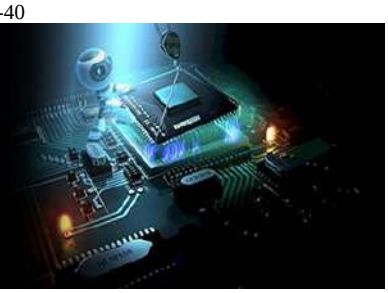
International Journal of Engineering in Computer Science



E-ISSN: 2663-3590
P-ISSN: 2663-3582
IJECS 2021; 3(2): 33-40
Received: 05-10-2021
Accepted: 09-11-2021

Rohit Panchal
Department Computer Science and Engineering, HMR Institute of Technology Delhi, India

Sarthak Sharma
Department Computer Science and Engineering, HMR Institute of Technology Delhi, India

Aman Negi
Department Computer Science and Engineering, HMR Institute of Technology Delhi, India

Ankit Arora
Department Computer Science and Engineering, HMR Institute of Technology Delhi, India

Isha Gupta
Department Computer Science and Engineering, HMR Institute of Technology Delhi, India

Correspondence Author;
Rohit Panchal
Department Computer Science and Engineering, HMR Institute of Technology Delhi, India

Secure and transparent password storing mechanism using blockchain

Rohit Panchal, Sarthak Sharma, Aman Negi, Ankit Arora and Isha Gupta

DOI: https://doi.org/10.33545/26633582.2021.v3.i2a.66

Abstract
Blockchain Technology has emerged in last decade and despite being a new Technology it is already taken over the Internet. Blockchain Technology serves a major role for the Project that works on trust, because of its transparency and not being centralized. We would like to expand the Blockchain Technology towards Password Managers. Password Managers are applications that can help you store all online credentials with at most security. In this Paper we will be discussing how the Adaptation of Blockchain Technology can be used in the field of Password Managers to make them secure and how the Blockchain Technology can help with the existing issues with Password Managers, and how we can create a system of nodes using the Blockchain Technology to efficiently and securely save the credentials.

Keywords: Blockchain, security, password managers, advanced encryption standard

1. Introduction
In the last decade Blockchain has emerged as a revolutionary Technology. The Blockchain was first introduced with the Bitcoin. In a paper published by Satoshi Nakamoto, along with Bitcoin a revolutionary technique was introduced which is Blockchain. In the last decade, after the introduction of Bitcoin and Blockchain, there has been enormous numbers of research has been done. The Blockchain is not only limited to Bitcoin or Cryptocurrency but can be extended to any field that demands Privacy, Security and Trust.
Blockchain is a decentralized Technology, which means there is no one having the ownership over the system, but many nodes come together to run this system together. Blockchain provides Transparency, Security, Privacy, Scalability and a lot more.
Password Managers are a online or offline program where you can store all your online Login Credentials, with the evolution of technology and most of our day to day task are depended on it, we are now online, and to verify our identity we need a Username and Password. Technically all the online applications or website need you to have a Login ID and Password. What you can do is, use a single set of Login ID and Password for all the online services. But few of these online services does not store your credentials with security and hence sometime a data breach occurs. If you are using same credentials everywhere, all of your online services are compromised with one single data breach. Hence, it's risky, another way is to use different credentials to for every service, but it will be very confusing and with greater number of credentials it will be impossible to remember all of them at some point. Therefore, we need a program which will store all our credentials and to access that program we only need to remember only one Login ID and Password and rest is safe with the program. Going ahead, we'll see what a Password Manager is in more details, and we will create a model to store all online Credentials using Blockchain and which are accessible through one Master ID and Master Key.

Literature Survey
As the number of Internet users is increasing daily, as per the recent report, it is estimated around 4.88 billion people around the world use the internet in October 2021, almost 62 percent of the world's total population. This number is still growing too. On the other hand, it has become the best place for many cybercriminals. India recorded around 50,035 cases of cybercrime in 2020, with an 11.8% surge in such offences over the previous year, and internet users increasingly employ longer passwords to provide adequate cyber security.

As a result, it is becoming increasingly difficult to track the assemblage of system usernames and passwords needed to access information systems for a typical business or personal transaction. The credit reporting service Experian found that the average user has 26 online accounts but uses only five different passwords. For users between the age of 25 and 34, the average number of accounts jumps to 1600. There seems always to be a dilemma – On the one hand, if we use the same password to access all our accounts, we may save much effort. However, the concern is that it is highly unsafe. We could have different passwords to access different systems, but another question appears – can we remember so many passwords.

To overcome this issue Password manager was introduced. First and foremost, we must define what a password manager is. Password managers are programs used to generate, encrypt, and store passwords for a client-side user. All that is required of a user is to remember one master password and user name. Such software will increase security, and Passwords will be stored on the local machine itself or some hosted server. In some cases, they may be hosted on cloud servers to ensure more security for the parties involved.

The earliest work for password managers comes from Luo and Henry in 2003, Where he demonstrated a proof of concept and implementation of a more effective password manager, compared to Microsoft Passport. In 2005, Halderman et al.'s work comprised the proof of concept and implementation of a password manager in the web browser, where an example implementation in Firefox was given in his work. Silver, Jana, Chen, Boneh, and Jackson made remarkable contributions to the auto-fill feature found in popular password managers such as LastPass, KeePass and those implemented in web browsers such as Google Chrome and Safari. They found critical vulnerabilities that abused the auto-fill feature; such attacks include password sync exploitation and injections. Their work would help greatly influence the policies auto-fill executes.

There are different types of password managers available to internet users. Some are built for web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge, while few serve as self-standing programs with the capability of web-browser integration. Some strictly enforce strong master passwords, while others do not. Few of them have integrated multi-factor authentication, which is very important to security. Let us dive into some password managers:

1. Passbolt is an open-source password manager supported by the GitHub community, which Kevin Muller initially developed along with Diego Lendoiro, Remy Bertot, and Cedric Alfonsi. Passbolt only runs on Firefox and Google Chrome. This is due to them still being in alpha Development. It is written in Javascript, PHP and Shell. It currently uses open PGP for its encryption standard. Passbolt is not a very friendly Passport manager, and no security audits on their products exempt an audit of OpenPGP, which Lack many essential features that other password managers have. There is also no enforcement of a strong master password which is in itself a significant security risk.
2. Encryptr by spideroak, which Tommy Williams initially developed, is a cross-platform password manager, e-wallet and note holder. Encryptr is written in Javascript, HTML, CSS, JSON, and XML. Encryption standard was built using the Crypton framework. Encryption and decryption are assured using AES-256 Galois/Counter Mode. Encryptry is the most minimal and incredibly simple, it does not require the use of an email, yet you can retrieve the same data from other devices. It has no strict enforcement of strong passwords and generated passwords have a defaults length of 12 characters.
3. In a Paper by Daniel Tse, along with Kecong Huang, Bin Cai and Kaicheng Liang, they discussed how we can create a model using Blockchain. Blockchain technology consists of two essential technology parts. One is the asymmetric encryption techniques, and the other is the decentralized distributed system. Asymmetric encryption enables each block of data to be linked together by cryptography. These two parts make blockchain a point-to-point platform. Remembering large and different Internet accounts and passwords is difficult. The blockchain password keeping system is based on blockchain technology. The system first encrypts the user's password by asymmetric AES-128 key, then encrypts the AES-128 key by an asymmetric key, called Master Public key, and finally stores both the symmetric AES-128 key and the asymmetric Public Master key in the block.

Blockchain

A. Overview of blockchain

The origination of Blockchain was initially designed for the digital currency Bitcoin as it permits digital data to be distributed and also secures it. However, it has more utilization and applications in this industrial world, which is changing quickly. It contains nodes (computers), and each node contains a copy of the digital ledger. The Blockchain is a digital ledger of similar data records and transactions spread across the whole webwork of computer systems. The database that the blockchain use is not centralized as well as it is not stored in a single location and is constantly updated. Hence, it is a decentralized database controlled by several participants and permits them real-time access, transparency, and data security. Pillars of cryptocurrencies are these decentralized systems, and the technology used behind them is Blockchain. Blockchain accommodates several transactions, and whenever a new transaction happens, its record is appended to each participant ledger. It has several applications that cover various fields, including financial services, Health Industry, Internet of Things (IoT), and many more. Therefore, it is one of the promising as well as demanding technologies in today's world.

B. Architecture of blockchain

The Blockchain architecture contains various components which have different functionalities.

1. **Node:** Any user or computer in the blockchain which transfers information regarding transactions is called a node.
2. **Transaction:** It stores the record in the blockchain and is the smallest building block of the entire blockchain system. It contains the recipient and the sender's address and a particular value. The owner publicly declared the transaction to the network by digitally signing the hash value, which contains a record of the previous transaction and the receiver's public key.
3. **Block:** Block is a data structure that carries transactions, and it consists of several components

roughly categorized as block header and the block body. It is spread across all the nodes in a network.

4. **Chain:** The segments of a block in a particular order are called a chain. It mainly consists of four types, Public Blockchain, Private Blockchain, Consortium Blockchain, and Hybrid Blockchain.
5. **Miners:** Miners are the nodes that add the new transaction to the giant distributed public ledger of existing transactions and carry out the block verification process.
6. **Consensus:** Consensus acts as a backbone of the blockchain, which deals with fault-tolerant mechanisms and provides rules to carry out blockchain operations.

C. Components of blockchain

1. Node Application: Among the logical components of the blockchain ecosystems, it is the first one that describes that every internet-connected computer must have this application to be a part of the blockchain ecosystem. It is generally categorized under two types that is Partial Node and the Full Node.

a. **Partial Node:** It is a lightweight node having low computational power and storage, and hence for a particular transaction, it preserves only its hash value.
b. **Full Node:** For a particular transaction, these nodes can validate and preserve its full copy and reject it.

2. Ledger: It is a digital database where the currency interchanges among the different nodes. Further, it is categorized between Public and Distributed Ledger.

a. **Public Ledger:** It is similar to the database of the bank records, which is transparent and open to everyone. Anyone can perform read/write operations.
b. **Distributed Ledger:** It uses independent nodes to maintain the local copy of the database. They synchronize and record static data like financial transactions having no central administration functionality.

3. Wallet: It is a digital cryptocurrency wallet that quickly exchanges funds and carries secure transactions among different parties. It contains both public and private keys and can be easily attainable on any of the web devices. Further, it is categorized between hot wallets and cold wallets.

a. **Hot Wallets:** They are online wallets which store private keys in the cloud-like MetaMask Wallet, and hence cryptocurrencies exchange transfer is high-speed.
b. **Cold Wallets:** They are offline digital wallets that carry high-security transactions and store private keys on paper documents (QR Code). For example, Trezor.

4. Nonce (Number only used once): It is an arbitrary 32-bit pseudo number generated using trial and error and is appended to the encrypted and hashed block. It is a non-repeating number that uses authentication protocols to validate a transaction.

5. Hash: It is developed in the block header and used in blockchain computation based on some information. It generally takes variable-length input and produces fixed-length output. Security of hash functions are increased using cryptographic hash functions, and it exhibits some properties like collision-resistant and Puzzle friendliness.

D. Types of blockchain

1. **Public Blockchain:** From its name itself, it is open to the public without any specific authorization and restrictions. Anyone can participate in this network and can efficiently perform read and write operations. Also, once the entry is validated, no one can change it, which implies it is immutable and decentralized.
2. **Private Blockchain:** It is accessed from permission under the governing body. Transaction history in these kinds of blockchain is private on blockchain ledger but decentralized within a close ecosystem. To perform read and write queries, its levels of access are different.
3. **Consortium Blockchain:** It is also called a federated blockchain as multiple organizations with permission govern this network. Consortium blockchain has low transaction costs. Since the platform is permission and not everyone has access to it, it produces a faster output, unlike in public blockchain.
4. **Hybrid Blockchain:** This Blockchain is revolutionary as it contains public and private blockchain features. It is one of the leading blockchain platforms as it is customizable and provides security, transparency, and integrity. Some of its real-world applications are the XRP token and Ripple Network.

E. Consensus algorithms

Consensus Algorithms are the set of protocols that verify all the transactions carried out by nodes in a network. All the parties decide to come to a similar agreement on the present state of the ledger. In distributed computing systems, they establish trust among unknown peers and also maintain security and integrity.

1. **Proof of Work:** This algorithm is a random process with low probability, which creates a new block and computes a hash value based on trailing zeroes called a nonce to validate the transactions.
2. **Proof of Stake:** Proof of Stake (PoS) is the most popular alternative to Proof of Work (PoW). Ethereum's consensus has moved from PoW to PoS. Instead of investing in costly gear to solve a complicated problem, validators invest in the system's currency by locking up part of their coins as a stake in this form of the consensus process. All of the validators will then validate the blocks. Validators will validate blocks by betting on them if they find one that they believe can be added to the chain. Validators get a return proportional to their bets based on the actual blocks uploaded to the Blockchain, and their stake increases proportionately. Finally, depending on their economic stake in the network, a validator is picked to produce a new block. As a result, PoS encourages validators to achieve a consensus via an incentive mechanism.
3. **Proof of Burn:** Instead of investing in costly hardware, validators in PoB 'burn' coins by sending them to an address where they are permanently lost. Validators acquire the right to mine on the system based on a random selection procedure by committing the coins to an unreachable address. As a result, validators have a long-term commitment in return for a short-term loss when they burn tokens. Miners may burn the native money of the Blockchain application or the currency of an alternate chain, such as bitcoin, depending on how the PoB is implemented. The more coins they burn, the

more likely they will be chosen to mine the next block. While PoB is an attractive alternative to PoW, it wastes resources inefficiently. Furthermore, it is questioned if mining power is merely given to those ready to burn more money.

4. **Proof of Capacity:** Validators are meant to contribute their hard drive space instead of investing in costly gear or burning coins in the Proof of Capacity consensus. Validators with more excellent hard drive space have a higher chance of choosing to mine the next block and winning the block reward.

Despite the Consensus Algorithm, to complete the transaction, a particular group of people, called miners, compete against each other, and they are rewarded for creating a valid block. This process is called mining.

Password manager powered by blockchain

Password Managers are the need of the time since a single person can have multiple online accounts, it can be hard to remember so many passwords, and in case the person decides to use a single password for all online services, this can be risky because Data Breaches occur a lot frequently, and also there are also other ways through which you can end up giving away your password and this can be worst if you are using a single password everywhere. So, two problems are remembering many big and tricky passwords and not using the same passwords. To the rescue, Password Managers come into play.

Passwords Managers are like a wallet to store your password securely and privately. There are a lot of Passwords Managers available, but the issue is someone owns them or they are managed by someone. As a user you don't know what are the security features, they are using to store your credentials and hence you cannot be sure to trust them with your credentials. This someone can be a person, group of people or an organization. So, it can be challenging for people to trust someone with such necessary credentials as Username and Passwords. Moreover, these credentials can be of Bank Accounts or something very private.

For the rescue, we will be using Blockchain Technology. Blockchain is well known for its security, privacy, decentralized network, immutability, and a lot more. By using Blockchain Technology, we will be removing this 3rd Party due to which the trust issues occur. As blockchain is open sourced, the user can actually see the implementation, security practices that are associated with process of storing credentials. Basically, blindly trusting someone to keep your credentials, we want to establish a way with which users can review the process and trust the process to store their credentials.

A. Understanding & Idea

Password Managers are applications that can store user passwords and usernames along with the tag of the corresponding account, to facilitate users to read the stored usernames and passwords. Using a Password Manager is to record all the usernames and passwords on various online platforms and have easy access to them.

One Password for All Online Needs

A Password Manager is not limited to storing your credentials securely, but can also have other features like Generating a Strong Password, let you check if the password you are using is compromised or not, and various other features.

We will be using Blockchain to store the user credentials to solve the trust issue and rather than trusting someone with your credentials, trust the process. But since Blockchain is transparent, we can't just store all the credentials into the Blockchain directly, but rather will be encrypting the credentials first with the user master key. The user can use this master key again whenever he wishes to view his Stored Passwords.

An easy-to-use web application to interact with the Blockchain to store your credentials will also be developed. Incase, the user who want to use the platform doesn't want to get into the hard part, but trusts the system and wishes to use the platform with ease.

B. Components of password manager

The Overall Project can be better understood in 2 parts. The first part is the Blockchain itself, and the second is the Web Application for interacting with the Blockchain without technical knowledge using a GUI.

1. Blockchain-Backend

a. **Block:** Blocks are the building build block for the Blockchain. A block can contain various information. For our Password Manager we will be storing Index, Timestamp, Data, Hash, Last Hash, Nonce and Difficulty in our Block.
b. **Index:** Index will store the index of the Block in the Blockchain.
c. **Timestamp:** Timestamp stores the time at which the block is mined and added to the blockchain.
d. **Data:** Data will be an array of Username, Password and the Tag to which the credentials are associated.
e. **Hash:** Hash is the hash created using SHA-256, after hashing the information stored in the block including the Last Hash.
f. **Last Hash:** Last Hash is the Hash of the previous Block which will help us establish the concept of Chain.
g. **Nonce:** Nonce is the number which is randomly generated in order to satisfy the given difficulty.
h. **Difficulty:** Difficulty refers to the numbers of trailing zeroes in the generated hash. This helps in building the concept of Proof of Work.
i. **Chain:** The chain is the connection between the previous block and the current block. It is established by passing the Last Hash itself as a part of the Block Information to create the current Block Hash. There are 2 additional functions we need to make sure the Chain is working properly.
j. **Replace Chain:** This is required to make sure we always have latest copy of the Blockchain, when a new Block is added, replace chain will check the length of our blockchain with the longest blockchain and will update our blockchain copy.
k. **Valid Chain:** This check if the Blockchain received is Valid or Not, it will check the difficulty, Last Hash and Actual Last Hash (In case the Blockchain is tempered).
l. **Proof of Work:** Proof of work is a computational problem to make sure that a Malicious Miner can not temper the whole Blockchain, if a Malicious Nodes wants to change a specific entry in the Blockchain he will have to change each and every entry after that

Block, and this will increase the difficulty and making it more time consuming to change the next entries.

m. Publisher and Subscriber: Published and Subscriber is a way through which multiple Nodes interact with each other. When a miner mines a block, he will make a publish request, it will let all the subscribers know that the blockchain has been updated, each of them will run the validation for the blockchain and then if the block is valid, replace chain would be called, replacing the blockchain with the updated copy.

2. Web Application-Frontend

The Web Application is an optional part, but to make the Password Manager accessible to wide range of users without enough technical knowledge, a front-end Web Application can be considered. Features of this Web Application will be:

a. Master Key Generator: It will generate the master username and master key for the user to use at the time of storing and retrieving the credentials from the Blockchain.

b. Add Password: This function will allow the user to make a request to the blockchain to store their credentials.

c. View Saved Password: This will allow the user to view all of his saved credentials using their Master Credentials.

C. Working of password manager

The process can be divided into 3 Parts

1. Generating your Master Credentials: Using the Frontend Web Application, these credentials can be generated with a single click. These Master credentials will act as the key and IV for generating the AES cipher which will be used to encrypt the credentials before storing them to the Blockchain. The same master username and password can also be used to view all your saved Username and Passwords along with the Tag.

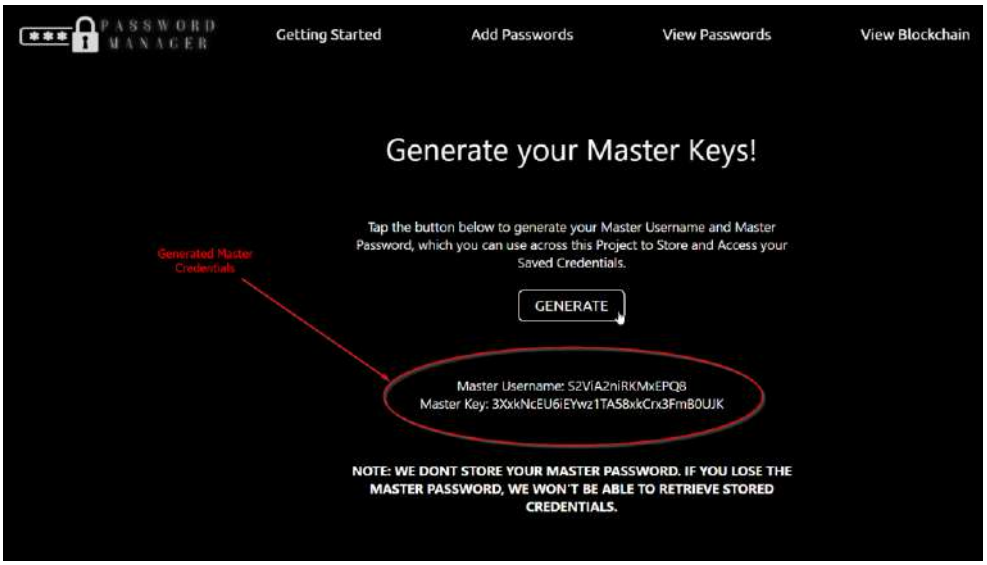


Fig 1: Generate your master keys

2. Adding Password to Blockchain: Adding your credentials using the generated master Credentials. When you make a request to add a new Block with your Password, it will first encrypt your credentials using the Master Key using the AES Algorithm, then it will add that block to your local Blockchain, and then the publisher request will be made, and each of the subscriber will receive the update in Blockchain and will make sure the chain is Valid, if the Validation is successful each node will update their local copies with the latest Blockchain Copy.

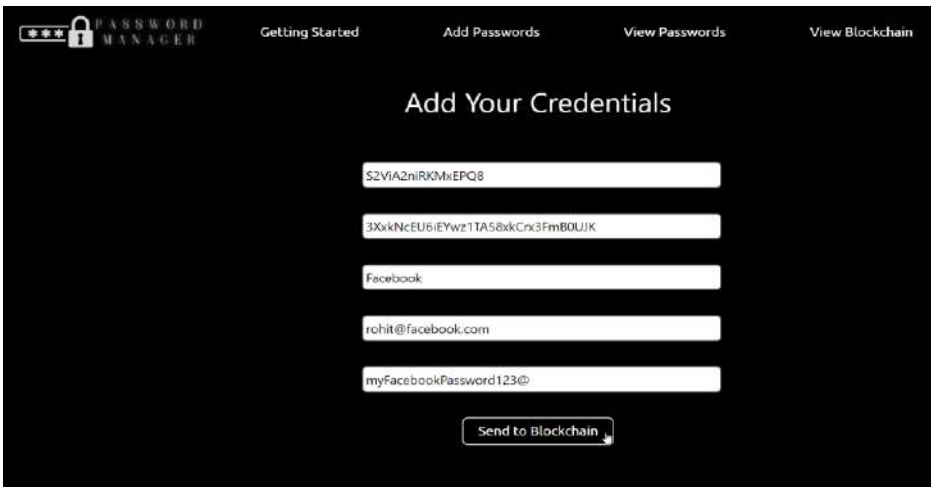


Fig 2: Add your credentials

```
[
  {
    "index": "0",
    "timestamp": "1",
    "data": [],
    "hash": "hash-one",
    "nonce": 0,
    "difficulty": 3
  },
  {
    "index": 1,
    "timestamp": 1638475349356,
    "data": [
      "Facebook",
      "a9b2d0df4ec578f874c6739383f26823108a9053093cb403bd5bd9e2759feac2",
      "e785d62e91e787d7184f2a09a542f9accd2859db5875169e558bd3f27ae52818",
      "cMT2Zz0io8wHasm9"
    ],
    "hash": "247aa69c7f24c32c2cacab198b9b21be8c26088c563a7ae006cf3aea95e3772d",
    "lastHash": "hash-one",
    "nonce": 7,
    "difficulty": 2
  }
]
```


Fig 3: Coding

3. View the Saved Credentials: To view your saved Credentials, just enter your master credentials (it will not send your credentials to someone else to get the data back rather it will get it from your local blockchain copy). It will send these master credentials to your local blockchain and find the blocks corresponding to the master username and then decrypting them using the master key and showing it on your screen.

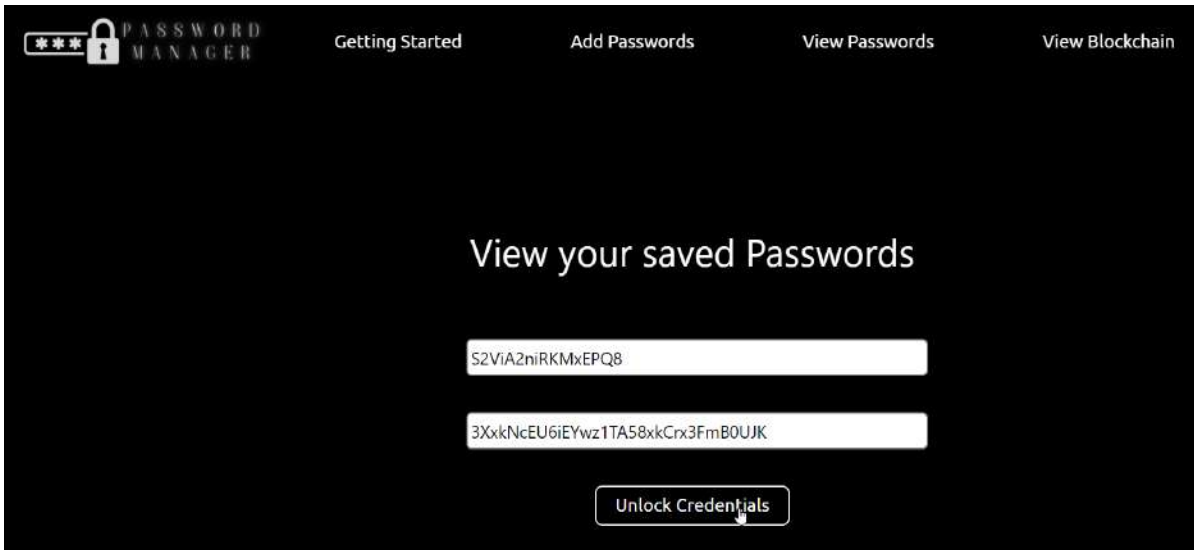


Fig 4: View your saved passwords

Fig 5: Coding 2

Even if we make a small change in Master Key, it will give the message "Error Decrypting". Therefore, credentials are safe until the Master Username and Master Password both are only accessible by the owner.

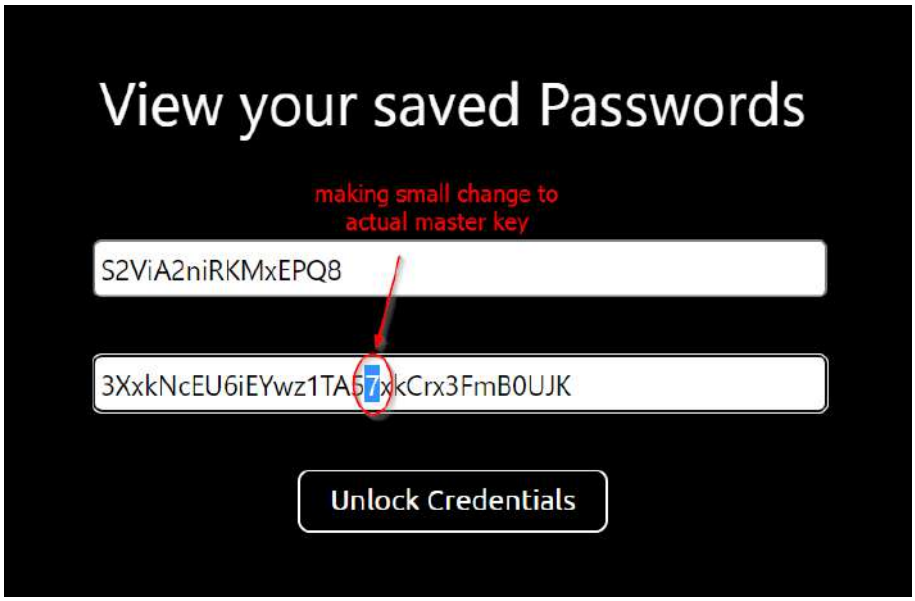


Fig 6: View your saved passwords

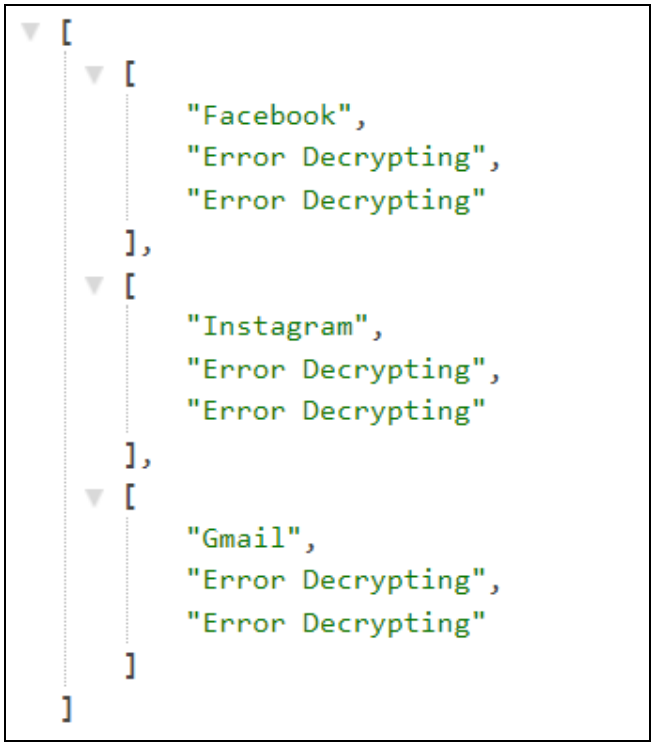


Fig 7: Coding 3

D. Why blockchain and importance of encryption?

Blockchain: The main objective of using blockchain is to make a process that is open sourced and users can have trust in the process upon verifying the process themselves. Blockchain has various other features that make it optimal for the process.

1. It is immutable, i.e., once a block is created no one can change the data of blockchain which ensures that the credentials are always safe, and you don't have to remember them.
2. It is decentralized, i.e., there is no one person controlling the Blockchain but it is open sourced, anyone can be the part of Blockchain.
3. Scalability, this implies that there is no end to the number of users that can be a part of the Blockchain and number of credentials that can be stored.
4. No down time, if a blockchain is used frequently and has a certain number of nodes, it is almost impossible that all of them can go down at a certain time. Therefore, no down time.
5. Trust, the source code can be reviewed by anyone can hence can trust the process.

Despite of so many advantages there is one risk that is involved, which is the blockchain is transparent that means anyone can view the data stored in the Blockchain, so we have to make sure the data is encrypted properly before adding it to Blockchain. Hence, the encryption is important.

We are using AES-256-CBC encryption method to encrypt the data. In CBC mode of AES encryption, each block data is XORed with the ciphertext generated from the preceding block, then it is passed through the encryption function, and the process continues. Since, for the very first block there is no ciphertext available, for that we need to pass on an Initialization vector.

Conclusion

We have discussed about Blockchain, the need of Password Manager, why Password Manager is important as well as a model of the working Password Manager on Blockchain, the features blockchain provides such as immutability, scalability, decentralization, transparency, builds sense, and much more. The model discussed consists of Blockchain, Consensus which is Proof of Work in our case, the importance of encryption, using aes-256-cbc, and finally the front-end application for easy interaction with the blockchain.

The Password managers are very necessary to make sure you are safe online, using strong and different passwords for each login credentials. There is a lot of work that can be done in the field to make Password Manager to make them even more secure.

References

1. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. https://bitcoin.org/. https://bitcoin.org/bitcoin.pdf.
2. Polshakova N. Secure password storage on the Bitcoin Blockchain. 2019. https://cs.brown.edu/. https://cs.brown.edu/research/pubs/theses/capstones/2019/polshakova.nina.pdf.
3. Brindha S, Vishnudarshan S, Arsaad S, Dinesh A. Improving Password System Using Blockchain. International Research Journal of Engineering and Technology (IRJET). 2020.
4. Tse D, Huang K, Cai B, Liang K. Robust Password-keeping System Using Block-chain Technology. 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). 2019. 10.1109/IEEM.2018.8607284.
5. Luevanos C, Elizarraras J, Hirschi K, Yeh J. Analysis On the Security And Use Of Password Managers. 2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT). 2018. 10.1109/PDCAT.2017.00013.
6. Sousi, Ahmad-Loay, Yehya Dalia, Joudi Mohamad. AES Encryption: Study & Evaluation. 2020.
7. Pitchaiah M, Daniel Philemon, Praveen. Implementation of Advanced Encryption Standard Algorithm. International Journal of Scientific & Engineering Research. 2012, 3(3).
8. Abdullah, Ako. Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data. 2017.