

Research Article

Banking Data Authentication Using Blockchain and Homomorphic Encryption

Tamukong Brian Tanyie¹ , Mih Thomas Attia² and Aurelle Tchanga³

^{1,2,3}*College Of Technology, Buea , Cameroon*

Correspondence should be addressed to Mih Thomas; mihthomas@ubuea.cm

Received 3 October 2022; Revised 26 October 2022; Accepted 29 October 2022; Published 27 November 2022

Academic Editor:

Copyright © 2022 Tamukong Brian Tanyie , Mih Thomas Attia and Aurelle Tchanga. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Blockchain is a promising prospect in the finance sector with multiple use cases which include cryptocurrencies, Non-Fungible Tokens (NFTs) amongst others. Existing online banking systems are susceptible to fraudulent transactions, identity fraud and data harvesting. This is due to the little control users have on who accesses their data and financial records. A blockchain-based solution can address these issues. This study made use of a blockchain based banking data authentication system with homomorphic encryption to allow for encrypted computation of users' financial records. In this study, existing data on blockchain data authentication in the banking sector was analysed and critiqued. From the analysis, the use of a public blockchain and an off-chain computation system were suggested as alternative to a private blockchain or running all transactions within the smart contract. A prototype of a blockchain based authentication system was built to serve as proof of concept. The results from this study, present the time of encryption and entry of deposit transactions in the blockchain at 182.2 microseconds, the time of encryption of a deposit or withdrawal transactions in the blockchain at 248.5 microseconds and the time taken to decrypt an average of 10 transactions from the blockchain at 257 microseconds. The mining processes for these transactions is estimated averagely at ~0.04348 Ethers per transaction as gas fee. This value is equivalent to 47144.89 FCFA and is significantly high due to the use of datatypes like strings which hold encrypted data of large sizes. These results signify that the use of blockchain as a data authentication method can greatly improve the efficiency and security of financial records with improved access control, authentication and accounting of user's financial records and data.

Keywords: Blockchain; Homomorphic Encryption; Banking Data Authentication; Financial Data

1. Introduction

1.1 Background to the study

In recent times, the internet has become a way of life for so many people. The prevalence of online business has grown with an increase in internet users worldwide. Latest figures show that an estimated 4.9 billion people used the internet in 2021 making 63 per cent of the world's population. This is an increase of almost 17 per cent since 2019, with almost 800 million people estimated to have come online during that period. ("ITU", 2022) and in Cameroon, there were 10.05 million internet users in January 2022. Cameroon's internet penetration rate stood at 36.5 percent of the total population at the start of 2022. This indicates that internet users in Cameroon increased by 967

thousand (+10.6 percent) between 2021 and 2022. (Data Reportal, 2022) . With this increasing presence of digital business has access to services directly without the need to visit the services provider physical location (Trappe, 2018). The finance industry has made strides to digitise monetary transactions but still faces setbacks in terms of security, transaction time amongst others.

Security of online financial transactions is a main concern for customers' trust in e-banking services and specifically, in internet banking products and services. (Koskosas, 2011). One of the most recurrent vulnerabilities in digital or internet banking is identity theft because with access to a user's credentials an infiltrator has leeway to do with a user account, whatever he or she pleases and this is done through methods such as brute interrogation, social engineering skills amongst others. This practice leaves the onus of making sure that users accessing accounts are authorized and user information stays confidential to the financial institutions. AAA (pronounced "triple A") is an acronym meaning Authentication, Authorization, and Accounting (sometimes referred to as Access Controls, Authentication, Accounting). The AAA model was created to maintain control over user access. It is the framework underlying who has access to what resources, when, and for how long. (Thigpen, 2005). This study will be focusing on the Authentication aspect of the AAA acronym.

Authentication is the process of successfully validating the identity of a person or device. (Jansen W., 2003). It could be seen as a procedure where the user confirms their identity by providing x to the system, which the system then verifies by calculating $F(x)$ and comparing it to a saved value y . (Papathanasaki, et al., 2022). The methodologies on which all authentication is done involve three basic "factors":

Firstly, Something the user knows which is termed as Knowledge based Authentication. (e.g., password, PIN), Something the user has termed as Possession based Authentication. (e.g., ATM card, smart card, token) and finally, something the user is termed as Biometric based Authentication. (e.g., biometric characteristic, such as a fingerprint, iris, speech recognition amongst others.). (Lawal, Ibitola, Longe, Lawal, & Ibitola, 2013)

There are a variety of technologies and methods financial institutions can use to authenticate customers. These methods include Use of customer passwords, personal identification numbers (PINs), digital certificates using a public key infrastructure (PKI), physical devices such as smart cards, one-time passwords (OTPs), Use of "tokens" such as USB plug-ins, Transaction profile scripts, Biometric identification, and others (E-Banking Booklet, 2003). In Cameroon, the authentication methods primordially used are passwords, PINs, tokens, and one-time passwords.

Over time researchers, penetration testers and attackers have been able to find loopholes in the above-mentioned authentication systems. The vulnerabilities of existing systems are shown below

1.2 Vulnerabilities of Current Authentication Methods

Password Authentication

- Passwords are highly susceptible to man in the middle attacks if someone simply watches you enter the code
- Passwords should be difficult to guess, changed from time to time, and unique to a single account. (Wiedenbeck, 2005) and because users easily forget passwords will write them down and keep close to the authentication device which leads to easy theft
- Due to the recent technological advances made, High powered computers make it quite efficient to initiate dictionary and brute force attacks to obtain the password (Thigpen, 2005)

One Time Password

- Like Password Authentication could be deciphered by use of dictionary and brute force attacks

Personal Identification Number (PIN)

- Similar to Password Authentication could be deciphered by use of dictionary and brute force attacks as a high powered can easily guess the user's Pin and compromise the account

Token

- The USB token generators may be stolen or compromised and if not reported the tokens can be used to access user accounts.

The management of data and protection of user information by the users is a particularly important part of every security system as often said the weakest link in any security system is the user as user credentials can be stolen and used in compromising a secure system.

The legally appointed body dedicated to data protection in Cameroon ANTIC (National Agency for Information and Communication Technologies), stipulates that, financial market data must be kept for a specified period of ten years. Persons managing such data have an obligation of confidentiality, and disclosure is subject to prior authorization by the competent authorities. Any offender shall be liable to penalties, including imprisonment, as provided by Article 38 of the Prevention and Suppression of Money Laundering and Financing of Terrorism Regulation. However, the ANTIC ensures, on behalf of the State, the regulation, control, and monitoring of activities related to the security of information systems and electronic communications networks. It also assists in the identification of cybercriminals. ("Cameroon - Data Protection Overview", 2022).

Alternatives have been proposed in recent times to better authenticate users the reason behind making these additions vary from access control reasons to business development purposes like adding e-commerce elements. Modern authentication methods implemented by modern day banking systems to enforce authentication of users include.

Two-factor authentication (2FA):

This authentication method illustrated in Figure 1 below uses two of the factors earlier mentioned to validate the users. These factors include Ownership factor (a thing that the user has, such as cell phones), Knowledge factor (a thing that the user is aware of, such as a password), Biometric factor (a fact about the user biometrics or behaviour) (Papathanasaki, et al., 2022)



Figure 1: Two-factor authentication (Papathanasaki, et al., 2022)

The downside of this authentication method is the authentication is tied to a secondary device or factor say a mobile device and if for any reason this device is lost access to this device is permanently restricted.

Multi factor authentication (MFA):

Like Two Factor Authentication, Multi Factor Authentication illustrated in Figure 2 below makes use of three or more authentication factors to validate users. It is a way to offer an increased level of security to safeguard the security of computer equipment and other vital services from unauthorized access by combining at least three types of credentials. (Azrour, et al., 2021).

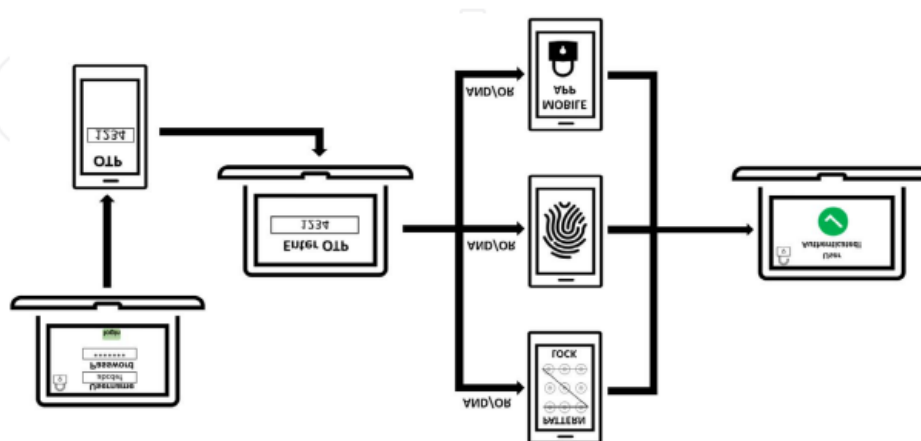


Figure 2: Multi-factor authentication (Papathanasaki, et al., 2022)

Consider the daily practice of withdrawing cash from an Automated Teller Machine (ATM). To gain access to a personal account and withdraw money, the user must submit a physical token (bank card) that represents the ownership factor, while the knowledge factor is represented by a PIN. This system might easily be made more secure by adding an additional biometric mechanism. (Papathanasaki, et al., 2022). One of the drawbacks of this system is the cost it takes to set up and maintain for multiple users and the complexity entailed in signing up which most users will try to avoid as they want the process of withdrawing money to be seamless. Also, from the perspective of biometric authentication, a disparity between the measured biometric presentation and the data recorded at the initial biometric registration can be problematic especially with inexpensive and inaccurate equipment. False Accept Rates (FAR) and False Reject Rates (FRR) are issues concerning biometric authentication. (Papathanasaki, et al., 2022)

In Cameroon presently, authentication in banks is primordial passwords, PINs, tokens, and One-Time passwords. This has known vulnerabilities and limitations such as

- Security breaches: These have caused user data hijack and breaches, identity theft
- Inability to access the growing digital economy as most transactions are physical in nature and validating online transactions takes longer than usual.
- Lengthy time to conduct transactions in the bank

Even though Cameroon has a diverse financial landscape has 454 formal institutions in activity, this includes 11 banks, 6 financial institutions, 412 MFIs, 19 insurance companies, 4 electronic money institutions, and the Post. Yet, the Cameroon market is at the start-up stage for its DFS ecosystem. DFS are provided by banks in partnership with telecom operators. Access to financial services grew from 12.2 percent in 2014 to 34.6 percent in 2017, along with similar trends in the rest of the CEMAC region, which is attributable to the growth of mobile money accounts. (The World Bank, 2019) . The banking sector's contribution to financing the economy is limited, accounting for only 10 percent of the domestic credit granted to the private sector (The World Bank, 2019) and this

phenomenon could be explained by a lack of trust of Cameroonians in electronic banking. Despite the statistics, the amplitude of cybercrime in Cameroon indicate that local banks have lost a colossal sum of over FCFA 3 billion in recent years with close to 1200 cases of identity theft recorded and over 800 cases of frauds registered. (ANTIC, 2022).

The above statistics stands to show that data authentication is a major hurdle most financial institutions face, and this has led to a waning trust of Cameroonian users in current online banking systems. Looking for a secure means of validating transactions and keeping records of these transactions becomes an ever paramount and of need.

In summary, this paper aims at building a working prototype for the implementation of blockchain and homomorphic encryption in banking data authentication. The specific objectives are

1. Building a working prototype as proof of concept for the implementation of blockchain and homomorphic encryption
2. Benchmarking the query time for executing banking transactions on the prototype with existing studies and systems in the same field
3. Making recommendations on the adoption and implementation of blockchain technology in banking data authentication.

2. Blockchain and Homomorphic Encryption

2.1 Blockchain

A blockchain can be defined as a digital record of transactions but unlike a typical recording system such as a database, a blockchain stores information using a distributed ledger technology. The World Bank in their paper “Distributed Ledger Technology (DLT) and Blockchain” define a Distributed Ledger Technology (DLT) as a novel and fast-evolving approach to recording and sharing data across multiple data stores (or ledgers). This technology allows for transactions and data to be recorded, shared, and synchronized across a distributed network of different network participants. (World Bank, 2017).

Blockchains employ cryptographic and algorithmic methods to record and synchronize data across a network in an immutable manner. Records of every transaction is stored in multiple ledgers with each of the ledgers existing in a node and if data is altered in one ledger by a malicious user or attacker the ledger is destroyed and all records on said ledger nullified. The visibility of block chain ledgers is dependent on the kind of block chain created. There exist various kinds of blockchains each having a specific use case these kinds of blockchain are in a later section. The Concept of a Blockchain was first introduced by Satoshi Nakamoto in a paper published in 2008 called “Bitcoin: A Peer-to-Peer Electronic Cash System”. There exist various kinds of Blockchains with each having their varying use cases. These types are detailed below:

- **Public Blockchains:** These are permissionless blockchains running on trustless systems as there exist no authority controlling the creation of blocks, they primarily use proof of work consensus and user identity is anonymous. Examples include Ethereum, Bitcoin
- **Private Blockchains:** These are permissioned blockchains having identified users. These blockchains are not publicly accessible they use they make use of voting and light proof of work as consensus. Examples include Hyperledger Fabric, Quorum and Corda
- **Hybrid Blockchains:** A blend of public and private blockchains, this blockchain aims at giving users the best of both worlds and does so by making parts of the blockchain public and others private. One big example that uses hybrid blockchain in the supply chain is the IBM food trust.

- **Federated Blockchains:** Also known as Consortium Blockchains these blockchains are a blend of private and public blockchains but unlike hybrid blockchains there exist two sets of users, those who control the Blockchain and decide who should get permission to access the network and create new blocks, and secondly, the users who can access those networks.

Structure of a Blockchain

In Nakamoto's study he states that there exist three key tenants of a block chain.

Maintain a Replicated Ledger: Blockchain utilizes a distributed network of nodes that stores and maintains a copy of a "public ledger" containing a full list of transactions (Nakamoto, 2022). Each time a new block is created in the blockchain it is appended using a cryptographic hash this cryptographic hash is used in connecting each block to another.

Cryptography: Use of cryptography in the ledger to guarantee the authenticity of the transactions, the privacy of the transactions and the identity of the participants (Nakamoto, 2022). In a Blockchain the cryptographic schemes used include:

Cryptographic hash function used in creating a block header, A cryptographic hash function takes a variable size input text and returns a fixed size alphanumeric output called a "hash value." (Godfrey-Welch, et al., 2018). The Hashed Result is irreversible and unique.

Public Key Cryptography: Also referred to as Public Key Cryptography this cryptographic scheme is used in signing transactions, it enables a trust relationship between users who do not know or trust one another, by providing a mechanism to verify the integrity and authenticity of transactions while at the same time allowing transactions to remain public. (Yaga, Mell, Roby, & Scarfone, 2018). Asymmetric Cryptography makes use of two keys a public and private key which is used to encrypt, mathematically computed, and used to sign transactions, verify digital signatures, and generate bitcoin addresses.

Consensus Logic: Due to the distributed structure of the Blockchain, there needs to exist, a structure with which blocks are created. This is so that the autonomy of creating blocks does not fall on a single individual there by enforcing the decentralized structure of the blockchain. Wazid describes the consensus logic as a decision-making process for a group, in which each individual member of the group constructs and supports the decision which works best for the rest of them. It yields a resolution which is supported by everyone to draw conclusions. (Wazid, Das, Shetty, & Jo, 2020). Over the years various consensus logics or algorithms have been formulated, with each having key advantages over others, the most adopted consensus logics include

Proof of Work: Here miners are tasked with solving a mathematical problem and the first miner to successfully solve the equation is rewarded as creator of that block and paid a certain amount for mining and maintaining the blockchain. The setback of this algorithm is its highly power consuming nature.

Proof of Stake: Here the task of adding blocks belongs to the user with the greatest number of assets on the blockchain, the drawback of this algorithm is it makes the system biased as users with the most assets control the system hence limiting the autonomy of the blockchain.

Proof of Capacity: Here, the block mining process is dependent on users with the most resources to conduct the mining process. Like proof of stake this makes users -with the most assets in this case storage space and computing power to mine blocks

The Byzantine fault Tolerance: The Byzantine Fault Problem poses a question on the trust of all nodes in a network to act honestly in the creation of a block. the Byzantine Generals' Problem was conceived in 1982 as a logical dilemma that illustrates how a group of Byzantine generals may have communication problems when trying to agree on their next move. (Byzantine Fault Tolerance Problem , 2022). In a blockchain, each general in the army can be likened to a node. and before a decision made, the only way to achieve consensus is by having at least $\frac{2}{3}$ or more reliable and honest network nodes. The Byzantine fault tolerance (BFT) is the property of a system that can resist the class of failures derived from the Byzantine Generals' Problem (Byzantine Fault Tolerance Problem , 2022).

Practical Byzantine Fault Tolerance: Designed to work in asynchronous systems, this algorithm orders all nodes in a sequence making one of the primary nodes the leader and allowing all nodes communicate with each other and an agreement made unanimously on if to add or not add a block to the network. The nodes need to verify each transaction or message to ensure it is authentic and unaltered.

Smart Contracts: A smart contract is a program written and stored on a blockchain; it is executed when a set of predetermined conditions are met. It was popularized by Ethereum and is written in Solidity which is a language designed by Ethereum. Smart Contracts serve as the backbone for decentralized applications. Which are applications which run as backbone of the blockchain.

Structure of a Block: Blockchain can be seen as a data structure used in distributed ledgers. Here data is stored and transmitted in packages called “blocks” that are connected to each other via a digital “chain.” The structure of a blockchain can be seen as below

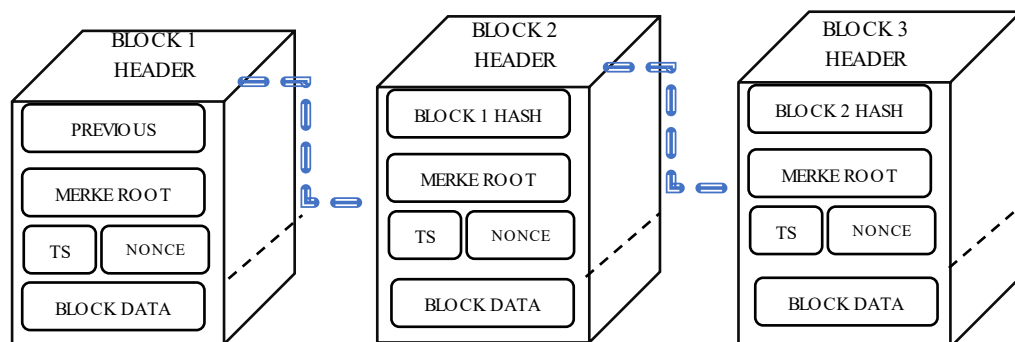


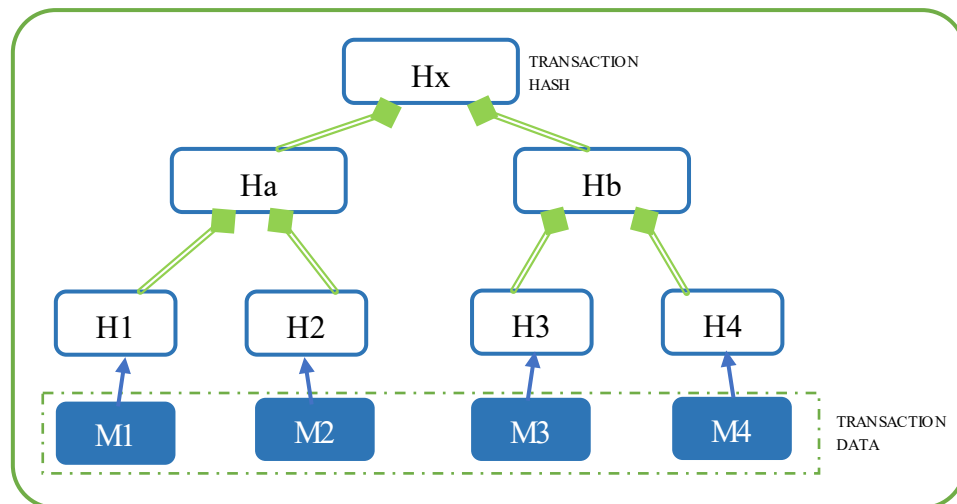
Figure 4: Structure of a Blockchain

Each block in a block chain contains a:

Previous Hash which is used to link each block to its preceding block and serves as a chain to make all records immutable which is one of the key features of a block chain

Merkle Roots are the results gotten from hashing a transaction using Merkle Trees, Merkle Trees were named after Ralph Merkle (Merkle, 1988). Merkle trees are created by repeatedly

calculating hashing pairs of nodes until there is only one hash left: the root hash or Merkle root,



which is a summary value. (Borde, 2022).

Figure 5: Structure of a Merkle Tree

Borde reports in his study that further research was done by the Ethereum foundation in 2015 to produce a Patricia Merkle tree which is a variation of the Merkle tree that allows for which aims at allowing light clients to make and get verifiable answers to various kinds of queries.

- Timestamp (TS): Records the time which each block was written on the blockchain
- Nonce is a portmanteau meaning “Number Only used Once,” it is a random whole number, which is a 32-bit (4 byte) field, the nonce serves as initialization vector used in the hashing process of the Merkle tree. This value is adjusted by miners during the mining process.
- Block Data which is a list of transactions done within a period. This data is split and used in calculating the Merkle root.

2.2 Homomorphic Encryption

Greek word homos meaning "same" and morphe meaning "shape". In computer science homomorphic encryption is used in the conversion of plaintext to ciphertext.

Oxford Languages describes homomorphism as a transformation of one set into another that preserves in the second set the relations between elements of the first. Greek word homos meaning "same" and “morphe” meaning "shape". (Ogburn, Turner, & Dahal, 2013)

Homomorphic encryption allows for the conversion of plain text to cipher text and executing computation on that cipher text such that the result of the cipher text when decrypted will be the same as if two plain text values were calculated on. It was first proposed in 1978 by Rivest, Adleman and Dertouzos, their goal was to devise a method of computing on encrypted data (Rivest, Adleman, & Dertouzos, 1978.)

Homomorphic Encryption uses asymmetric key cryptography for key generation, encryption, and decryption of data. There exist various kinds of Homomorphic encryption schemes based on their various properties as outlined by Wenyu et al.

If the two ciphertexts $E(x)$ and $E(y)$ are calculated by operation $E(x+y)$, i.e.

$$F(E(x), E(y))=E(x+y) \dots\dots\dots (1)$$

F stands for any operation, then E is an additive homomorphic algorithm.

If $E(x)$ and $E(y)$ can be computed by operation to $E(x \times y)$, i.e.

$$F(E(x), E(y)) = E(x \times y) \dots \dots \dots (2)$$

then E is a multiplicative homomorphism.

If $E(x)$ and y where y is a scalar value are computed by operation to $E(x \times y)$, i.e.

$$F(E(x), y) = E(x \times y) \dots \dots \dots (3)$$

then E is a scalar multiplicative homomorphism.

If an encryption scheme satisfies just one property it is termed as partially homomorphic encryption (PHE). The examples of PHE include

RSA - multiplicative homomorphism

El Gamal multiplicative homomorphism

Paillier additive homomorphism

For practical purposes and daily use, PHE is more suitable as it is faster and takes a shorter period to be executed. For this study and due to the additive nature of banking transactions I will be making use of partially homomorphic encryption, specifically the Paillier encryption system.

Nassar et al outline properties which makes Paillier encryption system suitable for conducting quick and fast transactions in their study, they include: (Nassar, Erradi, & Malluhi, 2015)

1. it makes use of an asymmetric key scheme making it impossible for a user with the public key to be able to decrypt a message
2. It is probabilistic making it impossible for adversaries to decipher if two ciphertexts are from the same plain text or not
3. It possesses the homomorphic properties for addition in particular which in this case is key in banking transactions.

Somewhat Homomorphic encryption is a term which defines encryption that makes use of some properties of fully homomorphic encryption and properties of partially homomorphic encryption. If an encryption scheme satisfies both additive homomorphism and multiplicative homomorphism, it is called total homomorphic encryption or fully homomorphic. If it satisfies just one property it is termed partially homomorphic.

In terms of computation, Fully Homomorphic Encryption (FHE) is more time consuming as each process produces an increasing number of irrelevant digits as residue from the computation after each transaction which can be defined as noise. To resolve these flaws multiple methods have been put forth. The first fully homomorphic scheme was presented in 2009 by Craig Gentry. This technique allowed one to compute arbitrary functions over encrypted data without the use of a decryption key (Gentry, 2009). In his method, Gentry first constructs a homomorphic encryption, then compresses the decryption circuit to a more uncomplicated form. It is then bootstrapped to obtain a fully homomorphic encryption procedure (Ogburn, Turner, & Dahal, 2013). In 2011 Craig Gentry and Shai Halevi devised an advanced approach that consisted of a fusion of Somewhat Homomorphic Encryption and another type of encryption called multiplicatively homomorphic encryption. This novel process eliminated the need for the compression step Gentry originally proposed in his dissertation. In this method, Gentry and Halevi devised a system to condense the

FHE ciphertext into a single ciphertext whose security was superior. (Ogburn, Turner, & Dahal, 2013)

Over the years huge strides have been made in perfecting fully homomorphic encryption, but efficiency remains an issue as the existing schemes are still too inefficient for practical purposes.

2.3 Research on Blockchain in the Finance Sector

Blockchain is currently a concept that has received significant attention in financial technology (FinTech). It combines computer technologies, including distributed data storage, point-to-point transmission, consensus mechanisms, and encryption algorithms. It has also been identified as a disruptive innovation of the Internet era. However, as blockchain is a breakthrough in data storage and information transmission, it might fundamentally transform the existing operating models of finance and economy, which might lead to a new round of technological innovations and industrial transformation within the FinTech industry (Mu Qi-Guo 2016). Research has been conducted and solutions and propositions have been made for the implementation of blockchain in the health sector as a recording system, in the voting sector as a means of aggregation and calculation of votes, in the finance system with the creation of crypto currencies

In his study, Masood states that (Masood & Faridi, 2018) . The blockchain technology and cryptocurrencies have opened the world to a whole new digital market with the availability of multiple coins in circulation and the use of non-fungible tokens (NFT) and sales of token. Prominent cryptocurrencies include Bitcoin Ethereum and Libra

Blockchain also have the potential to be of help in the insurance industries as it will be easy to detect frauds as there will be decentralized public Ledger so customer verification and policy details as well as transactions will be available on public network and since the minimum has been eliminated it would be much easier for customers to submit claims and easier for the insurers to settle these claims.

A key reason for the existence of banks is conduct asset transformation, store liquidity, and utilize economies of scale providing services based on any storable asset (Bunea, Kogan, & Stolin, 2016). using blockchain technology have implemented P2P lending platforms where sellers of capital are matched with buyers with minimum transaction costs. These FinTechs are mimicking the two-sided market model for banks explained by Rochet and Tirole (Rochet & Tirole, 2003) for matching excess capital savers with excess demand from borrowers in a decentralized manner.

Presently, blockchains are utilized in a diverse array of banking applications including financial asset settlement, economic transactions, market predictions and business-related services (Haferkorn & Diaz, 2015). Blockchain is anticipated to be central for sustainable global economic development in the future, benefiting society and consumers in general (Nguyen, Nguyen, Nguyen, & Pham, 2019). Several studies explore blockchain based applications developed for fiat currencies, securities, and derivatives (Christensen et al. 2015; Fanning and Centers 2016; Peters and Panayi 2016; Paech 2017; Nijeholt et al. 2017). Blockchains exist independent of the potential and economic value of blockchain and is independent of Bitcoin value or the value of any cryptocurrency (Buterin, 2015).

The concepts of blockchain authentication with the use of homomorphic encryption is a novel field and researchers have made strides to implement a working system which will be detailed in the next chapter.

3.Materials and Methods

In this section of my study, I will detail the materials of the study and methodology used in conducting this study; This section also describes a prototype, the reasoning behind the design and development process and the software architecture.

3.1 Materials of the Study

In the research done prior to building this prototype, a great many tools were studied as viable options for development with each having features which made them not ideal for practical and everyday use, and as such better alternatives were chosen. This section outlines the various materials used in building the prototype hereon described as a Decentralized Application (DApp), outlining their advantages over the alternatives.

3.1.1 Software Components Used

The Ethereum blockchain was selected as the foundation due to its public, permissionless nature, providing robust protection against the 51% attack, a vulnerability where malicious actors could potentially control the blockchain's mining power. For a local testing environment, Ganache was used to simulate a personal Ethereum blockchain.

To develop and manage smart contracts, the team used Truffle, a powerful development framework for Ethereum. The contracts were written in Solidity, a language tailored for the Ethereum Virtual Machine (EVM), and tested in Remix IDE, an integrated development environment that facilitated the smooth deployment of contracts without the need for extensive plugins.

In Ethereum, every blockchain transaction requires a small payment, or “gas,” to power the mining process. To handle these payments, the MetaMask crypto wallet was used. Installed as a browser extension, it allowed for easy handling of gas fees with dummy accounts preloaded with ethers by Ganache.

A web application was created to serve as the user interface, allowing users to view transaction records and manage their finances with options to withdraw, deposit, and transfer funds. React.js was employed for building the front end, while an API in NestJS connected the application to a MySQL database, retrieving essential user information and transaction IDs. A JavaScript library called web3.js facilitated the connection between the smart contract and the web application. Financial data was encrypted using the Paillier homomorphic encryption library (paillier-bigint.js), while additional sensitive information, like user keys and initial deposits, was further secured with AES encryption, using the Crypto.js library.

To support the blockchain and web application hybrid system, a MySQL database was chosen to store pointers to blockchain data and user credentials, and it was organized through a detailed class diagram.

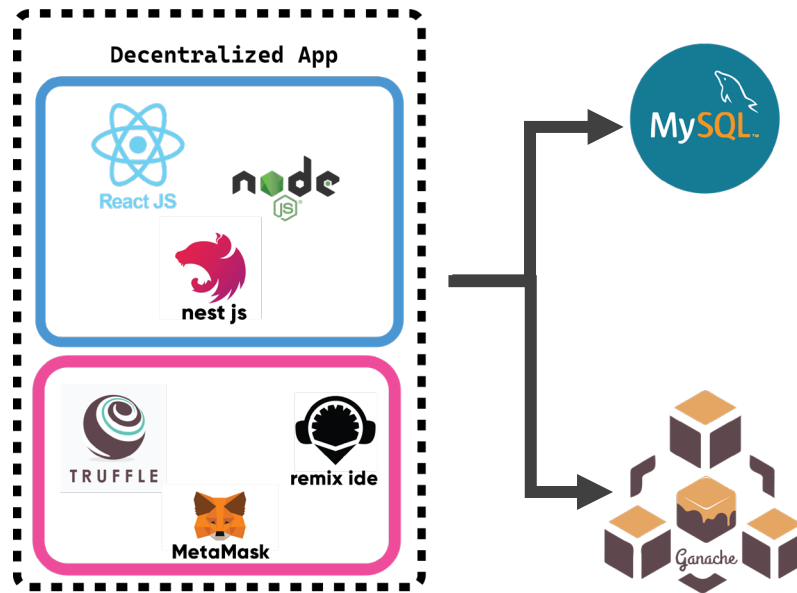


Figure 7: System Tools and Interaction

3.1.2 3.1.2 Hardware Components Used

This project, was built and run on a personal computer with

- An 11th Gen Core i7 Intel processor,
- 16 GB, 2 x 8 GB, DDR4, 3200 MHz
- NVIDIA GeForce 3060 RTX

The decentralized application was run on Microsoft Edge browser visuals of the results are shown in the appendix of this study. Images of the implementation process including code written during the study are shown in the appendix A to C

3.2 The Methodology

The prototype implemented, allowed for securely recording banking transactions on a blockchain such that details of each transaction were not available to the public and if for any reason the database was breached by attackers, the data could not be read or tampered with. The method of build can be split into three parts, these include:

1. Homomorphic Encryption
2. Decentralized Applications
3. External Database.

The build process is the above-mentioned parts interacting with each other to build the prototype.

3.2.1 Specific Procedures

Homomorphic encryption is used to securely encrypt the recordings which will be registered on the block chain. The encryption of transaction data was done off chain and the blockchain was solely used to store details of encrypted transaction data where it could be fetched and computed on. In implementing this scheme, Partially Homomorphic Encryption was made use of and specifically the Pallier Encryption System (Mark A. Will, 2015) which is an additive homomorphic encryption system was used. Reasons for using this encryption scheme are because it allows only one computational function (mostly addition or multiplication) to be performed an unlimited number of times on the ciphertext. (Regueiro, et al., 2021)

Also, Homomorphic Encryption is cost efficient as computing does not require as much compute power. Finally, the transactions to be conducted are additive in form as bank transactions are either addition or subtraction operations.

Paillier Cryptosystem Key Generation Algorithm

1. Select two large prime numbers p and q where $\gcd(p-1, q-1) = 1$
2. Calculate: $n = p * q$
3. Calculate $\lambda = \text{lcm}(\text{least common multiple}) (p-1, q-1)$
4. Select g as a random integer where $g \in \mathbb{Z}_n^{*2}$
5. Define $L(x) = x - 1n$
6. Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse
7. $u = (L(g^\lambda \bmod n^2))^{-1} \bmod n$
8. Public Key = (n, g)
9. Private Key = (λ, u)

The encryption process of the paillier cryptosystem is depicted as below

Paillier Cryptosystem Encryption Algorithm

Encrypt a message M where $M \in \mathbb{Z}_n$

1. Select r as a random integer where $r \in \mathbb{Z}_n^{*2}$
2. Calculate $c = g^M \times r^n \bmod n^2$

Paillier Cryptosystem Decryption Algorithm

Decrypt a message c where $c \in \mathbb{Z}_n^{*2}$

1. Calculate $m = L(c^\lambda \bmod n^2) \times u \bmod n$

The key generation, encryption, and decryption of values for the paillier cryptosystem could be described as follows:

Decentralized Application (DApp) was built which initiated and executed the transactions recorded on the block chain. User's gained access to the system using their account number and chosen password.

The smart contract was used in recording entries to the blockchain I recorded transaction details of various kinds be it transfer, deposit or withdrawal on the blockchain.

An external database was created with MySQL which held hash values of each transaction and the corresponding account balance of each customer. In production, this database will be a secure database existing in the banks servers holding encrypted details on each users' credentials such as private and public keys for transaction encryption, usernames, and passwords. A transaction table was also created to hold the corresponding transaction number for each transaction in the blockchain. This was done to ensure that the query time was much faster. The Truffle Suite which comprised Ganache, Truffle and Drizzle was used in building the decentralized App

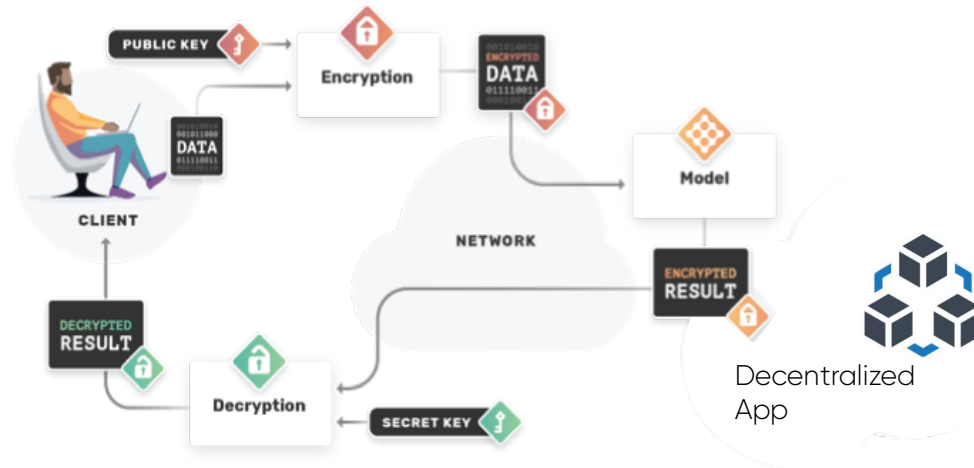


Figure 10: Data Flow of client information.

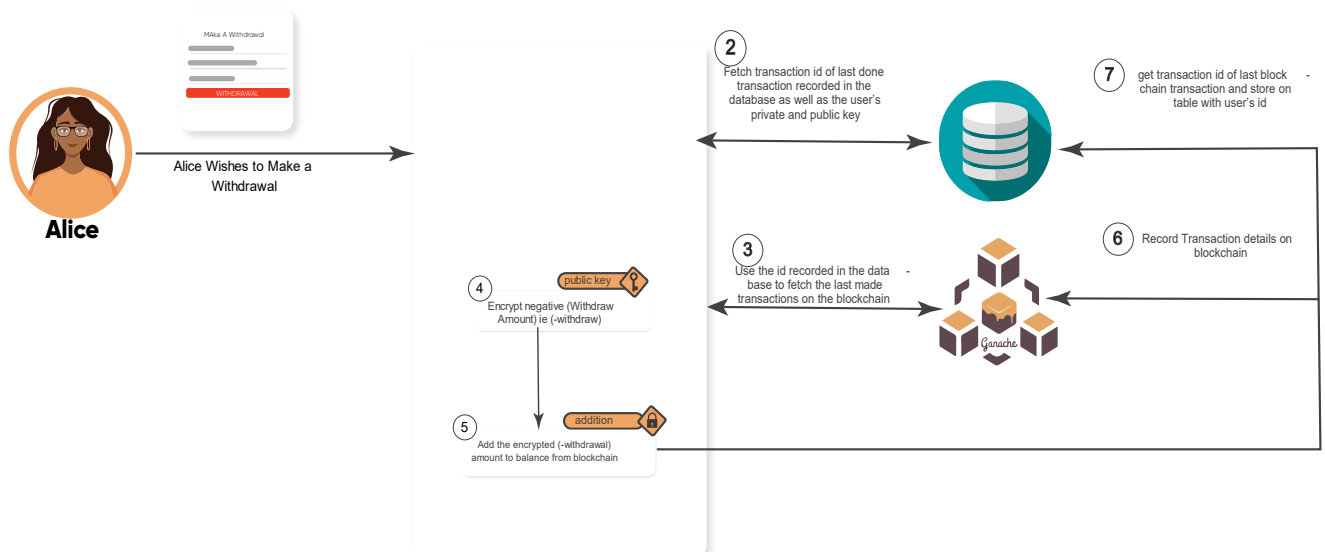
The diagram above, illustrates how a user interacted with the decentralized application. the data is initially encrypted with the public key before being uploaded to the blockchain. Computation conducted in the decentralized application was done on the encrypted data without decryption at any stage off the computation process. The results were then sent to the user encrypted and with the use of his private key, the user could decrypt these results and read the information that exists.

3.1.3 3.2.2 Decentralized Application Procedures

The processes involved in everyday banking for users include account creations, deposits, withdrawals, and transfers. In this section, these procedures were detailed explaining how they were implemented.

Account Creation.

This is one of the most fundamental processes fundamental processes in banking, each user of the system, upon signing up to the decentralized app creates an account and upon creation a private key and public key was generated that was used to encrypt user's data. This data comprised User Information (U_A) and Account Balance (AC_A). The data was then recorded on the blockchain.



Withdrawal

A description of the transactions could be depicted as follows.

Say a user Alice wanted to withdraw the sum of 10,000 FCFA from her account via the DApp. Alice would initiate a transaction request, which was done in the DApp. Checks were then done to verify if a user had the funds required to conduct the transaction. This was done by fetching the Alice's latest or last transaction from the blockchain, decrypting the amount with Alice's private key, and checking the value to make sure it correlates with the value in the remote database. If the balance was sufficient, the withdrawal process was initiated. First, the Alice's public key was fetched from the database and used in encrypting the withdrawal amount 10,000 FCFA converting it to a ciphertext (WP_A). The amount was then subtracted from Alice existing account balance cipher text (ACA) already existing in the blockchain and the transaction id was stored in the database alongside Alice's id number. The process was depicted by Figure 11: Withdrawal Process

Deposit

Making a deposit was as follows Say Alice wished to make a deposit of 5,000 FCFA, Initially, Alice's latest or last transaction was fetched from the blockchain using the id recorded in the database and the amount was decrypted with Alice's private key making sure the value correlated with the value in the remote database. The amount to be deposited was then encrypted with her cipher text and added to the account balance. The resulting value was recorded in the blockchain. The process is depicted below:

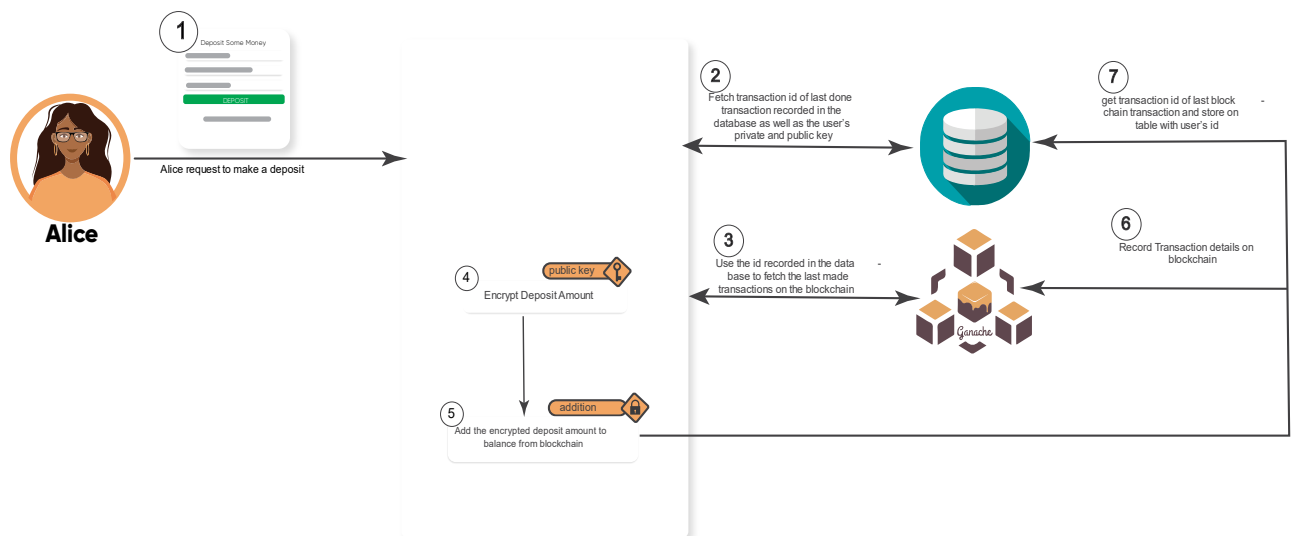


Figure 12: Deposit Process

Transfer

The transfer of funds between two parties was the combination of a credit and a debit of two different accounts in accounting terms. In lay terms, the transfer of funds is simply a withdrawal of money from one account and the deposit of that same amount in another account. In the same light, say Alice wished to transfer 20,000 FCFA to Brian the transfer is explained as follows.

Alice will input her account number, Brian's Account number and the transferred amount. When Alice initiates the transfer the DApp made a withdrawal of 20,000 FCFA as earlier mentioned from Alice's balance using Alice's public key to encrypt and make the withdrawal. Upon completion The DApp made a deposit of the same amount with Bob's public key into Bob's account. Once

successful the resulting amount of both parties is recorded in the blockchain with the entry labelled as a transfer. Details on the transactions were as below

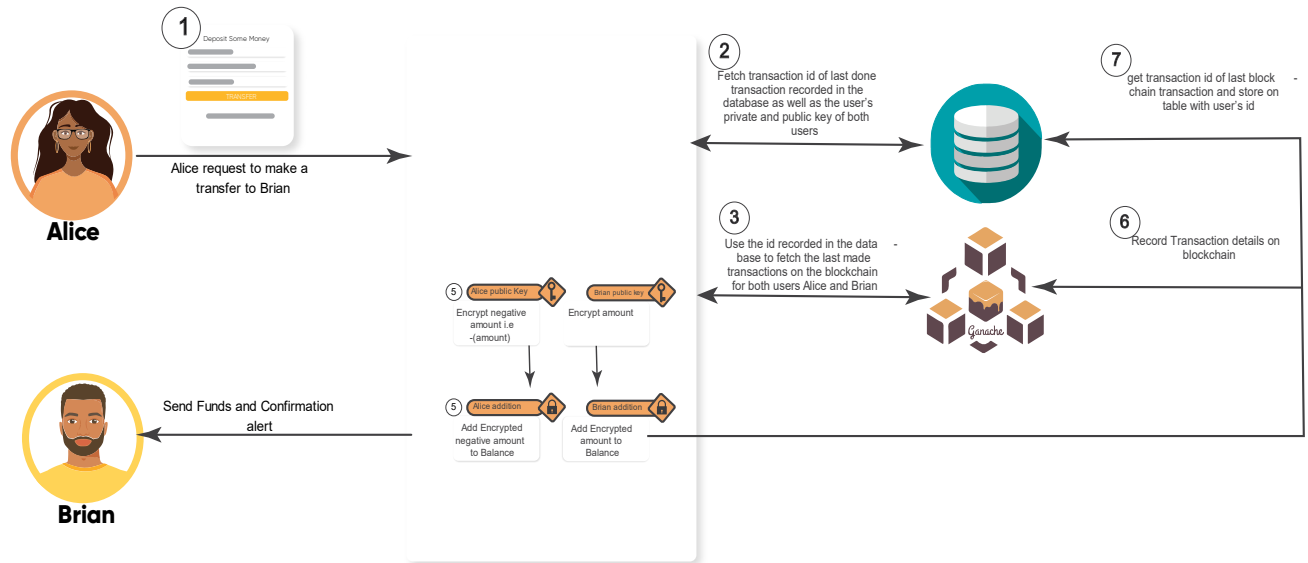


Figure 13: Transfer Process

Note should be taken that throughout the recording of transactions the private key is not used at any point in time. Hence, the data is not decrypted by the DApp throughout computation at any point but was computed on while encrypted and the actual content of the data was only decrypted at the user's device therefore allowing solely users themselves on their devices the ability of viewing their financial records. This setup also limited the access of financial records to the user solely. The user could then decide to share this information with whomever he/she chose. The diagram below shows the transaction history fetching process.

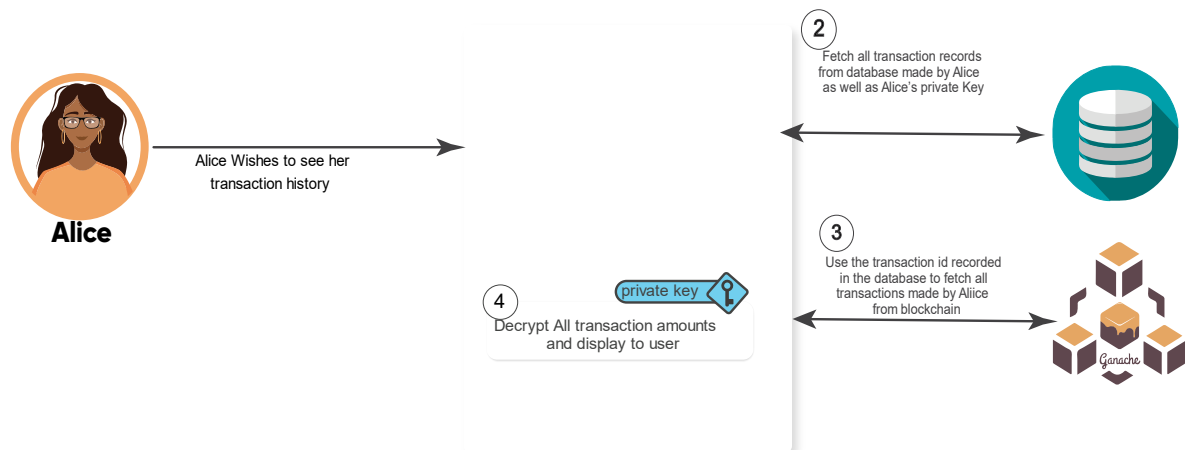


Figure 14: Fetching Transaction History

3.3 Ethical Considerations

This prototype was built with the following considerations and assumptions made:

- To ensure the safe keeping of user's data stored in the remote database an extra layer of encryption was placed on the data stored specifically the user's private key, and initial deposit this was done using AES encryption from the Crypto.js Library

- Due to the amount of gas required to pay for every transaction to be added to the blockchain, the use of one sole wallet through which all transactions are done was implemented,
- Also, to avoid increased gas prices for each process be it deposit, withdrawal or transfer, the actual encryption, crediting, and debiting of accounts was done off chain and the results were then recorded on the blockchain by a sole account, this made it more efficient and cost effective for banks in Cameroon and worldwide.

Ethically, as with all new technologies some ethical considerations needed to be made for the use of blockchain. By default, blockchain uses tight security due to its use of cryptographic hash to code transactions this inherently provides a degree of privacy for its participants, and it also provides beauty and integrity for all transactions in the blockchain but even with this sophisticated level of encryption and chaining, hack attempts have been made on the blockchain. The value of information existing on the blockchain influences the hack and increases the truth of exposure of the information that is why we made use of homomorphic encryption as an extra layer of protection of financial figures

4. Results and Applications

Having built a working prototype, this section reports the results of the build. The existing prototype serves as proof of concept, for the existence of an encrypted blockchain authentication system this system if applied in banks, will increase the verification and transaction time for customers who make use of online banking it also allows them to be able to make payments via online platforms much faster. This prototype serves as a base on which future and better iterations of the system can be built.

4.1 Results of the build

After implementing the prototype, the results of the build could be displayed as below.

3.1.4 4.1.1 Homomorphic Encryption and Blockchain Storing of Data

Due to the large size of keys, needed in homomorphic computation and the massively increasing size of numbers due to the noise after every computation, the int data type which is the conventional way of storing numbers is unsuitable as it has a limited size for holding keys and computed results. As a result, BigInt data type is used in storing encrypted keys. Each computation therefore begins with converting a standard integer such as 20,000 into BigInt which is 20,000n. The conversion process is as follows.

```
const deposit = 20000

const bigintdeposit = BigInt(deposit)

console.log(bigintdeposit)

//Result is 20000n
```

Sample keys generated from this library are depicted in Figure 15

```

Register.js:42
PublicKey {n: 26066160971373307406149482198090310010455050122022...6876147558904115488887
04233769571992208724399213n, _n2: 67944474778554504472185711494692791351487494737602...72
8902187748460862054904664269159418827795019369n, g: 6615303554528340707243540943164380
394437623946741...653756613036858761830235341435847621641994486424n}

Register.js:43
PrivateKey {Lambda: 65165402428433268515373705495225775026137625305055...3350217245077733
94871861769175867358833577988072n, mu: 80538364015912390320539472917630805578313475250
09...417755255162013740537952127728884965883956215240n, _p: 20267279185121808364002627745
529710855021783190038...633400373000683295322909898451829012655123944997n, _q: 1286120388
103621378409711558834341306834210018106...7141274848586346740783472586142755442192885019
29n, publicKey: PublicKey}

```

Figure 15: Sample of Generated Keys

The encryption and decryption of values is done using `PublicKey.encrypt()` and `PrivateKey.decrypt()` functions, respectively. The addition process similarly is done using the `PublicKey.addition()` function. Note should be taken that these functions as well as the key generation functions are found in the `paillier bigint` JavaScript library.

The encrypted values stored in the blockchain are as below

[illegible]

Figure 16: Encrypted Transaction Data in Blockchain

As seen, the data stored is a mixture of text and alphanumeric keys of large sizes. After each transaction, the transaction ID is fetched and stored in the database to easily fetch the data at any given time faster. Images of stored records are as below.

Structure

Data

Constraints











Indexes

Triggers

DDL

Grid view

Form view



Filter data

Total rows loaded: 16

	id	address	transaction	transid	date	userid
1	7032d37c-44f9-4925-9250-a3af8379408a	Come	DEPOSIT	39	1656390853083	743aaf4-8c01-4e1d-ad60-ae85b09a49
2	51e2963c-ca81-4371-a5b7-9d87a3e1a6cb	Come	DEPOSIT	40	1656390943231	743aaf4-8c01-4e1d-ad60-ae85b09a49
3	82d6c02c-8675-4a3d-ab2f-c4bd7c13186b	Come	WITHDRAW	40	1656391124048	743aaf4-8c01-4e1d-ad60-ae85b09a49
4	10262005-8e69-4884-2b63-7f24c26e29bb	Limbe	WITHDRAW	41	1656393919464	743aaf4-8c01-4e1d-ad60-ae85b09a49
5	fefe45ce-e173-4421-9c9a-afcd8ca1cc90	Limbe	WITHDRAW	42	1656421501508	743aaf4-8c01-4e1d-ad60-ae85b09a49
6	65b40158-0a48-4941-9717-b988d647c80b	Longer	DEPOSIT	44	1656421692041	743aaf4-8c01-4e1d-ad60-ae85b09a49
7	1c246ab4-67c8-4f40-8ac2-937268e10654	Douala	WITHDRAW	44	1656421782693	743aaf4-8c01-4e1d-ad60-ae85b09a49
8	b4d6f6ba-01c1-4e42-80eb-47905011f1d1	Brian	WITHDRAW	45	1656422244882	743aaf4-8c01-4e1d-ad60-ae85b09a49
9	74c60a63-4333-4a9d-80bc-9e4dc2c39c4	Buea	DEPOSIT	48	1656426359798	5af2beca-9aa2-4351-95df-7f0532df108
10	18c63d7f-d2de-447f-9e12-5943db3354ab	Buea	WITHDRAW	48	1656426573219	5af2beca-9aa2-4351-95df-7f0532df108
11	413abde0-06d7-4907-9e4d-ec3db483658	Buea	WITHDRAW	51	1656426768326	5af2beca-9aa2-4351-95df-7f0532df108
12	4638a47d-fd2e-4680-b6a4-fb2bc3b20f9	Buea	DEPOSIT	51	1656428572564	5af2beca-9aa2-4351-95df-7f0532df108
13	558730b4-7689-479f-843b-2b85f71840e3	Limbe	TRANSFER	52	165643842324	5af2beca-9aa2-4351-95df-7f0532df108
14	71da6e9f-cata-4b7-965c-ad67263a561f	Buea	WITHDRAW	52	1657785750539	5af2beca-9aa2-4351-95df-7f0532df108
15	b9c8ebff-4799-44d2-833c-707a03e5850a	Buea	TRANSFER	53	1657785837038	5af2beca-9aa2-4351-95df-7f0532df108
16	c2fe737d-3c83-4474-a458-a726d4ffadfe	Douala	WITHDRAW	54	1657789536424	5af2beca-9aa2-4351-95df-7f0532df108

Figure 17: Transaction details stored in a database

Each transaction is stored with the address of the branch in which it was made, the transaction type, the date, user ID and the transaction ID. When fetching all transactions, and I get of all transactions done by user ID is cumulative and used to display user's transaction history

3.1.5 4.1.2 The Web App

The User Interface built with ReactJs for capturing user data and displaying results are shown below.

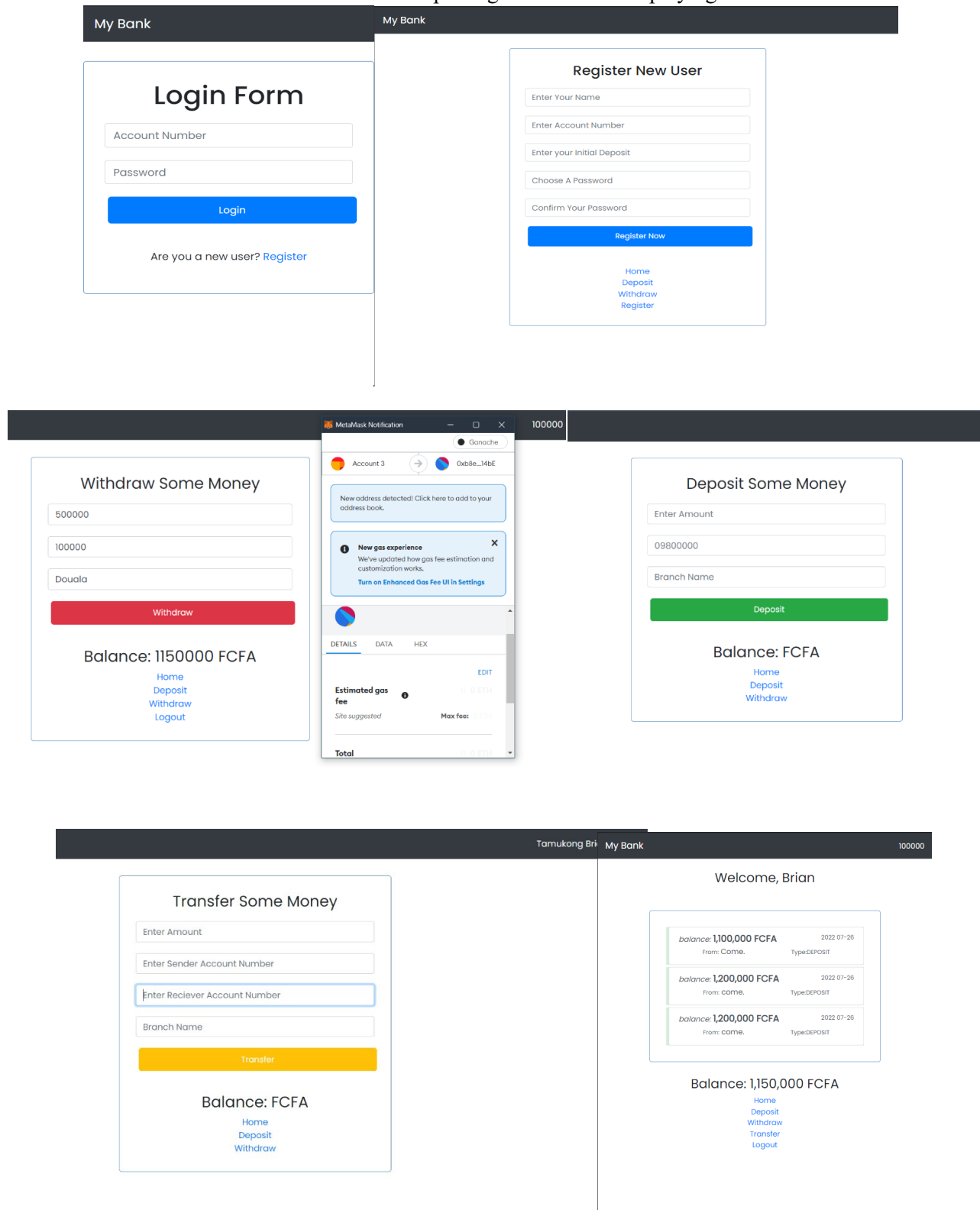


Figure 18: Web Application UI

4.2 Execution Time Statistics

As per objective, this study aims at making authentication and Validation of transactions quicker and even more efficient. To measure execution time for each transaction, some key metrics were measured to produce specific outcomes. These metrics include.

- The Time taken to encrypt and store data on the blockchain
- The Time taken to fetch and display of transaction data from the blockchain

To measure these metrics, Timepoints were set at the start and end of every code execution to be able to properly get the time taken to conduct every transaction.

3.1.6 4.2.1 Encryption Time

Measuring the time taken to conduct each transaction was done as thus

1. Timepoint variables were created at the start and finish of every transaction
2. A difference between the start time and the end time is computed.
3. The transaction is done 10 times and an average of each of the time difference is documented as average time for each transaction.

Given at base there exists two kinds of transactions that is withdrawal and deposit with a transfer being a combination of both a withdrawal and transfer, the results below include the time for both a deposit and withdrawal.

Table 2: Encryption Time execution

Execution	1	2	3	4	5	6	7	8	9	10	Mean
Withdrawal Time/ms	255	247	250	232	263	235	218	240	289	256	248.5
Deposit Time/ms	170	188	179	173	220	177	184	180	175	176	182.2

The execution time measuring process is seen below

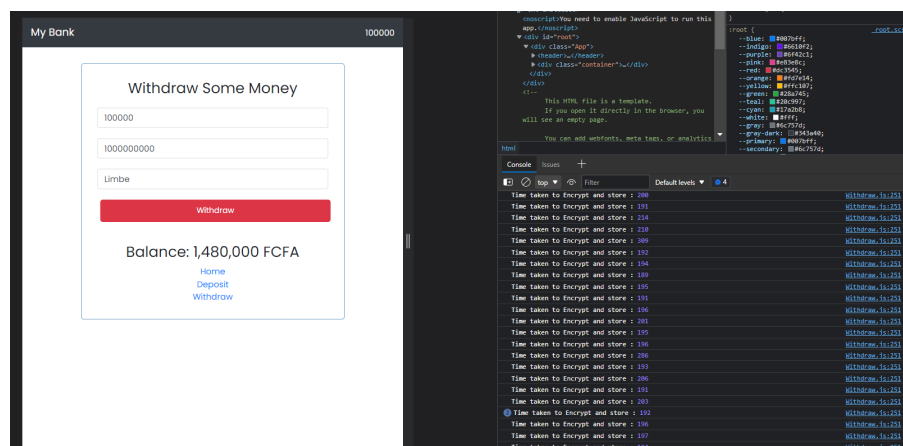


Figure 19: Code Execution Time measurement

From the data collected in Table 3, or graph of execution time over iterations can be drawn below.

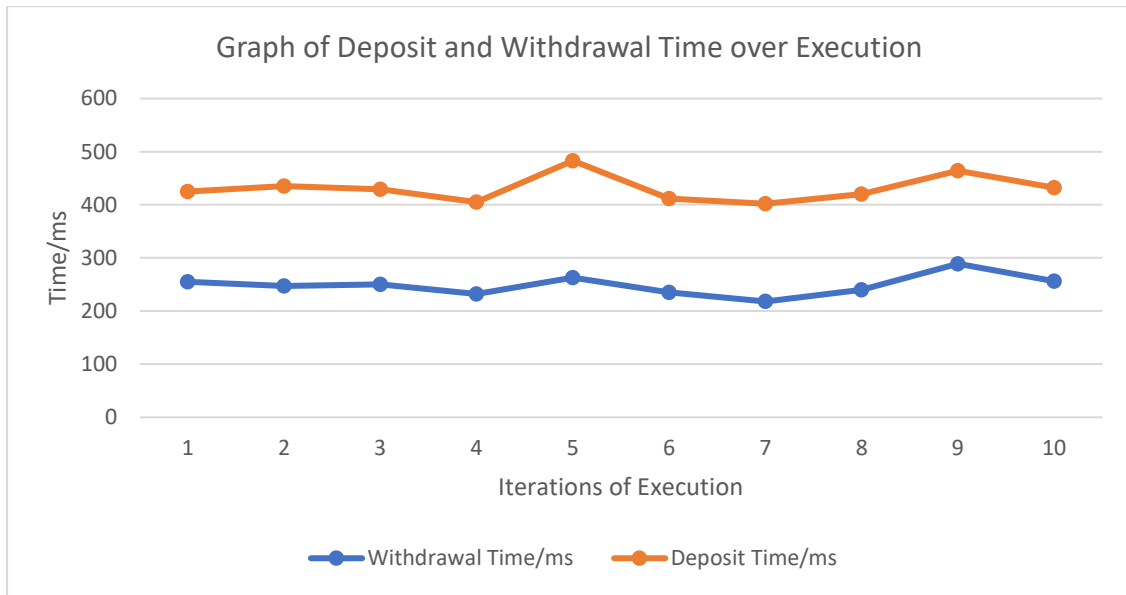


Figure 20: Graph of Deposit and Withdrawal Execution Time.

From table 3 above, the average time taken to execute the encryption of a withdrawal is 248.5 ms whereas the time taken to effect the encryption of a deposit is 182.2 ms.

3.1.7 4.2.2 Decryption and Display Time

Measuring the time taken to display user transaction history was done as thus

1. Timepoint variables were created at the start of the query of data from the blockchain and at the end of all decrypted transactions fetched from the blockchain.
2. A difference between the start time and the end time is computed.
3. The transaction is done 10 times with incremented number of transactions fetched and an average of each of the time difference is documented as average time for each transaction.

Table 3: Decryption and Display Time execution

Transactions fetched	1	2	3	4	5	6	7	8	9	10	Mean
Fetch Time/ms	164	189	232	213	228	191	251	255	248	257	222.8

The execution time measuring process is seen below

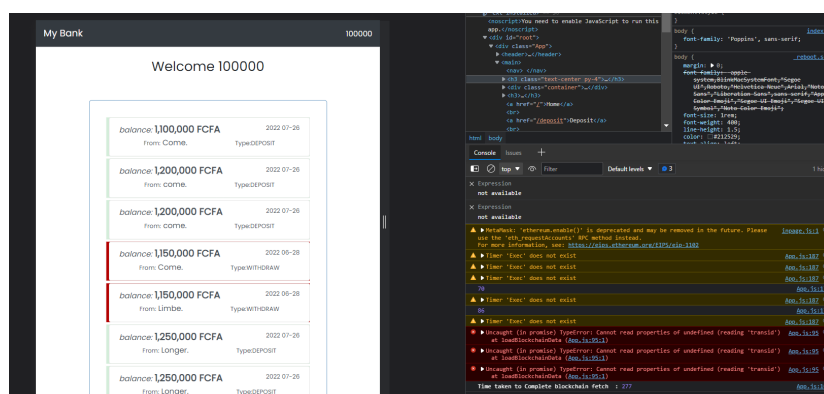


Figure 21: Code Execution Time measurement

From the data collected in Table 3, or graph of execution time over iterations can be drawn as in Figure 22.

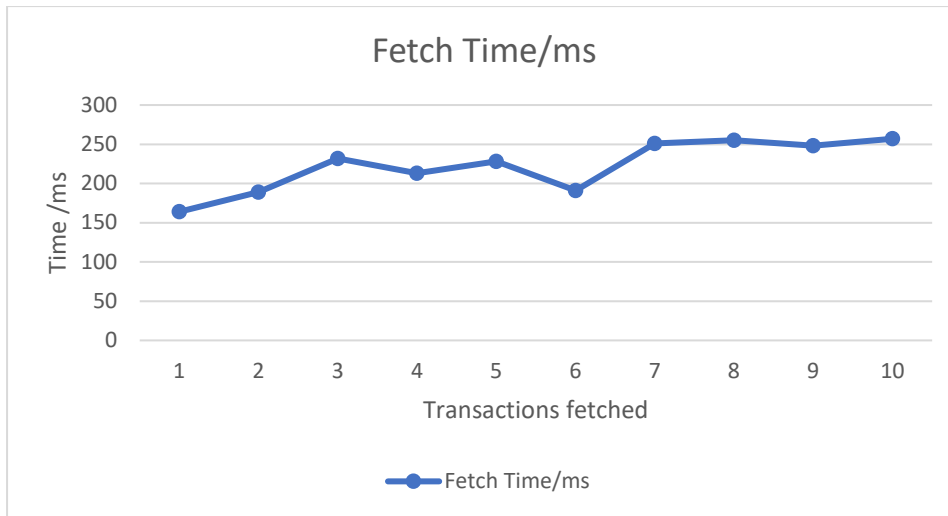


Figure 22: Graph of Execution Time.

In summary,

- The time taken to encrypt and store deposit transaction information to the blockchain is **182.2 ms**
- The time taken to encrypt and store withdrawal transaction information to the blockchain is **248.5 ms**
- The time taken to decrypt and display ten transactions from the blockchain is **257 ms**.

5: Discussions, Conclusion, Recommendations and Perspectives

5.1 Discussions

This study sought to encrypt data, such that it could be computed on while encrypted and the results could solely be decrypted by users. Transaction speed was key in the development prototype as well as efficiency. As reported by Godfrey-Welch and collaborators (Godfrey-Welch, et al., 2018) in their study, the use of blockchain, reduces transaction time significantly, reduces need for excessive bank charges and increase security in authorization.

The results in Figure 20 indicate that in average the time taken to encrypt transactions and store in a blockchain or fetch transactions and display on the block is less than a second. If the main net or main Ethereum blockchain was used to store the transactions, the transaction time will be increased. According to ethereum.org, the time taken to mine a new block range between 12 to 14 seconds. This sets the average block mining time to 13 seconds or 13000ms. Thus, the average time to encrypt and store a deposit is 13182.2 ms and the average time to encrypt and store a withdrawal on the blockchain is 13248.5 ms. This value is an estimate and may fluctuate based on the quality of network of the user or the bank. For this study, a 4G CAMTEL Internet connection was used with a 5.60 Mbps download speed and 1.37 Mbps Upload speed.

This data however contrasts the average time taken to make a traditional transaction on say an ATM where users must slide their cards and fill their information, put their secret pin and validate the transaction before making the withdrawal. Also traditionally, users will have to queue for a long time which on a busy day at the bank may entail them spending the whole day to be able to make a

bank transaction. Digital banking not only gives them access to faster transaction time, but it also allows them to be able to interact and make transactions quicker, even more efficiently and securely.

Decryption equally shows a reduced query time of 222.8ms for an average of 10 transactions as seen in Figure 22 ; The graph saw an increase in the amount of time taken for fetching 10 transactions over the time of fetching 2. This was expected as the number of transactions continuously increase thus requiring more time. In the study by Kim et al.(Kim, Eun Kim, Park, & Sohn, 2021) it was recorded that the average query execution time for 3 voting rounds which were homomorphically encrypted was 51ms. This difference could be explained by the fact that they used Hyperledger Fabric which is a private blockchain which though faster is permissioned and private hence removing the public and trustless structure of a public blockchain. The figure below shows a graph of fetching 0 to a thousand rounds conducted by Kim et. al.

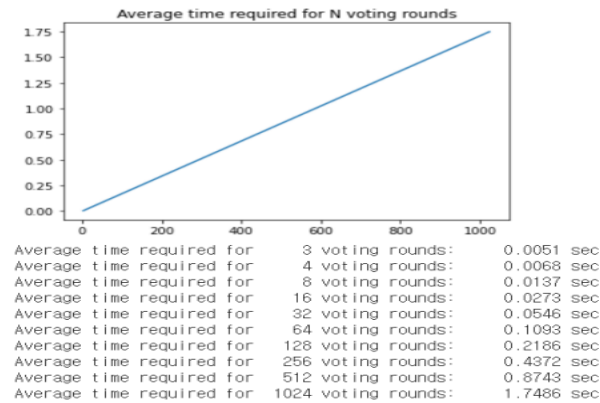


Figure 23: Average time required for N voting rounds with data size 400+ bytes (Kim, Eun Kim, Park, & Sohn, 2021).

Due to the use of the off-chain strategy to minimize the amount of gas, the amount of gas consumed per transaction was lesser with each transaction costing ~0.04348 Ethers per transaction as gas fee. This value is equivalent to 47144.89 FCFA per transaction on the blockchain as of August 1st, 2022. Reguerio et al (Regueiro, et al., 2021) in their conclusion indicate an increased amount of gas with greater data and Key sizes generated due to an increasing number of users. The Study shows that the more task heavy functions consumed significantly more gas and with an increase in the number of functions written on the smart contract, there was an increase in the gas paid, hence making task execution in the study more costly than in this study. Note should be taken that as per the study of Borgsten and Jiang, the amount of gas paid varies on the datatype used and the size of these datatypes as explained in the table below

Table 4: Gas cost and read latency associated with the basic data types in solidity (Borgsten & Jiang, 2018).

Type	Declaration	Assignment	Change	Read (μ s)	Latency
Bool	68,653	90,560	13,334	5,012.44	
Bytes1	68,653	91,097	26,680	4,736.13	
Bytes32	68,589	91,684	26,431	4,748.17	

<i>Uint256</i>	68,653	91,255	26,414	4,934.6
<i>Uint8</i>	68,525	90,564	26,667	4,827.19
<i>Bytes(array)</i>	68,653	125,083	32,106	6,762.11
<i>String</i>	68,653	146,095	37,194	7,255.62

This indicates that the use of heavier data types such as strings and arrays in solidity incur a higher cost in computing and a higher read latency as well. This leads to a significant growth in the cost of transactions on the blockchain. In this study we made use of strings due to the recording of encrypted financial transactions of alphanumeric values of large sizes, this made the other datatypes such as Boolean and Uint256 amongst others impossible to use as it did not have the required data size. This led to an increased amount of time to fetch records, and an equal increase in gas prices per record of transaction.

Finally, it is evident from the study that when blockchain and homomorphic encryption are used in securing data, there is greater efficiency compared to existing banking data authentication methods. This is noticed in the time taken in validating transactions, securing the data and the availability and integrity of this data . Given that we have not evaluated how the execution is affected by higher computational loads or concurrency, we can only reason about the scalability under circumstances where each transaction takes a constant amount of time. Blockchain has the potential to meet the requirements for protection of valuable and sensitive financial information. it needs to be applied and implemented by mainstream however, the application and implementation of blockchain needs to be adopted by mainstream banks.

The proposed technologies in this study can in many ways reduce the rate of identity theft, bank frauds and forgery of account information in the banking sector. However, due to the high gas prices for implementing and maintaining the blockchain, the use of blockchain though a viable authentication method still has a long way to go in its regulations, rates as well as its power efficiency as the mining of blocks is not presently energy efficient.

5.2 Conclusions

In conclusion, this study had three objectives, firstly building a working prototype as proof of concept for the implementation of blockchain and homomorphic encryption and this was done using decentralized application which made use of homomorphic encryption by use of the paillier-BigInt library to generate a public key and private key for multiple users which was then encrypted using AES encryption and stored in a database for access. The prototype was built as a mobile first web app in ReactJS with an api built with NestJs used in making calls to the server.

Using timepoints which were variables created and assigned the value of the time at its declaration, we were able to track the difference in code execution which we used in benchmarking the query time for executing banking transactions on the prototype with existing studies and systems in the same field. From these results, we concluded the use of blockchain significantly reduces the time of financial transactions such as deposit withdrawals to approximately 13.5s.

5.3 Recommendations

In this section, recommendations are made based on the study carried out, earmarking areas of potential improvement on the existing project. These include,

Firstly, the use of less time-consuming data types in designing contracts to reduce the time taken to fetch processes hence reducing the gas price per transaction. Secondly, the employment of a standard Exchange rate for paying gas fees will make adoption much quicker. Thirdly, the use of user's local storage to keep their public and private keys as with the case with crypto wallets will ensure users are the sole owners of their encryption and decryption keys

One of the key setbacks for the implementation of blockchain in the banking sector is the lack of agreed upon standards by all banks currently, there exists no chosen standards or norms on which blockchains exist in our current economy. the onus is therefore on banking institutions, to create a consensual standard which will be used in implementing blockchain in the banking sector. Also, the education of employees and stakeholders of on newer technologies and its implementation is imperial to its quick adoption.

Lost or stolen credentials still poses a huge threat in the banking sectors, if for any reason the users' credentials are obtained by malicious attacker, there exists no way of flagging the transaction as fraudulent. it is therefore recommended that banks, offer more sensitization to users and employees of guarding their credentials jealousy.

5.4 Perspectives for Further Research

Being a prototype, this study provides merely a proof of concept and there exists a lot of features on which improvement can be done. Areas of potential improvement include.

- Auto Generation of Bank Statements,
- Auditing Features such that the application can audit and watch user transactions in real time ensuring they meet Anti-Money Laundering Compliance policies
- Delegation of authorization to institutions and companies to be able to view users' financial records securely
- Transaction verifications using handheld devices

Blockchain bids for a promising future in digital business and the quicker integration of banks and other financial institutions into the digital ecosystem will usher in a new wave of business leading to an improved growth in the economy and standards of living worldwide.

REFERENCES

- Aldwairi, M., & Aldhanhani, S. (2017). Multi-Factor Authentication System. *International Conference on Research and Innovation in Computer Engineering and Computer Sciences* (pp. 1-8). Langkawi Island: Aldwairi, Monther.
- ANTIC, L. N. (2022, July 06). *Cybercrime, a threat to Cameroon's nascent digital economy*. Retrieved from Antic.cm: <https://www.antic.cm/index.php/en/component/k2/item/376-cybercrime-a-threat-to-cameroon-s-nascent-digital-economy.html>
- Antonopoulos, A. M. (2014). *Mastering bitcoin: Unlocking digital cryptocurrencies*. Newton: O'Reilly Media, Inc
- Azrou, M., Mabrouki, J., G. A., & Farhaoui, Y. (2021). New enhanced authentication protocol for internet of. *Big Data Min Anal.*, 1–9.
- Bani-Hani, A., Majdalweih, M., & AlShamsi, A. (2019). Online Authentication Methods Used in Banks and Attacks Against These Methods. *Elsevier B. V.*
- Borde, H. S. (2022). *An Overview of Trees in Blockchain Technology: Merkle Trees and Merkle Patricia Tries*. Cambridge: University of Cambridge.
- Bunea, S., Kogan, B., & Stolin, D. (2016). Banks versus FinTech: At last, it's official. *Journal of Financial Transformation*, 122-31.
- Buterin, Vitalik. 2015. *On Public and Private Blockchains*. *Ethereum Blog*, Crypto Renaissance Salon. August 7. Available online: <https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/> (accessed on 18 October 2020)
- Byzantine Fault Tolerance Problem* . (2022, July 06). Retrieved from BINANCE ACADEMY: <https://academy.binance.com/en/articles/byzantine-fault-tolerance-explained>
- Christensen, Clayton M., Michael E. Raynor, and Rory McDonald. 2015. What Is Disruptive Innovation? Harvard Business. Available online: <https://hbr.org/2015/12/what-is-disruptive-innovation> (accessed on 10 July 2022)
- Cristina Regueiro, Iñaki Seco, Santiago de Diego, Oscar Lage, Leire Etxebarria,. (2021). Privacy-enhancing distributed protocol for data aggregation based on blockchain and homomorphic encryption. *Elsevier*, 10-11.
- Cameroon - Data Protection Overview*. (2022). Retrieved 4 July 2022, from <https://www.dataguidance.com/notes/cameroon-data-protection-overview>
- Daluwathumullagamage, D. J., & Sims, A. (2021). Fantastic Beasts: Blockchain Based Banking. *Journal of Risk And Financial Management* , 4-6.
- Deloitte LLP. (2016). *Blockchains Applications in Banking*. London: Deloitte.
- Derrick R. (2013), *What Is Federated Identity?* Retrieved 16 February 2022, from <https://www.sciencedirect.com/book/9780124071896/federated-identity-primer>
- Digital 2022: Cameroon — DataReportal — Global Digital Insights*. (2022, July 04). Retrieved from DataReportal — Global Digital Insights: <https://datareportal.com/reports/digital-2022-cameroon>
- Eugene M. K. & Theodore F. C (2001). Willie Sutton is on the internet: Bank Security Strategy in a shared Risk Environment, 5 NC. Banking Inst. 167.
- FFIEC Information Technology Examination Handbook, E-Banking Booklet, August 2003
- Gentry, C. (2009). *A Fully Homomorphic Encryption Scheme* . New York : Symposium on the Theory of Computing .
- Godfrey-Welch, D., Anderwald, S., Lagrois, R., & Law, J. E. (2018). Blockchain in Payment Card Systems. *SMU Data Science Review*.
- Godfrey-Welch, D., Lagrois, R., Law, J., Scott Anderwald, R., Engels, D. W., & Engels, D. W. (2018). Blockchain in Payment Card Systems. *SMU Data Science Review*, 3-5.
- Guo, Y., & Liang, C. (2016). Blockchain application and outlook in the banking industry. *Financial Innovation*.
- Haferkorn, M., & Diaz, J. M. (2015). Seasonality and Interconnectivity within Cryptocurrencies—An Analysis on the. *Springer International Publishing*.
- Hall, J. et al. (2022). Azure AD Multi-Factor Authentication overview. Retrieved 16 February 2022, from <https://docs.microsoft.com/en-us/azure/active-directory/authentication/concept-mfa-howitworks>
- Ismail, R., *Enhancement of Online Identity Authentication Though Blockchain Technology*. 2017: Malaysia
- Jansen W. (2003). Authenticating users on handheld devices. *Proceedings of the Canadian Information*.
- Jon, N. 82 A.B.A.J. 94 (1996). Anytime, Anywhere, anyway. Online banking offers greater convenience and easier financial planning. A.B.A. Journal 94.

- Kim, H., Eun Kim, K., Park, S., & Sohn, J. (2021). E-voting System Using Homomorphic Encryption and Blockchain Technology to Encrypt Voter Data. *arXiv.org*.
- KOSKOSAS, I. (December 2011). THE PROS AND CONS OF INTERNET BANKING: A SHORT REVIEW. *Business Excellence and Management* , 51.
- Lim, Shu Yun & Fotsing, Pascal & Almasri, Abdullah & Musa, Omar & Mat Kiah, Miss Laiha & Ang, Tan & Ismail, Reza. (2018). *Blockchain Technology the Identity Management and Authentication Service Disruptor: A Survey*. International Journal on Advanced Science, Engineering and Information Technology. 8. 1735. 10.18517/ijaseit.8.4-2.6838
- Lawal, O., Ibitola, A., & Longe, O. (March 2013). Internet Banking Authentication Methods in Nigeria Commercial Banks. *African Journal of Computing & ICT*, 8.
- M I Awang, M.A.M., R Mohamed, An Ahmad, N A Rawi, A *Pattern-Based Password Authentication Scheme for Minimizing Shoulder Surfing Attack*. International Journal on Advanced Science, Engineering and Information Technology, 2017. 7(3).
- Masood, F., & Faridi, A. R. (2018). An Overview of Distributed Ledger Technology and its Applications. *International Journal of Computer Sciences and Engineering*.
- Measuring digital development: Facts and figures 2021. (2022). Retrieved 4 July 2022, from <https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx#:~:text=Latest%20figures%E2%80%8B%20show%20that,come%20online%20during%20that%20period.>
- Merkle, R. C. (1988). *A Digital Signature based on a conventional encryption function*. . Berlin: Springer Berlin Heidelberg.
- Nakamoto, S. (2022, July 9). *Bitcoin: A Peer-To-Peer Electronic Cash Sytem*. Retrieved from Bitcoin: <https://bitcoin.org/bitcoin.pdf>
- Nassar, M., Erradi, A., & Malluhi, Q. M. (2015). Paillier's Encryption: Implementation and Cloud Applications. *KINDI Center for Computing Research*.
- Nguyen, T. V., Nguyen, B., Nguyen, K., & Pham, H. (2019). Asymmetric monetary policy effects on cryptocurrency. *Research in International Business and Finance*.
- Norah Alwit. (2020). Authentication Based on Blockchain
- Ogburn, M., Turner, C., & Dahal, P. (2013). Homomorphic Encryption. *Elsevier B.V.*
- OMFIF, & CCBU. (2020). *The role of blockchain in banking: Future prospects for cross-border payments*. New York: OMFIF Limited.
- Papathanasakis, M., Maglaras, L., & Ayres, N. (2022). Modern Authentication Methods:A Comprehensive Survey . *IntechOpen Journals*.
- Ralph C. Merkle. *A digital signature based on a conventional encryption function*. In Carl Pomerance, editor, *Advances in Cryptology — CRYPTO '87*, pages 369–378, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.
- Rivest, R. L., Adleman, L., & Dertouzos, M. L. (1978.). On data banks and privacy hormomorphisms. *Foundations of Secure Computation,Academia Press.*, 169–179.
- Rochet, J.-C., & Tirole, J. (2003). Platform competition in two-sided markets. *Journal of European Economic Association* 1, 990 -1029.
- Section. (2022, July 12). *Understanding a 51% Attack on the Blockchain*. Retrieved from Engineering Education (EngEd) Program | Section: <https://www.section.io/engineering-education/understanding-the-51-attack-on-blockchain/#:~:text=A%2051%25%20attack%20happens%20when,and%20order%20of%20new%20transactions.>
- Shan, T., & Hua, W. (2006). Service-Oriented Solution Framework for Internet Banking. *International Journal of Web Services* , 29-48.
- Simplilearn. (2022, July 12). *What is Solidity Programming, its Data Types, Smart Contracts, and EVM in Ethereum?* Retrieved from Simplilearn: <https://www.simplilearn.com/tutorials/blockchain-tutorial/what-is-solidity-programming>
- The World Bank. (2019). *Cameroon Digital Economy Assessment*. Washington, DC: World Bank Publications.
- TC. Shan and WW. Hua,” *Service-Oriented Solution Framework for Internet Banking*,” International Journal of Web Services Research, vol.3, issue 1, 2006, pp. 29-48.
- Thigpen, S. (2005). Authentication Methods Used for Banking. *InfoSecWriters*, 16.
- Wazid, M., Das, A. K., Shetty, S., & Jo, M. (2020). A Tutorial and F orial and Future Resear e Research for Building a Block ch for Building a Blockchain-Based chain-Based . *IEEE Access* , 4-6.
- Wiedenbeck, S., Waters J., Birget J., Brodskiy, A., & Nasir Memon (2005). *Passpoints: Design and Longitudinal Evaluation of a Graphical Password*

- System. International Journal of Human-Computer Studies*, 63(1-2), 102-127
- Wenyu, X., Lei, W., & Yunxue, a. (2018). Privacy-Preserving Scheme of Electronic Health Records Based on Blockchain and Homomorphic Encryption. *Journal of Computer Research and Development*.
- Wick, D. (2022, July 09). *From Banking and Data Security to Compliance: Blockchain Grows Well Beyond its Cryptocurrency Roots*. Retrieved from <https://www.finextra.com/>: <https://www.finextra.com/blogposting/18383/from-banking-and-data-security-to-compliance-blockchain-grows-well-beyond-its-cryptocurrency-roots>
- World Bank, G. (2017). *Distributed Ledger Technology (DLT) and Blockchain*. Washington, DC: World Bank Publications.
- Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2018). *NISTIR 8202 Blockchain Technology Overview*. Gaithersburg: National Institute of Standards and Technology. Retrieved from <https://doi.org/10.6028/NIST.IR.8202>
- Zhao, L., Zhang, J., & Zhong, L. (2022). A blockchain-based transaction system with payment statistics and supervision. *Connection Science*.