D.**

### **PROFESSOR &**

### **HEAD OF THE DEPARTMENT**

Computer Science and Engineering,  
Government College of Engineering,  
Salem-636 011.

### **SIGNATURE**

**Dr. P. THARANI, M.E., Ph.D.**

### **ASSISTANT PROFESSOR &**

### **SUPERVISOR**

Computer Science and Engineering,  
Government College of Engineering,  
Salem-636 011.

Submitted for the Project Viva-Voce Examination held at Government college of Engineering, Salem-11 on .....

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

We convey our heartfelt gratitude to the honorable and respected principal **Dr. R. VIJAYAN, M.E., Ph.D.** for his encouragement and support for the successful completion of the project

We would like to thank our Head of the Department, Computer Science and Engineering, **Dr. A.M. KALPANA, M.E., Ph.D.** who took keen interest till the completion of our project work by providing all the necessary information for developing a good system.

We are grateful to our project guide **Dr. P. THARANI, M.E., Ph.D.** Assistant Professor, Computer Science and Engineering for her remarkable guidance and the incessant help in all possible ways from the beginning to accomplish the project successfully.

We are grateful to our project coordinator **Prof. S. RUBA, M.E.,** Computer Science and Engineering for her valuable suggestion and immediate counselling throughout the completion of the project. We also extend our gratitude to the teaching faculty members and non-teaching staff members for their timely support.

We also acknowledge with a deep sense of reverence and gratitude towards our parents, family members and friends, who supported us for the successful completion of the project.

## **ABSTRACT**

In legal proceedings, digital evidence plays a pivotal role, comprising electronic documents, recordings, and transaction records that influence outcomes significantly. However, ensuring the integrity and security of this evidence presents challenges like data tampering and unauthorized access. To address these issues, a proposed solution leverages blockchain technology to establish a robust Chain of Custody (CoC) and deploy effective tamper detection mechanisms. Through collaborative validation and recording of investigation activities on a blockchain ledger within a private network, stakeholders enhance integrity and mitigate vulnerabilities.

The solution, termed DB-CoC architecture, employs fuzzy hash functions to standardize forensic practices, fortifying the credibility of digital evidence. By orchestrating cryptographic protocols, the architecture provides a secure refuge for digital artifacts, assuring stakeholders of trustworthiness throughout the investigation lifecycle and facilitating evidence preservation for future reference.

## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	<b>iv</b>
	<b>LIST OF TABLES</b>	<b>ix</b>
	<b>LIST OF FIGURES</b>	<b>x</b>
	<b>LIST OF ABBREVIATIONS</b>	<b>xi</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 OVERVIEW	
	1.2 BLOCK CHAIN	<b>3</b>
<b>2.</b>	<b>LITERATURE REVIEW</b>	<b>8</b>
	2.1.	
<b>3.</b>	<b>SYSTEM ANALYSIS</b>	<b>12</b>
	3.1 EXISTING SYSTEM	<b>12</b>
	3.2 PROPOSED SYSTEM	<b>13</b>
	3.3 FEASIBILITY STUDY	<b>14</b>
<b>4</b>	<b>SYSTEM SPECIFICATION</b>	<b>19</b>
<b>5</b>	<b>SOFTWARE DESCRIPTION</b>	<b>20</b>
	5.1 PYTHON 3.7.4	<b>20</b>
	5.2 MYSQL	<b>23</b>

	5.3 WAMP SERVER	24
	5.4 BOOTSTRAP 4	25
	5.5 FLASK	25
6	<b>PROJECT DESCRIPTION</b>	28
	6.1 PROBLEM DEFINITION	28
	6.2 MODULE DESCRIPTION	29
	6.2.1 CoC FORENSIC TOOL	29
	6.2.2 DB BLOCKCHAIN INTEGRATION	29
	6.2.3 CASE AND DIGITAL EVIDENCE MANAGEMENT	32
	6.2.4 EVIDENCE ACCESS CONTROL	33
	6.2.5 EVIDENCE LOG	34
	6.2.6 ATTACKER MODULE	34
	6.2.7 TAMPER DETECTION	35
	6.2.8 SYSTEM USER	35
	6.3 SYSTEM ARCHITECTURE	37
	6.4 DATA FLOW DIAGRAM	38
	6.5 UML DIAGRAM	41
	6.5.1 USE CASE DIAGRAM	42

7	<b>SYSTEM TESTING</b>	43
8	<b>CONCLUSION AND FUTURE ENHANCEMENT</b>	46
	8.1 CONCLUSION	46
	8.2 FUTURE ENHANCEMENT	46
	Appendix-1 (Source Code)	47
	Appendix-2 (Screenshots)	80
	REFERENCE	89



## LIST OF TABLES

<b>Section</b>	<b>Subsection</b>	<b>Page</b>
6.1	CoC Evidence table	30
6.2	Access control table	32
6.3	Digital Evidence Request table	33
6.4	Digital Evidence Response table	34
6.5	Evidence log table	35
7.1	Testing Results	45

## **LIST OF FIGURES**

<b>Section</b>	<b>Subsection</b>	<b>Page</b>
1.1	Chain of Custody	1
1.2	Process flow of chain of custody	2
6.1	Process flow of Blockchain	27
6.2	System Architecture	37
6.3	Data Flow Diagram – Level 0	38
6.4	Data Flow Diagram – Level 1	39
6.5	Data Flow Diagram – Level 2	40
6.6	USE CASE DIAGRAM	42

## LIST OF ABBREVIATIONS

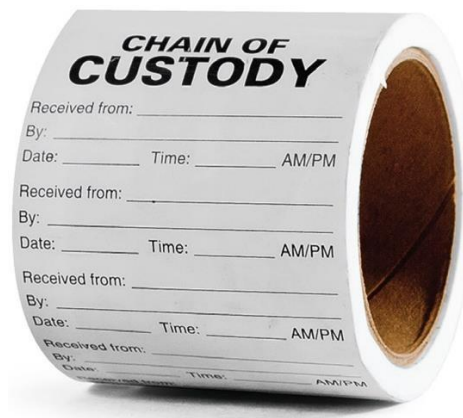
CoC	Chain of Custody
CTPH	Context-triggered Piecewise Hashing
DFD	Data Flow Diagram
DLT	American Sign Language
DPoS	Delegated Proof of Stake
FB	Fuzzy Blockchain
PoA	Proof of Authority
PoS	Proof of Stake
PoW	Proof of Work
pBFT	Practical Byzantine Fault Tolerance
RH	Rolling Hashing

# CHAPTER 1

## INTRODUCTION

### 1.1 OVERVIEW

Chain of custody (CoC) is a legal term that refers to the ability to guarantee the identity and integrity of the evidence from collection through to reporting of the test results. It also refers to the document or paper trail, showing the recovery, custody, control, transfer, analysis and disposition of evidence. Strict observance of the legal CoC in specimens obtained is mandatory to guarantee the reliability and integrity of the analysis. Chain of custody refers to the documentation that establishes a record of the control, transfer, and disposition of evidence in a criminal case. Evidence in a criminal case may include DNA samples, photographs, documents, personal property, or bodily fluids that were taken from a defendant or discovered at the scene of an alleged crime is depicted in Figure 1.1.



**Figure 1.1. Chain of Custody**

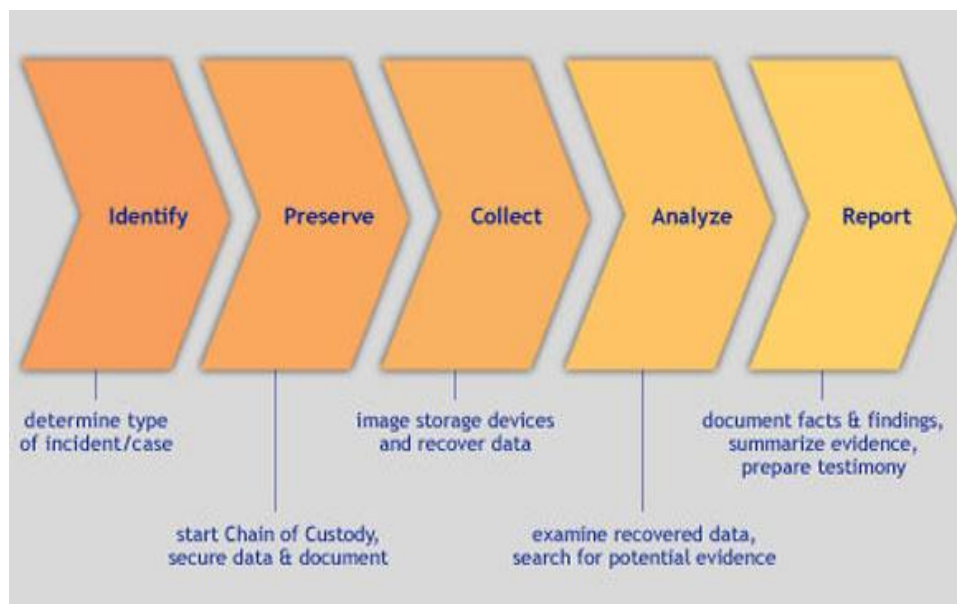
### **Process of a Chain of Custody for Digital Evidence**

To protect digital evidence, the chain of custody consists of four steps. These are:

- **Data collection:** When the chain of custody begins from the first item of data collected. The examiner must 'tag' each item acquired and document the source,

how and when it was collected, where it is stored, and who has access to it.

- **Examination:** When the chain of custody must be documented outlining the process undertaken. It is useful to capture screenshots throughout the process to show the tasks completed and the evidence exposed.
- **Analysis:** When it may be appropriate to capture the chain of custody information.
- **Reporting:** When the chain of custody is documented into a statement that explains the tools used, the sources of data, methods of extraction used, the process of analysis, and issues encountered, and how these were controlled. Ultimately, it is this statement that must make it clear that the chain of custody has been maintained throughout the process and that the evidence given is legally defensible. The process flow is depicted in Figure 1.2.



**Figure 1.2: Process flow of chain of custody**

## **1.2 BLOCK CHAIN**

Blockchain is defined as a ledger of decentralized data that is securely shared. Blockchain technology enables a collective group of select participants to share data. With blockchain cloud services, transactional data from multiple sources can be easily collected, integrated, and shared. Data is broken up into shared blocks that are chained together with unique identifiers in the form of cryptographic hashes. Blockchain provides data integrity with a single source of truth, eliminating data duplication and increasing security. In a blockchain system, fraud and data tampering are prevented because data can't be altered without the permission of a quorum of the parties. A blockchain ledger can be shared, but not altered. If someone tries to alter data, all participants will be alerted and will know who make the attempt.

### **1.2.1 Three types of blockchain**

- **Public blockchain**

A public, or permission-less, blockchain network is one where anyone can participate without restrictions. Most types of cryptocurrencies run on a public blockchain that is governed by rules or consensus algorithms.

- **Permissioned or private blockchain.**

A private, or permissioned, blockchain allows organizations to set controls on who can access blockchain data. Only users who are granted permissions can access specific sets of data. Oracle Blockchain Platform is a permissioned blockchain.

- **Federated or consortium blockchain.**

A blockchain network where the consensus process (mining process) is closely controlled by a preselected set of nodes or by a preselected number of stakeholders.

### **1.2.2 Process Flow of Blockchain Step 1 – Record the transaction**

A blockchain transaction shows the movement of physical or digital assets from one party to another in the blockchain network. It is recorded as a data block as each transaction.

## **Step 2 – Gain consensus**

Most participants on the distributed blockchain network must agree that the recorded transaction is valid. Depending on the type of network, rules of agreement can vary but are typically established at the start of the network.

## **Step 3 – Link the blocks**

Once the participants have reached a consensus, transactions on the blockchain are written into blocks equivalent to the pages of a ledger book. Along with the transactions, a cryptographic hash is also appended to the new block. The hash acts as a chain that links the blocks together. If the contents of the block are intentionally or unintentionally modified, the hash value changes, providing a way to detect data tampering.

## **Step 4 – Share the ledger**

The system distributes the latest copy of the central ledger to all participants.

### **1.2.3 Blockchain in Chain of Custody**

A Blockchain-based Chain of Custody for Evidences Management in Digital Forensics presents requirements that a Chain of Custody process should have:

**Integrity:** the evidence has not been altered or corrupted during the transferring.

**Traceability:** the evidence must be traced from the time of its collection until it is destroyed.

**Authentication:** all the entities interacting with a piece of evidence must provide an irrefutable sign as recognizable proof of their identity.

**Verifiability:** the whole process must be verifiable from every entity involved in the process.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Chain of Custody in Digital Forensic Investigations: Issues and Challenges**

**Author:** Ibrahim Baggili, Frank Breitingner, Harshvardhan J. Pandit, and Andrew Marrington

**Year:** 2022

**Link:** <https://ieeexplore.ieee.org/document/9651416>

##### **Problems Identified**

The article identifies various challenges in maintaining chain of custody in digital forensic investigations, including lack of standardization, difficulty in preserving digital evidence, and the need for specialized training and expertise.

##### **Objective**

The objective of the article is to provide an overview of the challenges faced in maintaining chain of custody in digital forensic investigations and to suggest solutions to overcome them.

##### **Methodology**

The article is a literature review and does not involve any experimental methodology.

##### **Merits**

The article provides a comprehensive overview of the challenges in maintaining chain of custody in digital forensic investigations and offers solutions to overcome them.

##### **Demerits**

The article does not provide any experimental data or empirical evidence to support its claims.



## **2.2 A Comprehensive Framework for Digital Forensic Chain of Custody Management**

**Author:** Salma Alharbi, Zhiyuan Tan, and Ameer Al-Nemrat

**Year:** 2022

**Link:** <https://ieeexplore.ieee.org/document/9651413>

### **Problems Identified:**

The article identifies various challenges in maintaining chain of custody in digital forensic investigations, including the need for standardization and the potential for evidence contamination.

### **Objective**

The objective of the article is to propose a comprehensive framework for digital forensic chain of custody management that addresses these challenges.

### **Methodology**

The article proposes a comprehensive framework for digital forensic chain of custody management and evaluates its effectiveness through simulations.

### **Merits**

The article proposes a practical solution for maintaining chain of custody in digital forensic investigations that addresses various challenges.

### **Demerits**

The article does not provide empirical evidence to support the effectiveness of the proposed framework in real-world scenarios.

## **2.3 Implementing Chain of Custody in Cloud Forensics: Issues, Challenges, and Solutions**

**Author:** Ahmet Okutan and Ali Dehghantanha

**Year:** 2021

**Link:** <https://www.sciencedirect.com/science/article/pii/S2666827021000125>

### **Problems Identified**

The article identifies various challenges in maintaining chain of custody in cloud forensics, including the difficulty of preserving and collecting digital evidence and the lack of standardization.

### **Objective**

The objective of the article is to propose a framework for implementing chain of custody in cloud forensics that addresses these challenges.

### **Methodology**

The article proposes a framework for implementing chain of custody in cloud forensics and evaluates its effectiveness through simulations.

### **Merits**

The article proposes a practical solution for maintaining chain of custody in cloud forensics that addresses various challenges.

### **Demerits**

The article does not provide empirical evidence to support the effectiveness of the proposed framework in real-world scenarios.

**2.4 Digital Forensics and Chain of Custody: A Review Author:** Israa Abdulwahid and Ziad Alkhwaja

**Year:** 2020

**Link:** <https://www.sciencedirect.com/science/article/pii/S2405452620300275>

### **Problems Identified**

The article identifies various challenges in maintaining chain of custody in digital forensic investigations, including the lack of standardization and the difficulty of preserving and collecting digital evidence.

## **Objective**

The objective of the article is to provide a comprehensive review of chain of custody in digital forensic investigations and to identify areas for future research.

## **Methodology**

The article conducts a comprehensive review of chain of custody in digital forensic investigations and identifies gaps in the existing literature.

## **Merits**

The article provides a comprehensive review of the existing literature on chain of custody in digital forensic investigations and identifies areas for future research.

## **Demerits**

The article does not propose any new solutions for maintaining chain of custody in digital forensic investigations.

## **CHAPTER 3**

### **SYSTEM ANALYSIS**

#### **3.1 EXISTING SYSTEM**

Nowadays, forensic software is used as better evidence for the process of the description and identification of the electronic user, digital signature and automatic audit trail, etc. Still, there is a great distance from the usual chain of custody software to the effective questions of the court and users. Nowadays, this process is executed by the process of CoC. The CoC is a set of consecutive documentation that records the order of custody, its control, transfer, analysis, and physical or electronic evidence. CoC contains unsafe steps during the process of investigation and at the time of submitting the evidence in court. The current traditional digital forensic process lacks standardized procedures and mechanisms making it inherently vulnerable to various tampering and forgery occurrences against the recent cybercrime incidents.

##### **3.1.1 Disadvantages**

- Low automation level in the process of data
- preservation
- High risk level in the process of data preservation
- Lack of safety guarantee of digital data
- Lack of mutual trust
- It is unable to detect similarities at a higher level of abstraction, for example, semantically.
- It is unable to properly match two image files that contain the same semantic image but are stored in various file kinds and formats as a result of their differing binary encodings.

## 3.2 PROPOSED SYSTEM

The core of the proposal is an efficient forensics architecture that leverages blockchain technology for establishing the Chain of Custody (CoC) and deep learning models for tamper detection. This combination aims to address security and forensic aspects throughout the investigation lifecycle.

- **DB-CoC Architecture**

The proposed architectural solution, referred to as DB-CoC, is designed to provide robust information integrity, prevention, and preservation mechanisms. It involves the permanent and immutable storage of evidence (chain of custody) in a private, permissioned, and encrypted blockchain ledger.

- **Blockchain for Chain of Custody:**

Blockchain technology is suggested to establish a secure and tamper-evident Chain of Custody. Participants in the investigation process create a private network to agree on and record various activities on the blockchain ledger.

- **Fuzzy Hash Functions**

The utilization of fuzzy hash functions is highlighted, enabling forensic investigators to handle permissible alterations of digital evidence. This involves standardizing forensic processes to ensure consistency and reliability.

- **Data Provenance and Traceability**

The DB-CoC architecture promises complete data provenance and traceability, ensuring trust between chain of custody events during the collection, storage, analysis, and interpretation of digital evidence.

### 3.2.1 Advantages

- In terms of security, nobody, not even the owners of the document, ought to be able to modify it once it has been recorded.
- The evidence must be traced from the time of its collection until it is destroyed.
- The evidence has not been altered or corrupted during the transferring.

- All the entities interacting with a piece of evidence must provide an irrefutable sign as recognizable proof of their identity.
- In terms of security, nobody, not even the owners of the document, ought to be able to modify it once it has been recorded.
- The evidence must be traced from the time of its collection until it is destroyed.
- The evidence has not been altered or corrupted during the transferring.
- All the entities interacting with a piece of evidence must provide an irrefutable sign as recognizable proof of their identity.
- Digital evidence is safeguarded from alteration or misrepresentation
- The evidence management system reduces input errors and eliminates duplication.
- Reduces liability.

### 3.3 FEASIBILITY STUDY

A feasibility study is a crucial step in the planning process of any project, whether it's a business venture, a new product development, or an infrastructure project. It is conducted to assess the practicality and viability of the proposed project before committing significant resources such as time, money, and effort. The primary objective of a feasibility study is to determine whether the project is feasible and worth pursuing based on various factors, including technical, economic, legal, operational, and environmental considerations. Here's a detailed explanation of each aspect of a feasibility study

**Technical feasibility:** Technical feasibility assesses whether the proposed project can be implemented from a technical perspective. It involves evaluating the availability of necessary technology, infrastructure, and expertise required to develop and execute the project successfully.

**Economic feasibility:** Economic feasibility evaluates the financial viability of the project by analyzing its costs and potential returns. It involves estimating the initial investment required to start the project, ongoing operational costs, and projected revenue streams

**Environmental Feasibility:** Environmental feasibility assesses the potential environmental impact of the proposed project. It involves evaluating factors such as resource consumption, waste generation, pollution, and habitat disturbance.

Three key considerations are involved in the feasibility analysis,

- Economic feasibility
- Technical feasibility
- Operational feasibility

### **3.3.1 Technical Feasibility:**

#### **Technology Requirements:**

- Assess the specific technologies needed for the project, such as programming languages, frameworks, databases, and hardware infrastructure.
- Evaluate the availability and compatibility of these technologies in the current environment.
- Consider alternative technologies or approaches if the required ones are not readily available.

#### **Expertise:**

- Identify the technical skills required to develop and maintain the project.
- Assess whether the team possesses these skills or if additional expertise needs to be acquired.
- Consider outsourcing certain tasks or hiring new team members with the necessary expertise.

#### **Compatibility:**

- Ensure that the proposed solution integrates seamlessly with existing systems,

platforms, and standards.

- Evaluate potential conflicts or compatibility issues and devise strategies to address them.
- Consider using standardized protocols and APIs to facilitate interoperability.

#### **Scalability:**

- Determine the scalability requirements of the project, including anticipated user growth and data volume.
- Design the solution to accommodate future expansion without significant architectural changes.
- Consider using scalable cloud-based infrastructure or distributed systems to support growth.

#### **Risk Assessment:**

- Identify potential technical risks and obstacles that may impede project success.
- Conduct a thorough risk analysis to understand the likelihood and impact of each risk.
- Develop contingency plans and mitigation strategies to address potential challenges.

### **3.3.2 Economic Feasibility:**

#### **Cost Estimation:**

- Estimate the total costs associated with the project, including development, deployment, and operational expenses.
- Consider both direct costs (e.g., hardware, software licenses) and indirect costs (e.g., training, maintenance).
- Account for potential cost overruns and unexpected expenses by building contingency reserves into the budget.

#### **Benefit Analysis:**

- Identify and quantify the anticipated benefits of the project, such as increased revenue, cost savings, or efficiency gains.



- Use qualitative and quantitative methods to assess the value proposition of the project.
- Consider intangible benefits such as improved customer satisfaction or competitive advantage.

#### **ROI Calculation:**

- Calculate the return on investment (ROI) by comparing the expected benefits to the total costs over the project's lifecycle.
- Determine the payback period – the time it takes for the project to recoup its initial investment.
- Conduct sensitivity analysis to assess how changes in key assumptions impact the project's financial viability.

#### **Risk Analysis:**

- Identify economic risks such as market volatility, regulatory changes, or competitive pressures.
- Assess the potential impact of these risks on the project's financial performance.
- Develop risk mitigation strategies to minimize the negative consequences of economic uncertainties.

### **3.3.3 OPERATIONAL FEASIBILITY:**

#### **User Acceptance:**

- Engage stakeholders and end-users early in the project to gather feedback and ensure their needs are addressed.
- Conduct usability testing and user acceptance testing to validate the system's functionality and usability.
- Provide training and support to users to facilitate their adoption of the new system.

#### **Organizational Impact:**

- Assess how the project will affect existing processes, workflows, and organizational structure.

- Identify potential barriers to adoption and develop strategies to overcome resistance to change.
- Communicate effectively with stakeholders to manage expectations and gain their buy-in.

### **Training and Support:**

- Develop comprehensive training programs to educate users and staff on how to use the new system effectively.
- Provide ongoing support and assistance to address any issues or challenges that arise during and after implementation.
- Establish feedback mechanisms to gather user input and continuously improve the system based on their needs.

### **Change Management:**

- Implement change management strategies to minimize disruption and maximize the chances of successful adoption.
- Communicate the benefits of the project and the reasons for change to stakeholders at all levels of the organization.
- Anticipate and address potential sources of resistance through proactive engagement and dialogue.

### **Sustainability:**

- Evaluate the long-term sustainability of the project in terms of its ongoing maintenance, support, and operational costs.
- Develop a plan for sustaining the project beyond the initial implementation phase, including funding sources and resource allocation.
- Consider the environmental and social impacts of the project and incorporate sustainable practices where feasible

## **CHAPTER 4**

### **SYSTEM SPECIFICATION**

#### **4.1 Hardware Requirements**

Processor	: Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAM
Disk space	: 320 GB
Operating systems	: Windows® 10

#### **4.2 Software Requirements**

Server Side	: Python 3.7.4(64-bit) or (32-bit)
Client Side	: HTML, CSS, Bootstrap
IDE	: Flask 1.1.1
Back end	: MySQL 5.
Server	: WampServer 2i
BC DLL	: JSON

## CHAPTER 5

### SOFTWARE DESCRIPTION

#### 5.1Python 3.7.4

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.



Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc. The biggest strength of Python is huge collection of standard library which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)

- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, Selenium)
- Test frameworks
- Multimedia
- Scientific computing
- Text processing and many more.

### 5.1.1 Pandas

Pandas are a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. pandas are a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.



Pandas is mainly used for data analysis and associated manipulation of tabular data in Data frames. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features. The development of pandas introduced into Python many comparable features of working with Data frames that were established in the R programming language.

The panda's library is built upon another library NumPy, which is oriented to efficiently working with arrays instead of the features of working on Data frames.

### 5.1.2 NumPy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.



NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

### 5.1.3 Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.



Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

### 5.1.4 Seaborn

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Visualization is the central part of Seaborn which helps in exploration and understanding of data.



Seaborn offers the following functionalities:

- Dataset oriented API to determine the relationship between variables.
- Automatic estimation and plotting of linear regression plots.
- It supports high-level abstractions for multi-plot grids.
- Visualizing univariate and bivariate distribution.

### 5.1.5 Scikit Learn

Scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.



Scikit-learn (formerly scikits. learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

## 5.2 MYSQL

MySQL tutorial provides basic and advanced concepts of MySQL. Our MySQL tutorial is designed for beginners and professionals. MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company. MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. There are also

given MySQL interview questions to help you better understand the MySQL database.



MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications. It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is My Ess Que Ell. However, you can pronounce it in your way. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

### **5.3 WAMPSEVER**

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your database.



WAMP Server is a reliable web development software program that lets you create web apps with MYSQL database and PHP Apache2. With an intuitive interface, the



application features numerous functionalities and makes it the preferred choice of developers from around the world. The software is free to use and doesn't require a payment or subscription.

## 5.4 BOOTSTRAP 4

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.



It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap

**Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops

**Mobile-first approach:** In Bootstrap, mobile-first styles are part of the core framework

**Browser compatibility:** Bootstrap 4 is compatible with all modern browsers (Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera).

## 5.5 FLASK

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

## Using an IDE

As good as dedicated program editors can be for your programming productivity, their utility pales into insignificance when compared to Integrated Developing Environments (IDEs), which offer many additional features such as in-edit or debugging and program testing, as well as function descriptions and much more.



Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have formed a validation support. Instead, Flask supports the extensions to add such functionality to the application. Although Flask is rather young compared to most Python frameworks, it holds a great promise and has already gained popularity among Python web developers. Let's take a closer look into Flask, so-called "micro" framework for Python. Flask was designed to be easy to use and extend. The idea behind Flask is to build a solid foundation for web applications of different complexity. From then on you are free to plug in any extensions you think you need. Also you are free to build your own modules. Flask is great for all kinds of projects. It's especially good for prototyping. Flask is part of the categories of the micro-framework. Micro-framework is normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that sometime you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins.

Flask, often hailed as a micro-framework, embodies simplicity and extensibility at its core. Unlike some other frameworks, Flask deliberately avoids including built-in abstractions for tasks like database handling and validation support. Instead, it provides a solid foundation upon which developers can add functionality using extensions. Despite being relatively young compared to other Python frameworks, Flask has garnered significant attention and adoption within the Python web development community.

The essence of Flask lies in its simplicity and ease of use. It offers a minimalist approach, allowing developers to start with a basic structure and progressively integrate additional features as needed. This flexibility makes Flask suitable for a wide range of projects, with a particular strength in rapid prototyping.

One of the defining characteristics of Flask is its classification as a micro-framework. Micro-frameworks typically have minimal dependencies on external libraries. While this lightweight nature offers advantages such as reduced overhead and simplified maintenance, it also necessitates more manual intervention at times. Developers may need to handle certain tasks themselves or augment the framework with additional plugins to meet specific requirements.

## **CHAPTER 6**

### **PROJECT DESCRIPTION**

#### **6.1 Problem Definition:**

- **Data Integrity and Trust:** Current systems lack reliability due to vulnerabilities in centralized databases, leading to concerns about data integrity and trustworthiness.
- **Transparency and Traceability:** Inadequate visibility into custody processes results in delays and disputes, necessitating real-time transparency and traceability.
- **Security Risks and Vulnerabilities:** Centralized systems are susceptible to security breaches, exposing custody data to unauthorized access and tampering.
- **Manual and Inefficient Processes:** Manual paperwork and administrative tasks lead to inefficiencies and errors, highlighting the need for automation and streamlined processes.
- **Compliance and Regulatory Challenges:** Meeting regulatory standards while maintaining accurate records poses complexity and requires effective management.
- **Cost and Resource Allocation:** Operational costs are high due to manual processes and reliance on intermediaries, necessitating cost-effective solutions.
- **Interoperability and Integration:** Integration with existing systems and ensuring interoperability is crucial for successful blockchain adoption.
- **Scalability and Performance:** As the volume of assets increases, scalability and maintaining performance become challenges that need to be addressed.

## 6.2 MODULE DESCRIPTION

### 6.2.1 CoC Forensic Tool

This module is intended to serve as an interface for authorization, access permissions, and media. It allows for the downloading of digital evidence and certificates of authenticity in line with access permissions and levels. The blockchain interface enables participants to see, invoke, and query blocks, transactions, and chain codes. The front end produces a hash of the digital evidence and a nonce that uniquely identifies it (Evidence ID). As the hash generates the ID and the value nonce is randomly selected to guarantee the uniqueness of the evidence's identification, it aids in preserving the integrity of digital evidence throughout its lifetime. This component is responsible for enabling communications between all the users. It incorporates access control and evidence management; creating a new record, evidence state verification, and disposal of evidence.

### 6.2.2 DB-Blockchain Integration

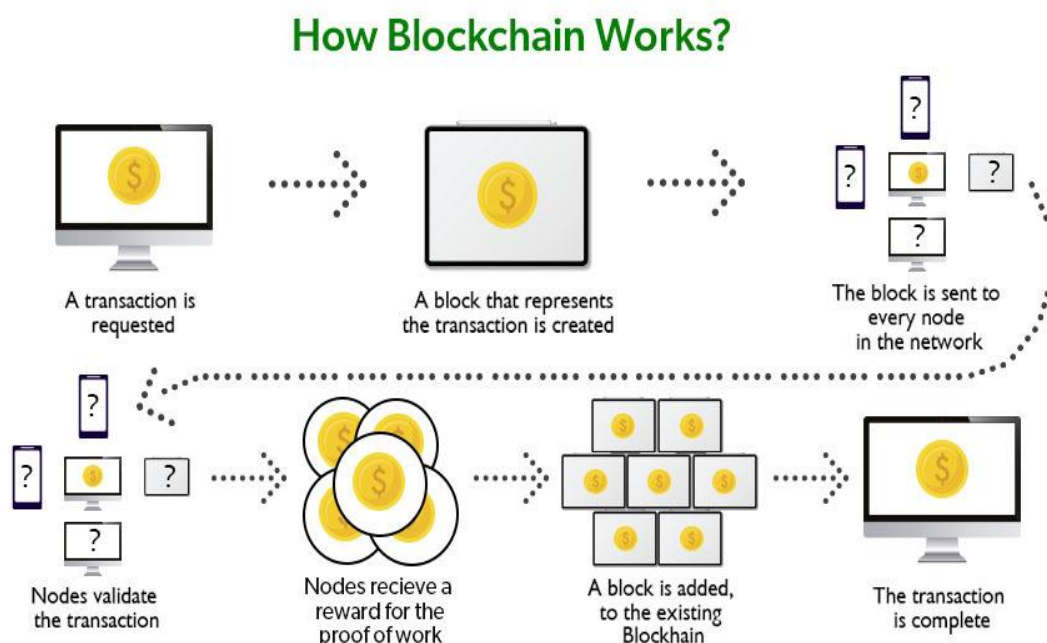
**Fuzzy Blockchain (FB):** This component describes the private blockchain implementation using, for instance, private Blockchain called Fuzzy Blockchain, as the main underlying system for cybercrime application. Participating roles and responsibilities will act as active nodes of the FB blockchain network. The FB contains an essential element to its structure, i.e., shared ledger or DLT, which will be able to log all collective and transferred evidence and immutability shared among all the different and authorized entities. The DLT is governed by lawmakers and law enforcement institutes.

The FB has three sub-functions, which together form the operation of FB. These are:

**Smart Contract:** Each transaction can be automated using a smart contract. A smart contract is a set of predetermined executable instructions based on the nature of a certain transaction or input. An output can also trigger another smart contract. For example, a case is created, the smart contract logs the submitter ID and associated

evidence provided by the analysis phase. Based on the analysis output, the smart contract initiates another instance to request more evidence from the submitter or witnesses. If the submitted evidence is sufficient for the case, then the smart contract proceeds to the analysis and investigation procedures.

**Consensus Node:** This is a function with a set of rules that is responsible for maintaining, verifying and approving BF records/transactions and updating the ledger. It also ensures trustworthiness when reliability, availability, accuracy, and authenticity are built in by design. There are different implementations of consensus algorithms, such as proof of work (PoW), proof of stake (PoS), delegated proof of stake (DPoS), practical byzantine fault tolerance (pBFT), proof of authority (PoA), etc. A private (permissioned) implementation of the CB model is suggested with the use of practical byzantine fault tolerance (pBFT) as a consensus algorithm. The pBFT is considered for the CB model with the assumption that some of the consensus nodes may act faultily or maliciously in the network, hence our taking proactive measures to ensure consistent and valid voting/validation. The pBFT does not scale to accommodate other blockchains or larger volume, but to maintain evidence handling, the author believes it should suffice.



## 6.1 Process flow of Blockchain.

### **6.2.2.1 Fuzzy Hashing**

To account for the uncertainty associated with evidence item changes, we utilized Fuzzy Hashing (FH) rather than conventional hashes such as SHA 256 in this project. FH, also known as Context-Triggered Piecewise Hashing (CTPH), is a mix of Piecewise and Rolling Hashing (RH). Unlike traditional hashes, where their hashes (checksums) can be interpreted as correct or incorrect, and as black or white, CTPH is more akin to the “grey hash type” as it can identify two files that are likely near duplicates of one another but would not be detected using traditional hashing methods. RH generates ‘segments’ of conventional hash strings by generating a pseudo-random value depending on the context of the input. In comparison, PH (Piecewise Hashes), such as conventional hashes, produce a final checksum for the whole picture. They circumvent the latter’s restrictions by segmenting the whole image into defined segments and then generating hash values for each of these parts. Finally, the produced values comprise the final hash sequence. FH employs the concept of PH to preserve data similarity in this study. Additionally, PH was designed to minimize possible mistakes during forensic imaging, ensuring that the data’s integrity is absolute and complete since only one hash segment is void.

### **6.2.2.2 Approximate Matching**

In DB-CoC, the system computes the similarity of two files based on their signatures throughout the comparison process. DB-CoC analyzes two strings and calculates the least number of operations required to convert one string into the other using an edit distance method based on Levenshtein distance. While DB-CoC is very efficient at detecting similarities between text files, it has a poor detection rate for images due to the possibility of an active adversary exploiting it.

### **6.2.2.3 DB – CoC Transaction Process**

- **Add**

Add a new evidence item to the blockchain and associate it with the given case identifier. For users’ convenience, more than one item id may be given at a

time, which will create a blockchain entry for each item without the need to enter the case id multiple times.

- **Check out**

Add a new checkout entry to the chain of custody for the given evidence item. Checkout actions may only be performed on evidence items that have already been added to the blockchain.

- **Check in**

Add a new check in entry to the chain of custody for the given evidence item. Check in actions may only be performed on evidence items that have already been added to the blockchain.

- **Log**

Display the blockchain entries giving the oldest first (unless -r is given).

- **Remove**

Prevents any further action from being taken on the evidence item specified. The specified item must have a state of CHECKEDIN for the action to succeed.

### **6.2.3 Case and Digital Evidence Management**

In this module, the government regulator register case and upload the digital evidence are to the DB-CoC Server. A typical approach involves storing a forensic image - a bit-by-bit copy - of the digital device which is done during the acquisition and preservation phase of digital forensic. Since certain device types can store terabytes of data, a corresponding environment to be able to store such large data is needed. Digital evidence is also more fragile and fleeting in nature than physical ones and therefore face the challenge of possible manipulation, e.g. due to acquisition technology. To make sure that evidence is valid, techniques like Fuzzy Hashing algorithm is used for the original data and the forensic image to assure that the copy is not corrupted.



### 6.2.3.1 Digital Evidence Creation

Besides, there are still other algorithms like evidence creation and evidence transfer. Evidence creation is a function when digital evidence first been submitted to the blockchain. This function takes evidence, case ID, owner ID, timestamp, evidence Description and current Location as input, these attributes are essential, and the previous evidence should be null due to the digital evidence is newly created.

Field	Type	Description
id	int(11)	Unique identifier for each evidence
case_id	varchar(20)	Unique identifier for the case the evidence belongs to
details	varchar(200)	Details or description of the evidence
filename	varchar(100)	Name of the file containing the evidence
dtime	timestamp	Timestamp indicating when the evidence was uploaded
status	int(11)	Status of the evidence (e.g., processed, pending)
upload_by	varchar(20)	Username of the user who uploaded the evidence

### 6.1 CoC Evidence Table

### 6.2.4 Evidence Access Control

This module presents a smart lock solution to be embedded into the evidence storage medium where it integrates blockchain smart contracts with a flexible web-based interface to allow authenticated parties involved in the forensic process to access evidence data while maintaining its security, integrity, and authenticity. Each party in contact with an evidence sample tries to open its lock, the system checks the party permissions and privileges through a smart contract that requests the party unique identifier (assigned from a central authority such as forensic lab) and the designated privileges such as request evidence, examine evidence, or transfer evidence.

Field	Type	Description
id	int(11)	Unique identifier for each access control entry
uname	varchar(20)	Username of the user accessing the evidence
eid	int(11)	ID of the evidence being accessed
case_id	varchar(20)	Case ID associated with the evidence
view_st	int(11)	View status (e.g., 1 for allowed, 0 for denied)

Field	Type	Description
download_st	int(11)	Download status (e.g., 1 for allowed, 0 for denied)
dtime	timestamp	Timestamp indicating when the access occured

## 6.2 Access Control Table

### 6.2.4.1 Digital Evidence Request

We assume that a lot of digital evidence has been submitted to the blockchain by the participants from the case. The participants could be police, prosecutor, lawyer, forensic and so on.

Field	Type	Description
id	int(11)	Unique identifier for each request
uname	varchar(20)	Username of the user making the request
message	varchar(200)	Message or reason for the request
reply	varchar(200)	Reply or response to the request

Field	Type	Description
status	int(11)	Status of the request (e.g., pending, approved, denied)
dtime	timestamp	Timestamp indicating when the request was made
cname	varchar(20)	Case name associated with the request

### 6.3 Digital Evidence Request Table

#### 6.2.4.2 Digital Evidence Response

Evidence transfer is a function we need to use when the evidence has to be transferred from someone to another one. We need to put this information on the blockchain to keep the entire footprint of the evidence intact so that the evidence can be trusted. This function takes previous evidence, evidence ID, owner ID, timestamp, current location and evidence description as input. It's worth noting that evidence ID and case ID should be the same as these attributes in previous evidence.

Field	Type	Description
id	int(11)	Unique identifier for each response
request_id	int(11)	Foreign key referencing the ID of the corresponding request
responder	varchar(20)	Username of the user responding to the request

Field	Type	Description
response	varchar(200)	Response or action taken regarding the request
dtime	timestamp	Timestamp indicating when the response was provided

## 6.4 Digital Evidence Response Table

### 6.2.5 Evidence Log

The evidence log keeps track of user interactions with digital evidence. This Evidence Log is implemented on the blockchain and contains information on each piece of evidence on which decision-making depends, including its ID, a description, the submitter's (creator's) identity, and the full history of owners up to the present one, including the dates of ownership transfers. The evidence log is built on top of a peer-to-peer network that includes all authorized entities. A network of this kind may be split into two distinct groups of nodes: (1) validator nodes: they are primarily responsible for maintaining a copy of the blockchain; validating transactions; and creating, proposing, and adding blocks to the chain (i.e., participate in the consensus protocol). (2) Lightweight nodes: they are considered clients of the chain since they just issue transactions and depend on validators to add and validate them.

Field	Type	Description
id	int(11)	Unique identifier for each log entry
event_type	varchar(50)	Type of event (e.g., upload, download, access granted)
username	varchar(20)	Username of the user performing

Field	Type	Description
		the action
case_id	varchar(20)	Case ID associated with the evidence
evidence_id	int(11)	ID of the evidence
event_desc	varchar(200)	Description of the event
event_time	timestamp	Timestamp of when the event occurred

## 6.5 Evidence Log Table

### 6.2.6 Attacker Module

In this module the attacker performs the following types of attack to change the evidence file. The name itself is an acronym for the following threat types:

#### **Spoofing:**

- The attacker impersonates another person or uses their password to act as that person. Spoofing is a threat to authenticity.

#### **Tampering:**

- It is the act of purposefully modifying data and violates the integrity of data.

#### **Denial of service (DoS):**

- A threat where the system becomes temporarily unavailable. These kinds of attacks lower the reliability of the system.

### 6.2.7 Evidence Tamper Detection

Tamper detection in blockchain using fuzzy hash functions involves the utilization of algorithms that generate unique identifiers for data blocks based on their content,

making alterations immediately evident. Unlike traditional hash functions, which produce fixed-length outputs, fuzzy hashing considers small variations in data, providing a degree of tolerance. By comparing fuzzy hashes of successive blocks, the blockchain system can promptly identify any discrepancies, ensuring the integrity and security of the distributed ledger.

### **6.2.8 System User**

The system model of DB-CoC consists of a Government Regulator, a police department, a court, a prison, victims, prosecution lawyers, defence lawyers, police investigators, crime scene analysts, witnesses, monitoring devices, a judge, a jury (with jurors)

#### **6.2.8.1 Government Regulator**

Government Regulator is a governmental agency, and it initializes the whole system. The motivation to use GR has two aspects. First, we believe there is a governmental agency that is not easily compromised given hardware-protected protocol running environments, rigorous monitoring, and detailed access logs. Hence, it can perform as a trusted authority. Second, the GR only works in system initialization, entity registration, and entity tracking. The first two phases do not conflict with the blockchain design.

The tracking function is considered here because the evidence management requires the ability to locate malicious insiders and protect justice. Based on the second reason, if the participants generate hash key material themselves, it would be more difficult to reveal their real identities.

- Login
- Register case and upload evidence file to the DB-CoC
- Create entity account and distribute login credentials

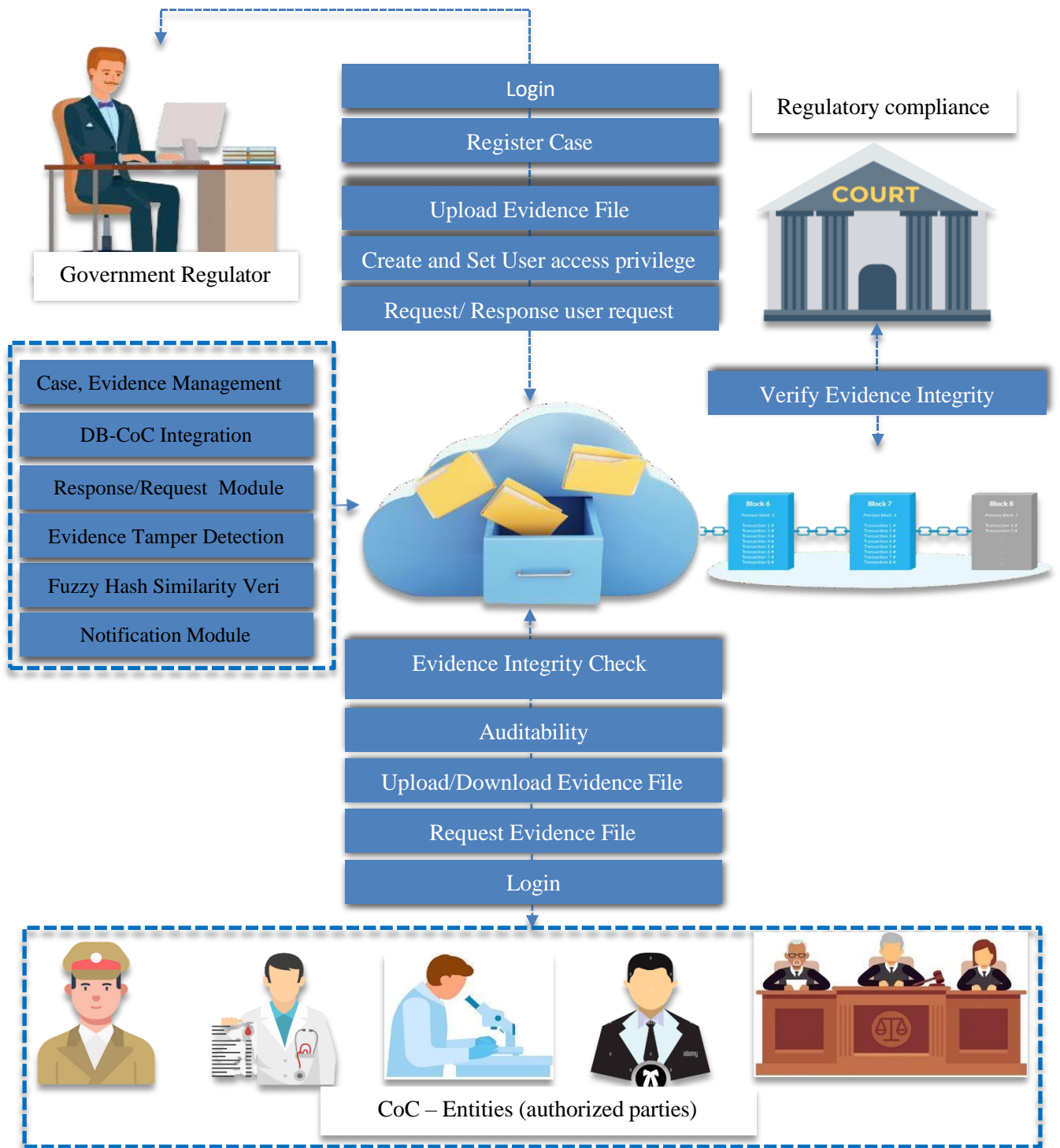
- Request/Response to the entity
- Verify the integrity of the digital stored evidence in the DB-CoC Blockchain.

#### **6.2.8.2 CoC – Entities**

- Login
- Request to View Evidence
- View Evidence



## 6.3 System Architecture

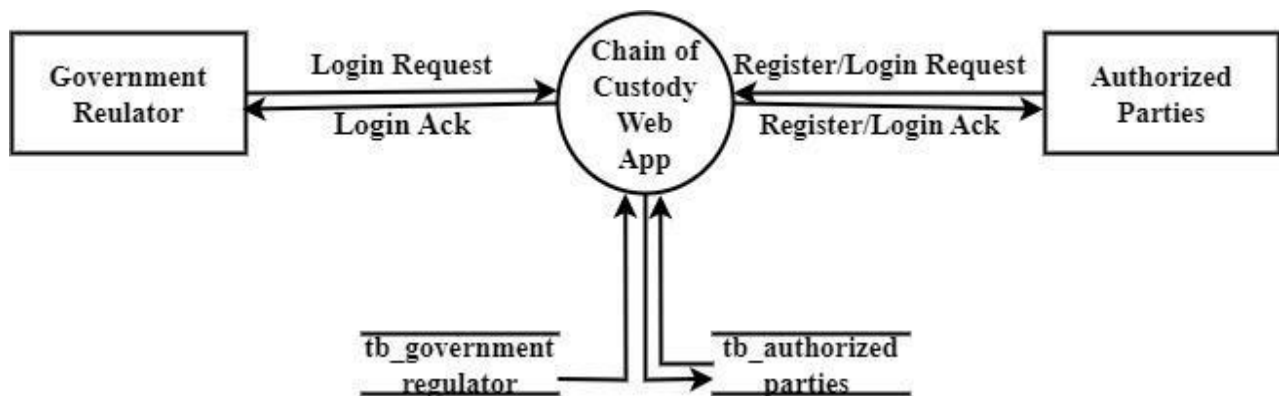


## 6.2 System Architecture

## 6.3 Data Flow Diagrams

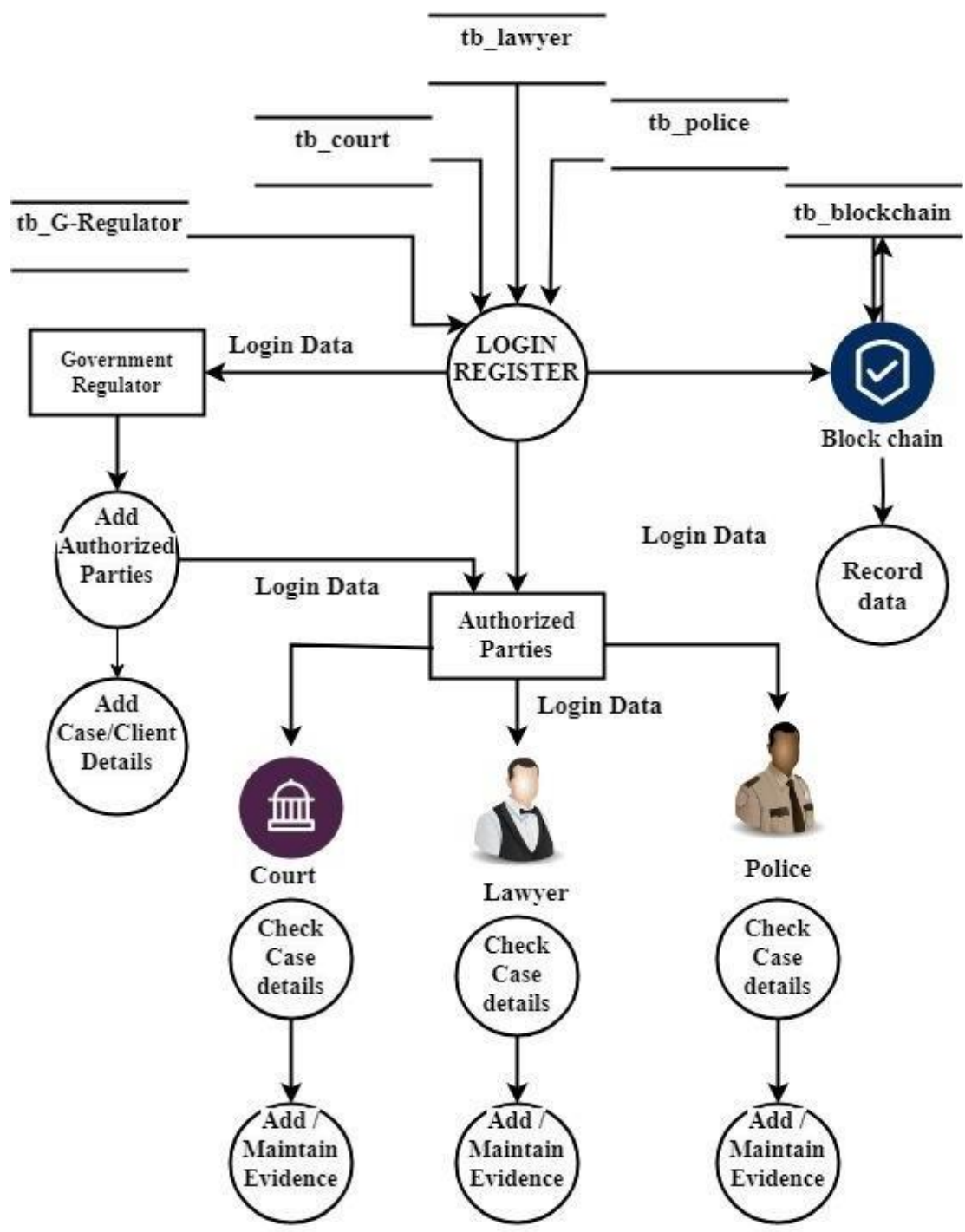
### 6.3.1 Data Flow Diagram - Level 0

A data flow diagram (DFD) is a graphical representation of the flow of data within a system. It shows how data moves through processes, stores, and external entities. DFDs consist of processes, which represent transformations of data; data stores, where data is held; data flows, which represent the movement of data between processes and stores; and external entities, which interact with the system. These diagrams help in understanding the system's data flow and processing logic, aiding in system analysis, design, and communication among stakeholders.



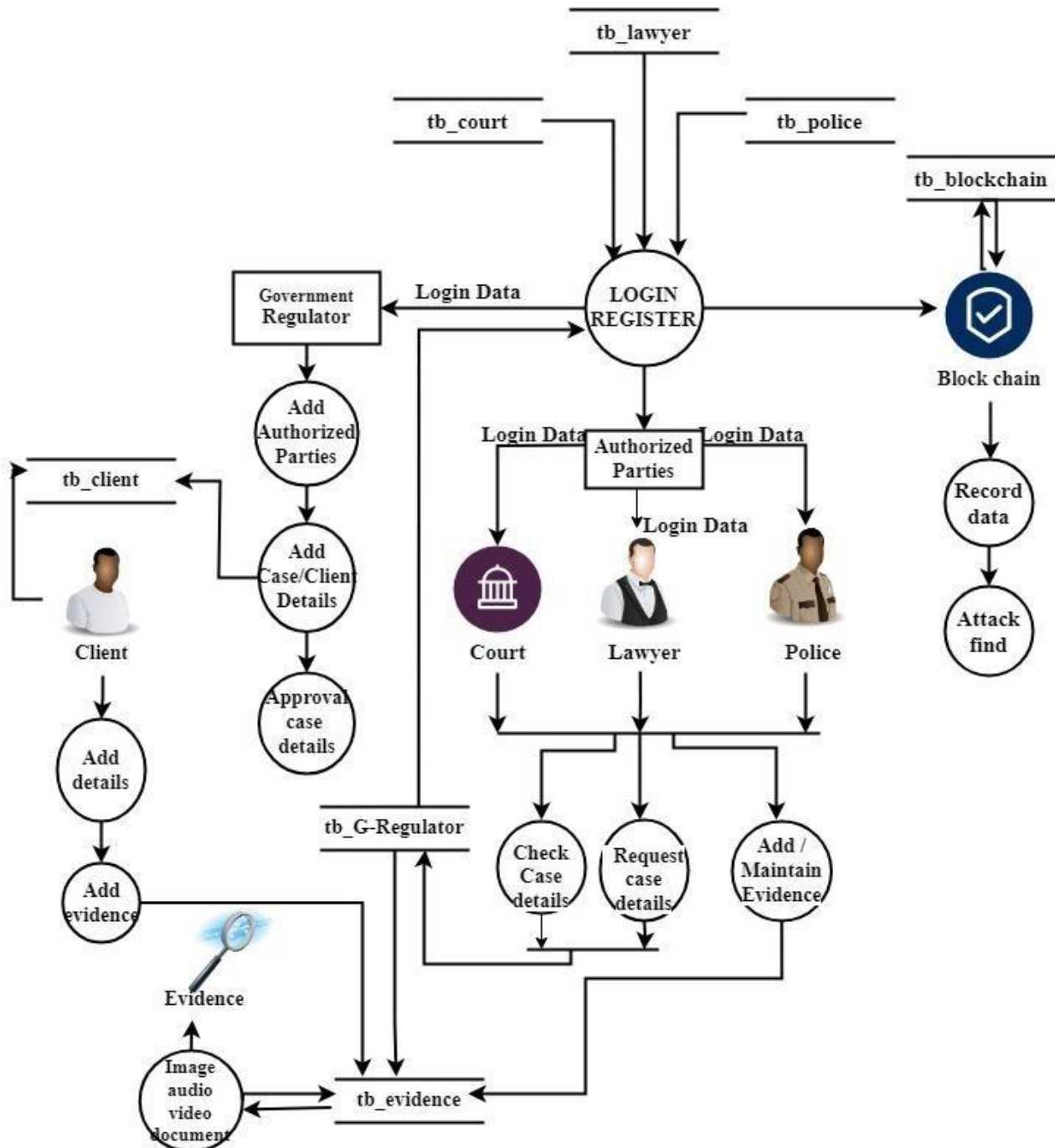
### 6.3 Data Flow Diagram - Level 0

6.3.2 Data Flow Diagram - Level 1



6.4 Data Flow Diagram - Level 1

### 6.3.3 Data Flow Diagram - Level 2



### 6.5 Data Flow Diagram - Level 2

## 6.5 UML DIAGRAM

Unified Modeling Language (UML) is a standardized modeling language used in software engineering for visualizing, specifying, constructing, and documenting the artifacts of software systems. UML provides a set of graphical notations for representing various aspects of software systems, including their structure, behavior, and interactions.

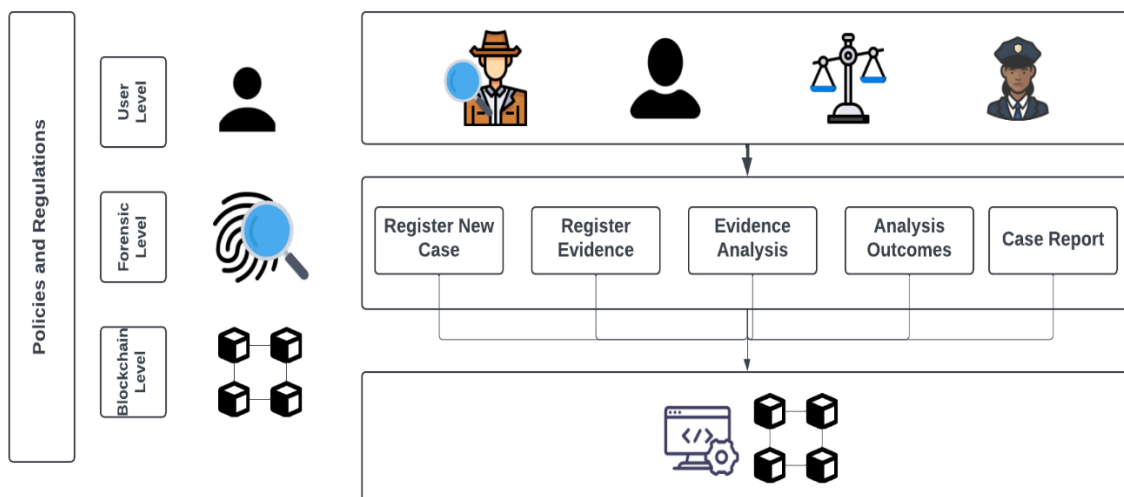
Here are some key elements of UML:

- **Class Diagrams:** Class diagrams represent the static structure of a system by showing classes, attributes, operations, and relationships between classes.
- **Use Case Diagrams:** Use case diagrams depict the interactions between users (actors) and the system to describe the system's functionality from the user's perspective.
- **Sequence Diagrams:** Sequence diagrams illustrate the interactions between objects over time, showing the sequence of messages exchanged among objects.
- **Activity Diagrams:** Activity diagrams model the flow of control within a system, depicting the actions and decisions that take place during the execution of a process or workflow.
- **State Machine Diagrams:** State machine diagrams represent the states of an object and the transitions between those states in response to events.
- **Component Diagrams:** Component diagrams show the physical components of a system and their relationships, including dependencies and interfaces.
- **Deployment Diagrams:** Deployment diagrams depict the physical deployment of

software components on hardware nodes, such as servers or devices.

### 6.5.1 USE CASE DIAGRAM

A use case diagram is a visual tool in software engineering that illustrates the interactions between users and a system. It provides a high-level overview of the system's functionality by outlining the various actions users can perform and how they interact with the system. Actors, which can be users or external systems, are represented along with use cases, which describe specific interactions or functionalities. This diagram helps in understanding the system's requirements, identifying user roles, and facilitating communication among stakeholders during the early stages of software development.



### 6.6 USECASE DIAGRAM

# **CHAPTER 7**

## **SYSTEM TESTING**

### **7.1 System Testing**

A blockchain is a chain of blocks that contains information. Its technology was originally intended to timestamp digital documents so that it's not possible to backdate or tamper with them. A blockchain technology or platform can be used to secure, store, and manage data in a decentralized and cryptic format. This addresses the current challenges of trust or data breach between B2B, B2C, and C2B entities. It was adopted by Satoshi Nakamoto in 2009 to create Bitcoin – a digital cryptocurrency. Blockchain technology has since then revolutionized the way businesses are conducted. It is at the core of digital currencies and utility tokens that have gone mainstream.

#### **Blockchain Testing**

Blockchain is being widely accepted in the industry. With the rise of popularity, we need to be ready to adapt existing testing strategies to blockchain technology. But the lack of best practices, the creation of suitable test data, and dealing with scale, security and performance are some of the key testing challenges in the blockchain. Blockchain testing assists in enabling smart records and ensures fraud security. Data in the blockchain are stored in blocks. Any change in the block will invalidate the subsequent blocks. This makes it important that whenever a new block is added, it is added in the right way. Since it is complex to exploit a blockchain, the testing of blockchain becomes even more complex. Since large transactions go through processes like encryption and decryption, it becomes necessary that these processes go smoothly.

### **7.2 Types of Testing**

#### **1. Unit Testing:**

- Unit testing involves testing individual units or components of the software in isolation.

- Developers typically write unit tests to verify the behavior of functions, methods, or classes.
- Tools like JUnit (for Java) or pytest (for Python) are commonly used for unit testing.

## **2. Integration Testing:**

- Integration testing verifies that individual units or components work together as intended.
- It focuses on interactions between different parts of the system.
- Integration tests are used to ensure that the integrated components interact correctly and exchange data properly.

## **3. System Testing:**

- System testing evaluates the behavior of the entire system as a whole.
- It verifies that the software meets the specified requirements and functions correctly in the intended environment.

## **4. Acceptance Testing:**

- Acceptance testing involves validating that the software meets the user's requirements and expectations.
- It may include user acceptance testing (UAT), where end-users or stakeholders verify the software's functionality in a real-world scenario.

## **5. Regression Testing:**

- Regression testing ensures that changes or updates to the software do not introduce new defects or break existing functionality.
- It involves re-running previously executed test cases to ensure that everything still works as expected.



Test Case ID	Test Scenario	Test Steps	Expected Outcome	Actual Outcome	Status (Pass/Fail)
TC_001	Verify access to /home route	1. Access /home route without login	Redirect to login page	Redirected to login page	Pass
		2. Login with valid credentials	Redirect to /home after successful login	Redirected to /home	Pass
		3. Access /home route after login	Homepage with user-specific data displayed	Homepage displayed with user data	Pass
		4. Logout from the application	Redirect to login page	Redirected to login page	Pass
TC_002	Upload new evidence	1. Access /a_upload route without login	Redirect to login page	Redirected to login page	Pass
		2. Login with valid credentials	Redirect to /a_upload after successful login	Redirected to /a_upload page	Pass
		3. Fill out evidence upload form and submit	New evidence uploaded successfully	Evidence uploaded successfully	Pass
TC_003	View and manage cases	1. Access /a_view_case route without login	Redirect to login page	Redirected to login page	Pass
		2. Login with valid credentials	Redirect to /a_view_case after successful login	Redirected to /a_view_case page	Pass
		3. View existing cases and perform actions	Cases displayed with options to manage and view details	Cases displayed with expected options	Pass
TC_004	Delete a case	1. Access /view_case route without login	Redirect to login page	Redirected to login page	Pass
		2. Login with valid credentials	Redirect to /view_case after successful login	Redirected to /view_case page	Pass
		3. Delete a specific case	Case deleted successfully	Case deleted successfully	Pass
TC_005	Request access to evidence	1. Access /a_req route without login	Redirect to login page	Redirected to login page	Pass
		2. Login with valid credentials	Redirect to /a_req after successful login	Redirected to /a_req page	Pass
		3. Submit a request for evidence access	Request submitted successfully	Request submitted successfully	Pass
TC_006	Verify download functionality	1. Access /down route without valid evidence ID	Redirect to error page	Redirected to error page	Pass
		2. Access /down route with valid evidence ID	Download evidence file	Evidence file downloaded successfully	Pass

## 7.1 Testing Results

## **CHAPTER 8**

### **CONCLUSION AND FUTURE ENHANCEMENT**

#### **8.1 Conclusion**

In today's ever-growing digital world, we are facing huge challenges in securing our digital infrastructures against different types of cybersecurity incidents. The goal of digital forensics is to perform a structured investigation and maintain a documented chain of evidence to find out exactly what happened on a digital infrastructure network or computing devices involved and who was responsible for it to mitigate and halt such cyber incidents. In conclusion, this project developed a FB-CoC model and a platform to secure Multimedia Forensic Digital Evidence (MFDE) and to ensure the forensic soundness of the stored evidence. With an DB-CoC an investigator does not need to be concerned about verification and authenticity of evidence when performing a digital investigation.

#### **8.2Future Enhancement**

##### **Enhanced User Experience and Accessibility:**

- Focus on improving the user experience and accessibility of Secure Chain by developing user-friendly interfaces, mobile applications, and browser extensions, and incorporating features such as voice recognition or natural language processing for seamless interaction.

##### **Cross-Platform Compatibility:**

- Ensure cross-platform compatibility and support for Secure Chain across different operating systems, devices, and browsers, enabling users to access and utilize the system effectively from anywhere, anytime.

##### **Community Engagement and Collaboration:**

- Foster a vibrant user community and establish partnerships with industry stakeholders, law enforcement agencies, legal professionals, academia, and cybersecurity experts to gather feedback, share best practices, and drive innovation in digital evidence management and forensic analysis.

## **APPENDIX-1**

### **CODING**

#### **Python code**

#### **Package**

```
from flask import Flask, render_template, Response, redirect, request, session, abort,
url_for
import os
import base64
from datetime import date
import shutil
import hashlib
import cv2
import imagehash
import mysql.connector
```

#### **Login**

```
def login():

cnt=0
act=""
msg=""

if request.method == 'POST':

username1 = request.form['uname']
password1 = request.form['pass']
mycursor = mydb.cursor()
mycursor.execute("SELECT count(*) FROM coc_login where username=%s &&
password=%s",(username1,password1))
myresult = mycursor.fetchone()[0]
if myresult>0:
```

```
session['username'] = username1
return redirect(url_for('admin'))
else:
msg="You are logged in fail!!!"
```

### **Case Register**

```
if request.method=='POST':
district=request.form['district']
station=request.form['station']
title=request.form['title']
cdate=request.form['cdate']
details=request.form['details']
suspect=request.form['suspect']
name=request.form['name']
fname=request.form['fname']
gender=request.form['gender']
dob=request.form['dob']
address=request.form['address']
district2=request.form['district2']
pincode=request.form['pincode']
mobile=request.form['mobile']
email=request.form['email']
aadhar=request.form['aadhar']
mycursor.execute("SELECT max(id)+1 FROM coc_case")
maxid = mycursor.fetchone()[0]
if maxid is None:
maxid=1

now = date.today() #datetime.datetime.now()
rdate=now.strftime("%d-%m-%Y")
mm=now.strftime("%m")
yy=now.strftime("%Y")
```

```

case_id="C"+mm+yy+str(maxid)
coc_case(id,case_id,district,station,title,cdate,details,suspect,name,fname,gender,dob,
b,address,district2,pincode,mobile,email,aadhar,status) VALUES

(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
val =
(maxid,case_id,district,station,title,cdate,details,suspect,name,fname,gender,dob,address,district2,pincode,mobile,email,aadhar,'0')
mycursor.execute(sql, val)
mydb.commit()
print(mycursor.rowcount, "Registered Success")
msg="success"

```

### **Upload Evidence**

```

if request.method=='POST':
    details=request.form['details']
    file=request.files['file']
    mycursor.execute("SELECT max(id)+1 FROM coc_evidence")
    maxid = mycursor.fetchone()[0]
    if maxid is None:
        maxid=1

    now = date.today() #datetime.datetime.now()
    rdate=now.strftime("%d-%m-%Y")
    if file:
        fname = file.filename

        filename = secure_filename(fname)
        efile="E"+str(maxid)+filename
        file.save(os.path.join("static/upload1", efile))
        with open("static/upload1/"+efile, "rb") as image2string:
            converted_string = base64.b64encode(image2string.read())

```

```

print(converted_string)
bfile1="E"+str(maxid)+".hash"

with open('static/upload/'+bfile1, "wb") as file:
file.write(converted_string)

mm=now.strftime("%m")

yy=now.strftime("%Y")

sql = "INSERT INTO coc_evidence(id,case_id,details,filename,upload_by)
VALUES (%s,%s,%s,%s,%s)"
val = (maxid,case_id,details,efile,'admin')
mycursor.execute(sql, val)
mydb.commit()
print(mycursor.rowcount, "Registered Success")
msg="success"
mycursor.execute('SELECT * FROM coc_evidence WHERE id=%s', (maxid,))
dd = mycursor.fetchone()
dtime=str(dd[4])

bdata="Evidence ID:"+str(maxid)+", Case ID:"+case_id+", Status: Evidence File:
"+efile+", Upload by admin, Date: "+dtime

```

### **Verification**

```

mycursor.execute('SELECT * FROM coc_evidence WHERE id=%s', (maxid,))
dt = mycursor.fetchall()
cutoff=10
for rr in dt:
hash0 = imagehash.average_hash(Image.open("static/upload1/"+rr[3]))
hash1 = imagehash.average_hash(Image.open("static/upload1/"+efile))
cc1=hash0 - hash1
print("cc="+str(cc1))
if cc1<=cutoff:
ss="ok"

```

```

pre_id=str(rr[0])
break
else:

ss="no"


if ss=="ok":

mycursor.execute('SELECT * FROM coc_evidence where id=%s',(maxid,))


    sp3 = mycursor.fetchone()
    dtime=str(sp3[4])
    mycursor.execute('SELECT * FROM coc_evidence where id=%s',(pre_id,))
    sp1 = mycursor.fetchone()
    pre_user=sp1[6]

    mycursor.execute('SELECT * FROM coc_register where uname=%s',(pre_user,))
    sp2 = mycursor.fetchone()
    pre_vid=sp2[0]

    bdata1="ID:"+str(pre_vid)+", Case ID:"+sp3[1]+", Status:Attack Found, Similar
Evidence uploaded by "+uname+", Evidence ID:"+str(maxid)+", File: "+sp3[3]+"
(Previous ID:"+str(pre_id)+"), Date:"+dtime
    msg1="attack"
def down():
    eid=request.args.get('eid')
    mycursor = mydb.cursor()
    mycursor.execute("SELECT * FROM coc_evidence where id=%s",(eid,))
    data = mycursor.fetchone()
    fn=data[3]
    ff="E"+eid+".hash"
    file = open('static/upload/'+ff, 'rb')
    byte = file.read()
    file.close()

```

```
decodeit = open('static/down/'+fn, 'wb')
decodeit.write(base64.b64decode((byte)))
decodeit.close()

path="static/down/"+fn

return send_file(path, as_attachment=True)
```



## APPENDIX-2

### SCREENSHOTS

#### HOME PAGE




This landing page serves as a gateway to access login pages designed for specific user groups, including administrators, police personnel, and authorized advocates. From here, users can seamlessly navigate to their respective login interfaces tailored to their roles and permissions. Whether you're an administrator managing system settings or an advocate accessing legal resources, this centralized platform provides secure and efficient access.

## LOGIN PAGE FOR GOVERNMENT REGULATOR

**CHAIN OF CUSTODY**

[HOME](#) [AUTHORIZED PARTIES](#) [GOVERNMENT REGULATOR](#)



Government Regulator

**Username**

**Password**

Login

Welcome to the administrators' login page, where authorized personnel can access a dedicated platform to perform essential administrative tasks. Once logged in, administrators can efficiently manage case registration, create new authorized parties, and review case evidence. Our secure interface ensures seamless navigation and robust data management, empowering administrators to oversee operations effectively. Explore our user-friendly features designed to enhance administrative workflows and optimize case management processes. Join us in simplifying administrative tasks and enhancing productivity within your organization.

## LOGIN PAGE FOR AUTHORIZED PARTIES

CHAIN OF  
CUSTODY

[HOME](#) [AUTHORIZED PARTIES](#) [GOVERNMENT REGULATOR](#)



### Authorized Parties

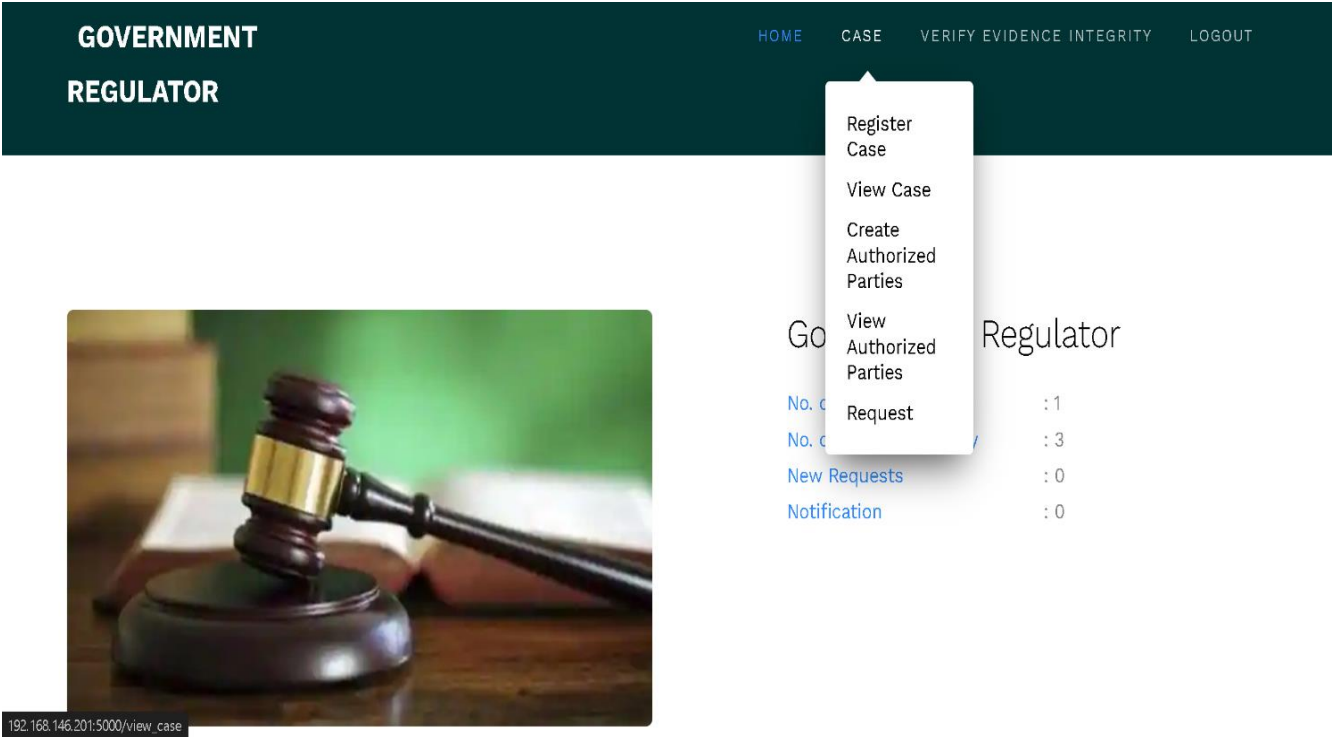
Username

Password

Login

Welcome to the login portal for authorized users. Here, you can securely access your personalized workspace to carry out tasks such as collecting, attaching, and reporting evidence promptly to the administration. Additionally, authorized users can conveniently view their assigned cases and request necessary resources from the administrators as needed. Our platform is designed to streamline workflow processes, ensuring efficient collaboration and communication between authorized personnel and administrators. Experience a seamless interface that facilitates effective case management and resource allocation. Join us in optimizing your responsibilities and contributing to a more organized and responsive system.

HOME PAGE FOR GOVERNMENT REGULATORS





Welcome to the admin home page, your central hub for managing all administrative tasks. Here, administrators can stay updated on new requests and notifications. By hovering over the case navigation, admins can access various features including viewing cases, creating new cases, managing authorized parties (creating, deleting), requesting updates for cases assigned to specific parties, and monitoring current case statistics such as the number of active cases and authorized parties. Explore our intuitive interface designed to streamline administrative workflows and provide comprehensive oversight of system operations. Join us in optimizing administrative efficiency and maintaining control over essential functions within your organization's operations.

# CASE REGISTRATION

GOVERNMENT  
REGULATOR

[HOME](#) [CASE](#) [VERIFY EVIDENCE INTEGRITY](#) [LOGOUT](#)

## Case Information

			
Case ID	: C0420244	Case ID	: C0320243
Police Station	: Karuppur, Salem	Police Station	: vangal, Karur
<hr/>		<hr/>	
Title of Complaint	: Theft	Title of Complaint	: death
Occurance Date	: 2024-04-08	Occurance Date	: 2024-03-29
Complainant's Name	: Surya R	Complainant's Name	: Umesh
Case Register Date	: 2024-04-08 12:46:15	Case Register Date	: 2024-03-29 12:41:02
<hr/>		<hr/>	
<a href="#">Full Details</a> / <a href="#">Evidence</a> / <a href="#">Permission</a> / <a href="#">Delete</a>		<a href="#">Full Details</a> / <a href="#">Evidence</a> / <a href="#">Permission</a> / <a href="#">Delete</a>	

Welcome to the case registration page designed for administrators. Here, you can access a comprehensive overview of all registered cases and explore detailed information, including evidence and permissions. Administrators have the capability to view full case details, manage evidence, and adjust permissions regarding who can access and edit specific information. Our intuitive interface empowers administrators to efficiently oversee case management, ensuring secure and controlled access to sensitive data. Experience streamlined navigation and robust controls that enhance administrative capabilities within your organization.

## REGISTRATION OF AUTHORIZED PARTIES

CHAIN OF  
CUSTODY

[HOME](#) [CASE](#) [REQUEST](#) [LOGOUT](#)



Authorized Party

[Naresh \(ID:AT3\)](#)

Mobile No.: 8945673490

Email: [naresh@gmail.com](mailto:naresh@gmail.com)

Aadhar No.: 432367458990


Location: mecheri, salem

Welcome to the home page dedicated to authorized parties such as police officers or advocates. Here, you can conveniently access your personal details including Aadhar number, phone number, email, and location. Navigate to your allocated cases to review and manage ongoing tasks efficiently. Additionally, utilize the request feature to submit resource requests as needed for seamless workflow operations. Our user-friendly interface ensures quick access to essential information and functionalities tailored to your role. Explore a centralized platform designed to optimize productivity and support effective collaboration within your professional responsibilities. Join us in enhancing your experience and leveraging technology to streamline your daily tasks.

## ACCESS PRIVILEGE OF AUTHORIZED PARTIES:

GOVERNMENT  
REGULATOR

HOMECASEVERIFY EVIDENCE INTEGRITYLOGOUT



Access Privilege

Case ID: C0320243

Evidence File: ESIMG\_8200.JPG

Authorized Party

AT1-Ramkumar

AT1-Ramkumar

AT2-Dharun

AT3-Naresh

Submit

Welcome to the admin page where you can assign cases to police officers or advocates with specific access permissions. Admins have the capability to manage access levels such as view-only, download-only, or view and download for assigned personnel. This centralized platform empowers administrators to allocate cases efficiently and control access to sensitive information based on role requirements. Explore seamless case assignment features designed to enhance collaboration and optimize workflow processes. Join us in leveraging advanced access management tools to ensure secure and controlled data sharing within your organization's operations.

# BLOCKCHAIN TOOL

Blockchain

JSON Data

[Decrypt Block](#)

Block ID	: 1
Data	: 5d41402abc4b2a76b9719d911017c592
Block ID	: 2
Data	: b345c5981843a83155e9ea4ea6ca6e
Block ID	: 3
Data	: 252441f35b605de8663fdd3a13d9bee4
Block ID	: 4
Data	: 85130b2bc96535b8e509fc134e8cd9d1
Block ID	: 5
Data	: 47e1ead81081b23285d0fed35010d5c1
Block ID	: 6
Data	: 35d5f1c245c2dfe9d78c9fde27c9f046
Block ID	: 7
Data	: c79d87b48eb7479078ebadd89f409520
Block ID	: 8
Data	: 95ebe10e08ecd5c54c9265c174bfa4

Welcome to the evidence viewing page, where all evidence and details are securely encrypted using blockchain technology. Administrators can access and decrypt evidence using the appropriate encryption key. This platform allows admins to view encrypted evidence associated with specific cases and filter evidence based on case details. Explore advanced encryption features that ensure data security and integrity, providing a reliable method for accessing critical information. Join us in leveraging cutting-edge blockchain technology to maintain confidentiality and enhance evidence management within your organization's operations.



# TAMPER DETECTION

## BLOCKCHAIN

### JSON Data

Block ID	: 6
Data	: Access ID:2, Case ID:C0120241, Status:Access for View and Download, User:AT1, Date: 2024-01-28 09:48:29
Block ID	: 7
Data	: Access ID:3, Case ID:C0120241, Status:Access for View and Download, User:AT2, Date: 2024-01-28 09:53:29
Block ID	: 8
Data	: Allow ID:2, Case ID:C0120241, Status:Allowed for View and Upload, User:AT2, Date: 2024-01-28 10:02:00
Block ID	: 9
Data	: Evidence ID:3, Case ID:C0120241, Status: Evidence File: E3evidence_1.png, upload by AT2, Date: 2024-01-28 10:03:32
Block ID	: 10
Data	: ID:2, Case ID:C0120241, Status:Attack Found, Similar Evidence uploaded by AT2, Evidence ID:3, File: E3evidence_1.png (Previous ID:3), Date:2024-01-28 10:03:32
Block ID	: 11
Data	: ID:2, Case ID:C0120241, Status:Attack Found, Similar Evidence uploaded by AT2, Evidence ID:4, File: E4Evidence_1_new.png (Previous ID:4), Date:2024-01-28 10:03:53
Block ID	: 12
Data	: Evidence ID:4, Case ID:C0120241, Status: Evidence File: E4Evidence_1_new.png, upload by AT2, Date: 2024-01-28 10:03:53
Block ID	: 13

Welcome to the evidence integrity verification page. Here, you can identify if evidence has been intentionally deleted, changed, or re-uploaded by authorized parties. Modified or re-uploaded data will be displayed in black color, allowing administrators to quickly identify discrepancies or missing/duplicated evidence. With blockchain technology, uploaded evidence cannot be deleted or removed; instead, new evidence can be uploaded, ensuring a transparent and immutable record of changes. Explore our secure platform designed to maintain evidence integrity and facilitate accurate case management. Join us in leveraging blockchain to enhance data reliability and trustworthiness within your organization's operations.

## REFERENCES

1. D. Li, W. Liu, L. Deng, and B. Qin, "Design of multimedia blockchain privacy protection system based on distributed trusted communication," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 2, p. e3938, Feb. 2021.
2. M. Uddin, "Blockchain medledger: Hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry," *Int. J. Pharmaceutics*, vol. 597, Mar. 2021, Art. no. 120235.
3. M. Li, C. Lal, M. Conti, and D. Hu, "LEChain: A blockchain-based lawful evidence management scheme for digital forensics," *Future Gener. Comput. Syst.*, vol. 115, pp. 406-420, Feb. 2021.
4. M. R. Kumar and N. Bhalaji, "Blockchain based chameleon hashing technique for privacy preservation in E-governance system," *Wireless Pers. Commun.*, vol. 117, no. 2, pp. 1-20, 2020.
5. M. Luseti, L. Salsi, and A. Dallatana, "A blockchain based solution for the custody of digital files in forensic medicine," *Forensic Sci. Int., Digit. Invest.*, vol. 35, Dec. 2020, Art. no. 301017.
6. J. Jeong, D. Kim, B. Lee, and Y. Son, "Design and implementation of a digital evidence management model based on Hyperledger fabric," *J. Inf. Process. Syst.*, vol. 16, no. 4, pp. 760-773, 2020.
7. L. Zarpala and F. Casino, "A blockchain-based forensic model for financial crime investigation: The embezzlement scenario," 2020, arXiv:2008.07958. [Online]. Available: <https://arxiv.org/abs/2008.07958>.
8. H. R. Hasan, K. Salah, R. Jayaraman, M. Omar, I. Yaqoob, S. Pesic, T. Taylor, and D. Boscovic, "A blockchain-based approach for the creation of digital twins," *IEEE Access*, vol. 8, pp. 34113-34126, 2020.
9. Z. Tian, M. Li, M. Qiu, Y. Sun, and S. Su, "Block-DEF: A secure digital evidence framework using blockchain," *Inf. Sci.*, vol. 491, pp. 151-165, Apr. 2019.

- 10.E. Yunianto, Y. Prayudi, and B. Sugiantoro, ``B-DEC: Digital evidence cabinet based on blockchain for evidence management," *Int. J. Comput. Appl.*, vol. 181, no. 45, pp. 22-29, Mar. 2019.
- 11.M. Takemiya and B. Vanieiev, ``Sora identity: Secure, digital identity on the blockchain," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2018, pp. 582-587.
- 12.K. Widatama, Y. Prayudi, and B. Sugiantoro, ``Application of RC4 cryptography method to support XML security on digital chain of custody data storage," *Int. J. Cyber-Secur. Digit. Forensics*, vol. 7, no. 3, pp. 230-237, 2018.
- 13.M. Shah, S. Saleem, and R. Zulqarnain, ``Protecting digital evidence integrity and preserving chain of custody," *J. Digit. Forensics, Secur. Law*, vol. 12, no. 2, pp. 121-130, Jun. 2017.
- 14.B. Bayar and M. C. Stamm, ``Design principles of convolutional neural networks for multimedia forensics," *Electron. Imag.*, vol. 2017, no. 7, pp. 77-86, Jan. 2017.
- 15.J. Patel and N. Bhatt, ``Review of digital image forgery detection," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 5, no. 7, pp. 152-155, Jul. 2017.
- 16.S. Battiato, O. Giudice, and A. Paratore, ``Multimedia forensics: Discovering the history of multimedia contents," in *Proc. 17th Int. Conf. Comput. Syst. Technol.*, Jun. 2016, pp. 5-16.
- 17.Y. Prayudi and A. Sn, ``Digital chain of custody: State of the art," *Int. J. Comput. Appl.*, vol. 114, no. 5, pp. 1-9, Mar. 2015.