Blockchain and Android: A data integrity preservation method

MSc Practicum Part 2
MSc Cybersecurity

Anjana Unnikrishnan
Student ID: 23124865

School of Computing
National College of Ireland

Supervisor: Jawad Salahuddin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Anjana Unnikrishnan		
Student ID:	23124865		
Programme:	Master of Science in Cybersecurity	**Year:**	2024
Module:	MSc Practicum part 2		
Supervisor:	Jawad Salahuddin		
Submission Due Date:	12/12/2024		
Project Title:	Blockchain and Android: A data integrity preservation method		
Word Count:	5952 **Page Count** 17		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Anjana Unnikrishnan

Date: 28/01/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Blockchain and Android: A data integrity preservation method

Anjana Unnikrishnan
23124865

Abstract

Data integrity preservation is a tiring but critical task in applications, especially in Android applications. This study focuses on integration of blockchain technology with traditional database as a dual data storage and incorporating it with strong hashing technique, the SHA-256. This can bring a promising development in ensuring tamper-proof applications across various domains like healthcare, supply chain management and finance. Various experimental case studies were conducted to evaluate the efficiency of the system which gave a better response time of 72ms, for retrieving user data from dual storage, calculating hash of user records and comparing the hash to evaluate the integrity of the data. The research opens a new pathway for combining traditional systems with modern data systems like blockchain to bring a revolution in the cybersecurity domain.

1 Introduction

Smartphones have become an integral part of people's lives. People rely on smartphones from setting an alarm to wake up in the morning to manage their personal identifiable information. As per the user stats by Howarth (2024), about 70% of the global market share is owned by android. There are over three billion active users for android as per Curry (2024). But there arises a question of how secure android is to hold and process user data. While installing an application and enter personal details into it, nobody thinks of where this information is stored or is the data integrity is preserved. Furthermore, it requests permissions more than necessary which raises a security concern among users. But most users grant these permissions for the application to work properly. If there was a stricter way to control the user data that the service providers take from applications, the android applications become more secure. This gave the motivation to research deep on data integrity issues in android and try to employ Blockchain technology to preserve the user data integrity. Researching through various conference papers, journals, articles and eBooks, gave a deep understanding about how dangerous android applications data storage and the attacks that can occur when overprivileged. To ensure security in such applications, latest and emerging technology like blockchain can be a suitable option. As the technology is in developing stage, most of the research materials have their own limitation. Thus, it's better to store a hashed or checksum or encrypted unique value against each user's data, which on tampering will easily be detected.

The research question identified to this problem is: Evaluate the use of blockchain technology to preserve data integrity in Android mobile applications. Study of various solutions to address the data integrity issue in applications and introduction of blockchain technology to store the sensitive user data and performing SHA-256 hash comparison of data to ensure the integrity of data is the proposed solution for this problem. To achieve the same, a combination of traditional database storage and innovative, trusted blockchain storage is employed to build a feasible and secure system for this use case. A smart contract is developed to store and retrieve the sensitive data from blockchain platform. The transactions will be securely stored in a distributed ledger.

The contributions in this document include,

- Related Works (Section 2): Various conference papers, industry journals and academic articles were studied and used for research as part of this project and identified the advantages and disadvantages of methodologies used by them. Also, identified key points from each research material while researching the android permission analysis part.
- Research Methodology (Section 3): This section outlines the methodology followed by the proposed system.
- Design Specification (Section 4): This section details the key design considerations for the development of the proposed solution.
- Implementation (Section 5): This section explains the complete functioning of the system.
- Evaluation (Section 6): This section includes different case studies and discussions to evaluate the robustness of data integrity preservation model.
- Conclusion and Future Work (Section 7): Concludes the proposed research with areas that can expanded as part of the future work.

2 Related Work

Data stored in any system is vulnerable to invasion or alteration. In case of android platforms, data stored in a database server can have an additional layer of security to detect data corruption or any signs of failure of data integrity, which is discussed and implemented in this research. For the identifying the best choices, development and analysis of the method implemented, extensive research was undertaken with the help of peer-reviewed research papers, journals, articles, technical reports and books. A review of related works and relevant materials are discussed in this section:

2.1 Analysing the usage of blockchain technology in android applications

Implementing blockchain technology in android applications can bring a huge advantage to user data security. In a survey on security of blockchain-based data storage in android applications by Musa et al., the authors discuss the challenges related to data storage costs, its scalability and performance, and suggested a Blockchain-based Secure Android Data Storage (BSADS) framework with six layers. They argue that the decentralization and tamper-resistance power of blockchain technology can help in securing data in android application. Additionally, they have also pointed out the scalability and cost issues that could arise while integration of blockchain. The major takeaway from this journal is that, while implementing blockchain technology in mobile applications we should utilize lightweight nodes instead of full nodes which can be resource intensive causing inadequate user experience and large data consumption.

Another article published by Ichikawa et al., 2017 has developed a tamper-resistant mHealth application using blockchain technology by storing the data in Hyperledger fabric. They used consensus algorithm on node for the distributed database to defend against if the node goes down. The limitation identified in this implementation is that the data is directly stored in a Hyperledger fabric, which is not a secure even if it is in a private blockchain. The tamper resistance feature which needs to be improved, and the key findings (selection of lightweight node for mobile applications) are valuable for developing this practicum project.

This project is developed as an improvement to a conference paper published by Khiem et al., 2023 which proposes the application of blockchain technology for privacy preservation in android platforms. The approach detailed in the paper can be summarized as: a user installs a medical application, and if the application requests sensitive data that which is acquired through dangerously categorized permissions (BODY_SENSORS; BODY_SENSORS_BACKGROUND) in android, end-user is presented with a request with all the information regarding their data usage. This request is validated through a smart contract and gives more control to the user over their data. If the user gives consent, the data is encrypted with the public key of the service provider using elliptic curve cryptography and transmitted as a Non-Fungible Token (NFT) encapsulated block leaving an immutable trace of transaction in blockchain, which is depicted in Figure 1.

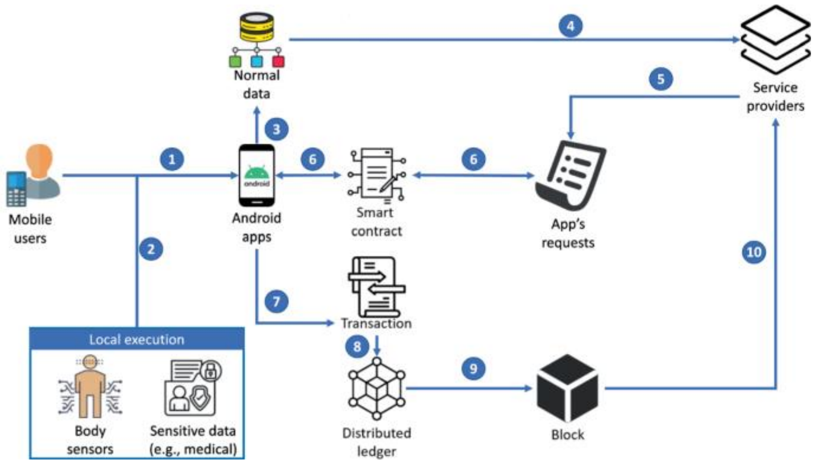


Figure 1: Blockchain integration in Android Systems

In the research paper by Le et al., 2023, they have developed a proof-of-concept where blockchain technology is leveraged to transform the access, utilization and protection of user data. The proposed model integrates blockchain with permission management in Android systems giving users control over their important data. They have also evaluated the implementation in different blockchain platforms to conclude that Fantom is an optimal choice due to its low transaction and gas costs. This paper is more focused on an android user perspective and their implementation will be dependent on service providers and Android OS producers. Considering the developers perspective, there should be a more secure system than just saving the data in a blockchain.

Another research by Lakhan et al., 2023 presents a new ransomware blockchain efficient framework (RBEF) for digital networks, which can identify the ransomware attacks in transactions. The project is based on Kotlin, Java, Android and other socket programming. This method helps in processing and validating data among healthcare blockchain nodes. They were able to reduce the execution delays and processing costs while performing healthcare transaction in a secure manner. This paper suggests an effective methodology in ransomware detection in blockchain transactions, but how effective or secure is saving data in blockchain alone.

In the research work by Pal et al., 2022 the authors have designed a multi app spy system based on blockchain technology to monitor the permission or resource requested by android applications and compare it with what is granted by the user. This would help the users to identify if the applications are performing any unauthorized requests to record them in a blockchain database. In their proposed method, the have a hash table with android application permissions and every permission request from an application will be validated against this table and if validated the request will be sent to android kernel, if not the request will be prevented from reaching kernel and will be stored as a transaction in blockchain system. This helps in keeping the users secure from malicious applications and keep a record of the same which will be quite useful in developing defensive systems for android mobile applications. This paper highlights the usage of blockchain technology in securing android devices or users from dangerous applications and malicious actors which is quite helpful in deciding to use blockchain technology in this project.

In the present world, cryptocurrency has taken a notable step and is found in every domain, so does in android. A research paper by Li et al., 2020, they have evaluated security threats possessed by android-based cryptocurrency wallets and some defense strategies against them. Their study proves that different functionalities like backing up files, accessibility services and android clipboard can steal information from famous cryptocurrency wallet applications in google play store indicating how design flaws during development of android application can be dangerous. The defensive mechanisms can be adopted widely, which are quite easy but good enough to prevent data leaks.

2.2 Blockchain Technology: Application, Development and Integration

Data is one of the most important assets, and ensuring the integrity of data has become complex. Introducing emerging technologies to secure data can have an effective impact. In the article by Vacca et al., 2020, the authors have conducted a literature review of challenges in developing and maintaining smart contracts or decentralized applications (DApps) in blockchain-oriented software. This article overviews: smart contract security, code analysis and so on, helping in the development of this project. The key takeaway from this article is that the Hyperledger Fabric, which is a permissioned blockchain framework, in which the chaincode (smart contract) can be developed in Go, Java, JavaScript or Typescript, but the developer must address all the loopholes these languages can create as they were not designed to create smart contracts.

In a research paper by Thakkar et al., 2018, the authors have detailed the evaluation of performance by Hyperledger Fabric blockchain platform. They have identified the key bottlenecks like endorsement policy verification, sequential policy validation of transactions and state validation and commit could adversely affect the performance of this blockchain platform. Their study reveals that Hyperledger Fabric achieves higher throughput and lower latency as compared to Ethereum, which is why Hyperledger Fabric is the top choice for the development of this project.

After reviewing various research papers, articles and journals, an implementation plan and resource confirmations, evaluation strategies were finalized. To secure the integrity of data, the data can be stored in a traditional database and blockchain platforms, and the hash of records stored is calculated and compared for both platforms to ensure no tampering has occurred. The evaluation of transaction speed and overall efficiency can determine the effectiveness of the system.

3 Research Methodology

Data integrity is a well-known critical issue in most of the systems, specifically in Android systems. This research study focuses on a solution to detect the tampered data in traditional databases and help to retrieve the original data from a secure source like blockchain network. Thus, the research proposes a method to preserve the integrity of data by integrating blockchain networks with traditional databases, specifically for Android systems. This methodology is developed by performing extensive literature reviews of various research papers, journals, articles and online open-source repositories. Based on the insights from the literatures review, various decisions like using Hyperledger fabric as a blockchain platform for the development (research paper by Thakkar et al., 2018) of this project and secure smart contract development using JavaScript (article by Vacca et al., 2020) were chosen.

The research procedure involves two major steps – setting up a blockchain platform and traditional database, developing a client application to interact with these. In this research, the following considerations are taken:

- An android application has a feature of collecting sensitive user information like, name, date of birth, email address and PPSN.
- The data is stored in a traditional database, PostgreSQL in this case.
- The data storage and retrieval are performed with the help of application programming interface (API) calls.

The detailed steps involved in the research procedure is as follows:

3.1 Base Platform Setup

- **Hyperledger Fabric Network –** Hyperledger fabric is an open-source blockchain platform that provides a high degree of confidentiality, scalability and flexibility. A Hyperledger fabric network was set up with two organizations with peers, that run the chaincode containers to perform all the operations in the ledger. A smart contract to be used by the client application to interact with the fabric network was developed in JavaScript language. The smart contract helps the client application to store and retrieve the user data stored in the ledger. A channel shared across the authenticated peers was set up for making secure blockchain transactions.
- **PostgreSQL** – This was chosen as the primary database for the storage of user data. PostgreSQL is one of the widely used relational database management system (RDBMS) in the industries due to its ability to handle huge amount of data and support for complex data structures. The PostgreSQL database was set up in WSL with the help of VS Code and its extensions like PostgreSQL explorer, SQLTools etc. As per industry standards, migrations were utilized to create the table and its attributes to enable the flexibility of using this functionality in any system.
- **JavaScript Client Application** – A server was built with two APIs for Android application communication. Node.js and Express.js were the main libraries used for the application development. Postman was leveraged for API testing.
- **Windows Docker Desktop and WSL –** The network setup of Hyperledger fabric, the database PostgreSQL and the JavaScript client application were setup in the windows subsystem for Linux (WSL2) with the help of docker containers (built using Windows Docker Desktop).

The tools, libraries and platforms used for the development of this project is summarized in the below Table 1.

Table 1: Tools, Libraries and Platforms Used

Tool/Platform/Library	Version	Description
WSL (Windows Subsystem for Linux) - Ubuntu	2.3.26.0	To get access to Linux libraries in Windows to setup the Hyperledger fabric network.
Docker Desktop	4.34.2	Platform to build, run, and manage Hyperledger fabric application on Windows.
Node.js	18.20.5	JavaScript runtime for building and running server-side application.
Hyperledger Fabric	2.5.10	Blockchain framework used for storing and retrieving user data.
Logspout	2.5.10	A tool for monitoring Hyperledger fabric Docker containers, particularly using logs.
PostgreSQL	14.13	Traditional database management system for storing user data.
Host Machine	Windows 11	The host machine in which entire project is configured.
VSCode	1.95.3	VSCode is the integrated development environment (IDE) used for the project/

3.2 Project Working

The detailed working of the entire system is explained below:

1. WSL and windows docker desktop were installed and configured in the windows host machine.
2. Node.js, python3 were also installed as a prerequisite for the project.
3. The Hyperledger fabric samples was cloned with the help of a well-structured and detailed documentation from their official website (Hyperledger Fabric, 2020-2024), which includes Hyperledger fabric docker images, CLI tool binaries and config files which helps to interact with the test network.
4. After successful cloning of the repository, required tools and files, the test-network was started, creating a channel with two channel members namely Org1 and Org2. This also allows peers of each organization to join the channel.
5. Logspout was setup for monitoring the smart contract and any docker containers running within the test network.
6. Deployment of the smart contract was the major step in setting up a Hyperledger fabric network. The following steps were performed for deploying a smart contract in JavaScript to store and retrieve the user's data in blockchain network:
    a. In the JavaScript specific folder of asset-transfer (basic) chaincode, there was a default smart contract for asset transfer in 'libs/assetTansfer.js'. This smart contract was modified for the requirement of this project, which is user records insertion and retrieval.

    b. The dependencies required, i.e., node_modules for the smart contract are installed in the ‘asset-transfer-basic/chaincode-javascript’ folder.
    c. The chaincode was packaged using a Hyperledger fabric peer CLI, which was installed on the peers of each organization. This installed chaincode needed approval by the organizations and after approval the chaincode was committed to the communication channel. The chaincode was prepared to be invoked by client application.

7. The next step was to configure the PostgreSQL database. A database, table and attributes of the table were created and maintained using migration to ensure consistency across all environments and reduce the errors.
8. A JavaScript application was designed to connect Android application to the Hyperledger fabric network and PostgreSQL.
    a. The user data storage request from an android application was received as an API request, which was then stored into PostgreSQL and blockchain network.
    b. And when the android application requested a user data, the data was retrieved from traditional database and blockchain. The SHA-256 hash of the user records from both storage systems was compared to ensure the integrity of the data, i.e. the data was not corrupted.

This methodology helps the stakeholders for the application to identify data corruptness and recover the user data from a secure and reliable storage like blockchain. Additionally, this can help the users to identify if their data is secure in the android application, making the control over their data more transparent.

4 Design Specification

The architectural framework and design considerations for the implementation of the proposed data integrity solution is included in this section.

4.1 Architecture

The proposed data integrity model architecture is depicted using two layers of data storage, a blockchain based storage and a traditional relational database storage. The blockchain based storage is a Hyperledger fabric-based network setup on a docker container with peer nodes. This layer ensures the data immutability and secure storage of data even if the data in traditional data storage is compromised. The smart contract is the key in enforcing the data integrity rules and helps in a secure data storage and retrieval functionalities. The second layer is the traditional database layer, which is a PostgreSQL database in this case as the primary storage for user records. The JavaScript backend application serves as a bridge between the Android applications and double layered storage systems for flexible user data storage and retrieval.

The first phase of the project is the user data storage. The Android application collects the sensitive user information like date of birth, personal public service number (PPSN). The user data was encoded in the body of an API request which is send to the Node.js server. The Node.js was configured with a logic to send the data to a primary database storage and blockchain storage. For storing the data in PostgreSQL, a table was created with required attributes using migration which ensures the adaptability of the functionality. Following which a connection between the server and database system were established, and the query to store users in the specified table was executed. For storing data in a Hyperledger fabric network, a secure connection was established with a new identity and SHA-256 hash signing between the server and fabric network. Then a transaction for submitting the user data was initiated. The data will be stored in the blockchain network spread across nodes making it extremely difficult for hackers to manipulate. The visualization of the data storage process is depicted in the Figure 2.

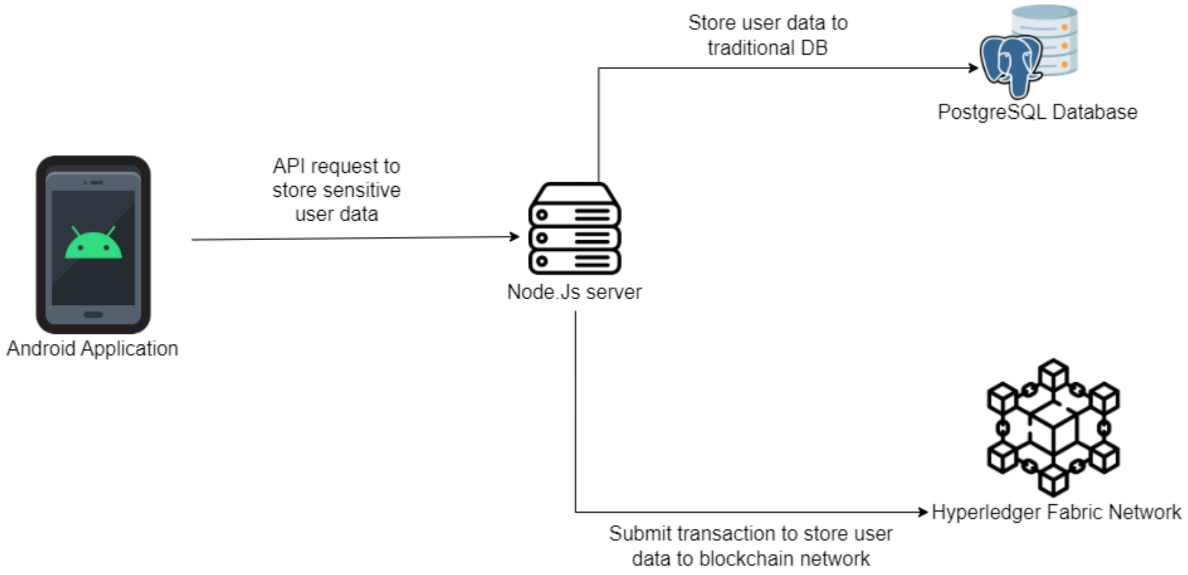

Store user data to traditional DB
PostgreSQL Database
API request to store sensitive user data
Android Application
Node.Js server
Hyperledger Fabric Network
Submit transaction to store user data to blockchain network

Figure 2: The storage of user data

The second phase is the user data retrieval along with the data integrity check. When a data is requested by the application, it sends an API request with required user email to the Node.js server. The server handles the logic to extract the data from both the storage systems. For retrieving the user data from PostgreSQL, a query is executed. Now, to retrieve the data from blockchain network, the async function in the smart contract was invoked, which basically calls transaction evaluation. If the user is found, details would be retrieved from fabric network. Now, to verify the integrity of the user data from primary storage the hash of data from both storage systems is compared. The SHA-256 hash of the user records from PostgreSQL and fabric network were calculated. This hash is then compared to evaluate the integrity of the data. If there is a corruption in the data stored in primary data storage, the hash value calculated might mismatch and reports as corrupted. Otherwise, the retrieved data can be returned to the android application. The visualization of the data retrieval process with integrity check is depicted in the Figure 3.

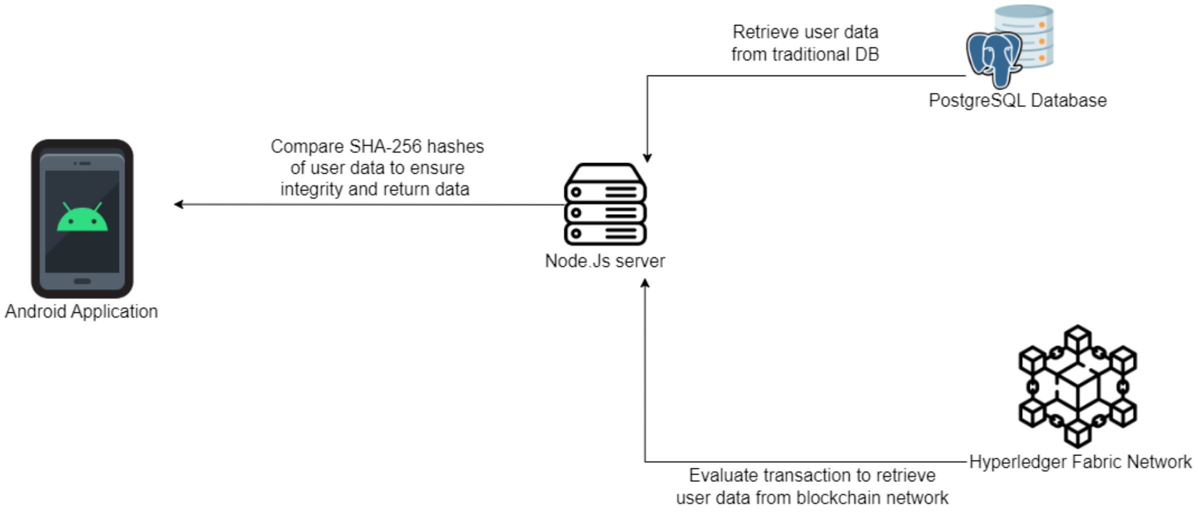


Figure 3: The user data retrieval and integrity check

4.2 Algorithm of the Model Proposed

The algorithm of the proposed model includes the following steps:

1. A smart contract with user data storage and retrieval is prepared and deployed in the Hyperledger fabric network.
2. When an android application requests, data storage, the user input is collected and
    a. Stored in PostgreSQL (traditional database)
        i. Query to create new record in the “users” table is executed.
    b. Blockchain storage system (Hyperledger fabric network)
        i. A gRPC connection between server and fabric network is established.
        ii. Fabric network and smart contract instances are initialized following the connection.
        iii. A transaction is submitted to the blockchain network by invoking the smart contract async function “createOrUpdateUser(ctx, email, name, dob, ppsn)” with the user details as the parameter.
3. When the android application requests, data retrieval, the user email of the required user is passed to the server and
    a. Query to retrieve user from PostgreSQL is executed.
    b. A evaluate transaction method is called with smart contract function “getUser(ctx, email)” called with required user email that returns user data.
4. In the last step, the data record retrieved from both the storage systems hashed using SHA-256 hash and compared.
    a. If both the hashes are equal – data in primary database is safe and returned to android application.
    b. If the hashes mismatch – data is corrupted in the primary database.

5 Implementation

The implementation of the proposed solution involves the development of a data integrity validation method integrating of two storage systems namely, blockchain and relational database. The process includes configuration of infrastructure – windows docker desktop, WSL, VSCode, Hyperledger fabric network and PostgreSQL DB setup, development of smart contract, implementation of Node.js server for interaction between android applications and dual-storage systems. The tools and languages leveraged for the development of the solution are detailed below:

Tools

- Hyperledger fabric – To setup blockchain network and storage
- PostgreSQL – For primary data storage
- Node.js and Express,js – For backend and API development
- VSCode – IDE for whole development operations
- Postman and SQLTools Extensions – To test APIs and SQL query in PostgreSQL
- Logspout – To track and monitor fabric network containers

Languages

- JavaScript – To develop smart contract, Node.js server and data integrity checking logic
- SQL – To handle database schema designing and query execution

A Hyperledger fabric network with two peer nodes and one ordering node for two organizations were setup. A smart contract to handle secure data storage and data retrieval in fabric network was developed in JavaScript language, which was then deployed in the fabric network with the approval of two organizations. This smart contract serves as a middleman between Node.js server and fabric storage. For storing data in a Hyperledger fabric network, a new secure gRPC connection is established with a new identity and SHA-256 hash signing between the server and fabric network. Then the instance for smart contract is declared to initiate the transaction for submitting the user data using an async function "createOrUpdateUser(ctx, email, name, dob, ppsn)". The PostgreSQL database storage "blockchain_app" was created. The database schema was designed using migration to store user records "name", "email", "dob" and "ppsn" in the table "users". To retrieve the data from blockchain network, the async function in the smart contract "getUser(ctx, email)"was invoked, which basically does a transaction evaluation. The fabric network if the user is found would be retrieved. The Node.js backend application was developed to operate the communication between android applications and dual-layered storage system, handle data integrity checks. The "calculateHash" in the Node.js application calculates the SHA-256 hash of the user records from PostgreSQL and fabric network. Now, to verify the integrity of the user data from primary storage the hash of data from both storage systems is compared. The key functions of the server include handle API requests from android applications, store and retrieve user records in and out from both storage systems and calculate and compare SHA-256 hashes of records for data integrity validation. The RESTful APIs developed using Express.js ensured smooth communication between android and Node.js server. The endpoints and parameters include:

1. To store users to PostgreSQL and fabric network

a. API - http://localhost:3000/users
b. JSON body –
```
{
 "name": "John D",
  "email": "user16@example.com",
  "dob": "1990-01-01",
  "ppsn": "12345PP"
}
```

2. To retrieve user data from PostgreSQL and fabric network
    a. API - http://localhost:3000/users/:id
    b. Params – Key:id, value: user16@example.com

The user data records retrieved from relational database and blockchain network were processed and SHA-256 hashes for both records were calculated. The calculated records were validated to ensure the integrity and reliability of the data.

6 Evaluation

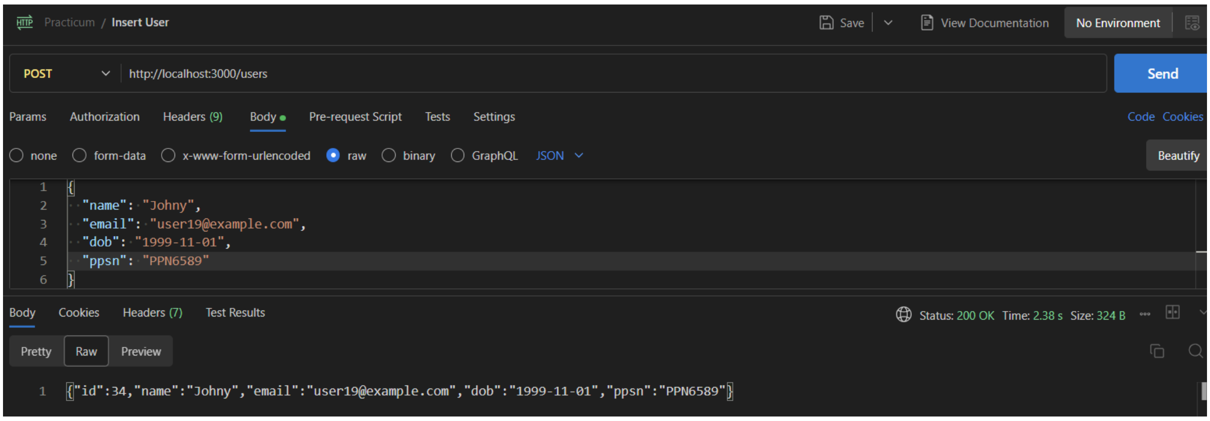
The implemented solution was evaluated through a set of test scenarios constituting normal operations, simulated data corruption, and both user storage and retrieval with hash comparison functionality. The test scenarios were designed to test the data integrity validation capabilities, the time taken for each transaction and query execution and scalability.

6.1 Experiment / Case Study 1: User Data Storage

The system was tested for user storage functionality. The test scenario and outputs obtained for this case study are as follows:

Scenario – An API request from an android application with user details passed as request body was triggered. The data was inserted into PostgreSQL and Hyperledger fabric network simultaneously.

Output – The data was successfully inserted into both the storage systems in 2.20s. This was expected as there were many complex operations performed - established a gRPC connection between the server and Hyperledger fabric network, invoked a smart contract function as well as inserted data to relational and blockchain storage.



POST http://localhost:3000/users

JSON

```
{
  "name": "Johny",
  "email": "user19@example.com",
  "dob": "1999-11-01",
  "ppsn": "PPN6589"
}
```

Status: 200 OK Time: 2.38 s Size: 324 B

```
{"id":34,"name":"Johny","email":"user19@example.com","dob":"1999-11-01","ppsn":"PPN6589"}
```

Figure 4: User data storage

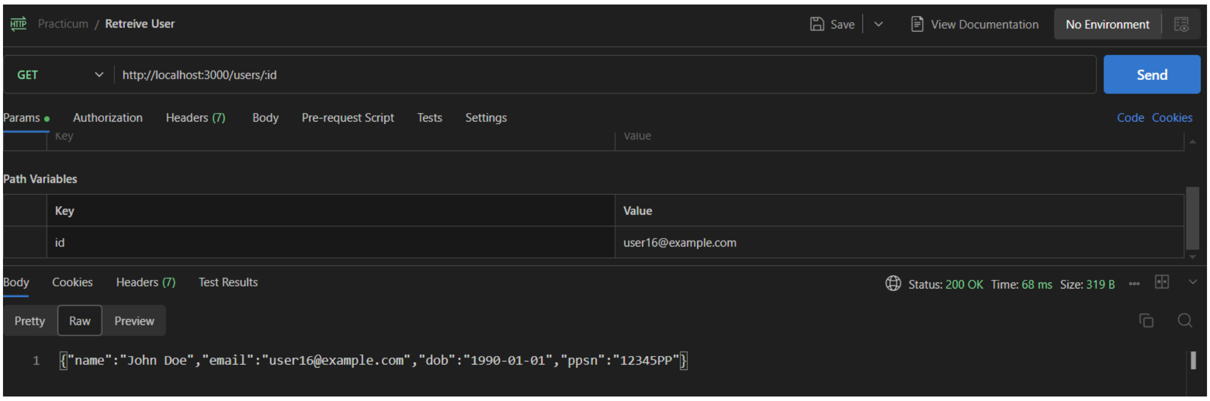
6.2 Experiment / Case Study 2: Normal Operation

The system was tested under normal conditions. The test scenario and outputs obtained for this case study are as follows:

Scenario – A user record was collected in an android application and send to server for storage in an API request as body. The data was stored successfully in PostgreSQL and fabric network (by invoking smart contract function). Now, the data was retrieved from both storage systems and the SHA-256 hash of both records were calculated and compared for integrity check.

Output – The user data was reliable, and integrity was verified. The overall time taken for the retrieval of a user record without any corruption was 68ms, which is faster as a normal API request. There were three processes happened during the API call:

- The user data was retrieved from PostgreSQL and fabric storage.
- The SHA-256 for the user records from each storage was calculated.
- The calculated hashes were compared to ensure data integrity.



GET http://localhost:3000/users/:id

Path Variables

Key	Value
id	user16@example.com

Status: 200 OK Time: 68 ms Size: 319 B

```
{"name":"John Doe","email":"user16@example.com","dob":"1990-01-01","ppsn":"12345PP"}
```

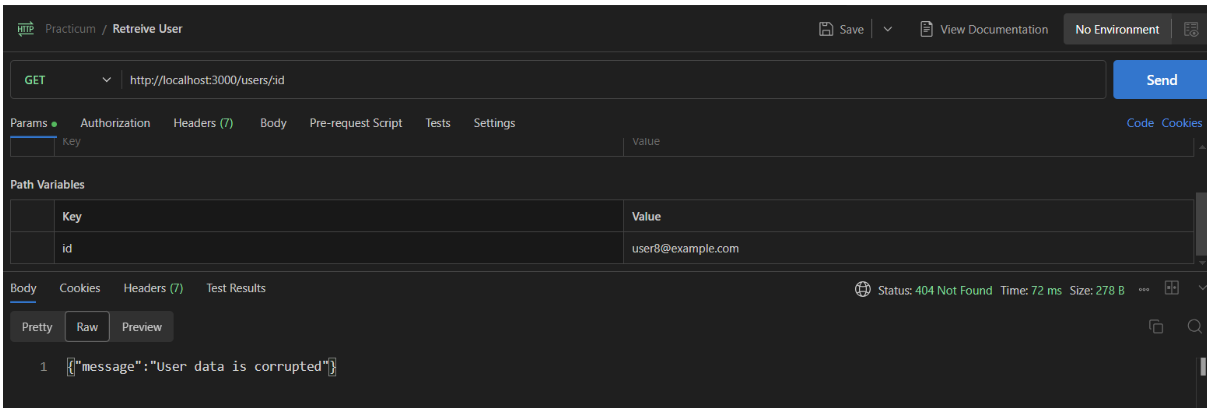
Figure 5: Proposed system under normal conditions

6.3 Experiment / Case Study 3: Simulated Data Corruption

The system was tested with simulated data corruption. The test scenario and outputs obtained for this case study are as follows:

Scenario - Another user record was collected in an android application and send to server for storage in an API request as body. The data was stored successfully in PostgreSQL and fabric network (by invoking smart contract function). The data stored in PostgreSQL database was manipulated using SQL query, i.e. the "dob" of a user was changed. Now, the data was retrieved from both storage systems and the SHA-256 hash of both records were calculated and compared for integrity check.

Output – The user data was not returned as the hash data does not match. This indicates that the data has been corrupted. As blockchain storage is the trusted and immutable storage the data from relation database was identified as corrupted. The overall time taken for the API execution was 72ms which is good even after such long processes like blockchain data retrieval, hash calculation and comparison.



GET http://localhost:3000/users/:id

Path Variables

Key	Value
id	user8@example.com

Status: 404 Not Found Time: 72 ms Size: 278 B

```
{"message":"User data is corrupted"}
```

Figure 6: Proposed system under simulated data corruption

6.4 Discussion

Analysing the different scenarios and outputs in the case studies, it is evident that the integrity of data stored in a traditional database can be validated against a much trusted and immutable storage like blockchain. The SHA-256 hashing technique performs a consistent and important part in verifying the data ensuring the data integrity is preserved. Combining a reliable and immutable storage like blockchain with traditional relational database enhances the robustness of data storage and its preservation. This system when incorporated with strong cryptographic hashing technique like SHA-256 increases the success rate. The adaptability of the system enables it to be combined with web-based or mobile application, indicating that this is not limited to android. The testing could have improved into more sophisticated scenarios like simulated data corruption in timestamps and the experiments conducted here are primarily focused on single-user operations, which could be upgraded to multiple users interacting at the same time. However, with the time taken for each experiment shows a promising improvement in data integrity preservation.

7 Conclusion and Future Work

The research question addressed in this project was: Evaluate the use of blockchain technology to preserve data integrity in Android mobile applications. This research evaluated the integration of two data storage systems – one traditional relation database, PostgreSQL and other trusted and reliable blockchain storage, Hyperledger fabric with one of the strongest hashing technique SHA-256 in preserving data integrity. The system shows practically faster timings for storing user data to relational database and fabric network simultaneously. The experiments conducted to assess the effectiveness of the solution shows a significant impact on validating data corruption. The stimulated data corruption case study, where the API request should retrieve data from dual storage, calculate the SHA-256 hash and returns the hash comparison results takes only 72ms, which is close to a standard API response time, but with assurance of reliable and trusted data. The hybrid data storage system not only ensures the data integrity preservation, but also a backup of original data from the immutable blockchain storage. Combining an innovative technology like blockchain with traditional database systems for securing data is a promising upgradation in the industry. Additionally, the system can also be incorporated into various applications like web-based or mobile applications, rather than just limiting it to android applications.

Utilizing more advanced resources and amble amount of time, the research can be widened to experiment on different blockchain platforms like Ethereum, Fantom, Polygon or Celo. Furthermore, the integrity checking can be expanded to check the timestamps or other malicious attacks. Though, the systems produced good results for single-user operations, the system can be validated against multiple users in a single time to be used widely in applications.

References

Musa, H. S., Krichen, M., Altun, A. A., and Ammi M., (2023) ‘Survey on Blockchain-Based Data Storage Security for Android Mobile Applications’, *Sensors 2023*, 23(21):8749. doi: https://doi.org/10.3390/s23218749.

Ichikawa, D., Kashiyama, M., and Ueno T., (2017) ‘Tamper-Resistant Mobile Health Using Blockchain Technology’, *JMIR Mhealth Uhealth 2017*. 5(7): e111. doi: 10.2196/mhealth.7938.

Khiem, H. G., Nam, T. B., Triet, M. N., Huong, H. L., Khoa, T. D., Bao, Q. T., Phuc, N. T., Hieu, M. D., Loc, V. C. P., Quy, T. L., Anh, N. T., Hien, Q. N., Bang, L. K., Trong, D. P. N., Ngan, N. T. K., Son, H. A. and Hong, K. V., (2023) ‘Applying Blockchain Technology for Privacy Preservation in Android Platforms’, *Web Services – ICWS 2023 : 30th International Conference, Held as Part of the Services Conference Federation, SCF 2023, Honolulu, HI, USA, September 23–26, 2023, Proceedings,2023*. Volume: 14209, pp. 47-61. doi: https://doi.org/10.1007/978-3-031-44836-2_4.

Le, B. K., Nguyen, N. T. K., Huynh, K. G., Nguyen, P. T., Nguyen, A. T., Tran, K. D., and Phan, T. H. T., “Elevating Android Privacy: A Blockchain-Powered Paradigm for Secure Data Management”, *International Journal of Advanced Computer Science and Applications (IJACSA)*, 14(11), 2023. doi: http://dx.doi.org/10.14569/IJACSA.2023.01411136.

Lakhan, A., Thinnukool, O., Groenli, T. M. and Khuwuthyakorn, P., “RBEF: Ransomware Efficient Public Blockchain Framework for Digital Healthcare Application”, *Sensors 2023*, 23(11), 5256; doi: https://doi.org/10.3390/s23115256.

Pal S., and Kumar, V., "Blockchain Based Multi App Spy System,", *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, Uttar Pradesh, India, 2022, pp. 18-21, doi: 10.1109/IC3I56241.2022.10072438.

Li, C., He, D., Li S., Zhu, S., Chan, S., and Cheng, Y., "Android-based Cryptocurrency Wallets: Attacks and Countermeasures,", *2020 IEEE International Conference on Blockchain (Blockchain)*, Rhodes, Greece, 2020, pp. 9-16, doi: 10.1109/Blockchain50366.2020.00010.

Vacca, A., Sorbo, A. D., Visaggio, C. A., and Canfora, G., “A systematic literature review of blockchain and smart contract development: Techniques, tools, and open challenges”, *The Journal of Systems & Software,* 2020. doi: https://doi.org/10.1016/j.jss.2020.110891.

Thakkar, P., Nathan, S., and Viswanathan, B., "Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform," *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Milwaukee, WI, USA, 2018, pp. 264-276, doi: 10.1109/MASCOTS.2018.00034.

Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., and Wang J., "Untangling Blockchain: A Data Processing View of Blockchain Systems," in *IEEE Transactions on Knowledge and Data Engineering,* vol. 30, no. 7, pp. 1366-1385, 1 July 2018, doi: 10.1109/TKDE.2017.2781227.

Hyperledger Fabric, “Install Fabric and Fabric Samples”, 2020-2024, Available at: https://hyperledger-fabric.readthedocs.io/en/latest/install.html.