De-Centralized Certificate Creation and Verification using Block Chain (DCCVuB)

V Brindha Devi, R Skanda Gurunathan, N Keerthi vasan

Abstract*: *The rapid growth in the population has lead to generation of large amount of data from each individual. Each and every individual holds several physically signed documents. Currently, the documents, certificates, and contracts are all printed in papers and manually signed. It is difficult for other party say a recruiter, or a government official or any other custom officer to verify the validity of the certificates and other documents of the individual. It consumes a tremendous amount of time for validating and verifying such documents manually. Thus we propose a system to develop a Decentralized application (DApp) for implementing a Blockchain[1] to store and verify the documents. By the nature of blockchain, the documents are securely stored with high integrity, and no further modifications can be done to the blocks in the chain which in turn reduces the creation of forged documents. Also using Distributed Ledger technology(DLT)[5] and IPFS the data is decentralised so that it is readily available with integrity. Also, using MultiSig[3] concepts, the system is more secured by two step authentication. Thus, blockchain creates trust and DLT provides integrity ease of access. And with use of IPFS the DApp is decentralized4]

Index Terms*: *Document Verification, Blockchain, Certificates, Smart Contracts, MultiSig

I. INTRODUCTION

Our paper aims at providing trust to the user documents such as certifications, contracts, legal documents, identity documents, etc., stored on a blockchain in a distributed environment. Our system involves three categories of user. The Certificate Issuer, the Certificate Recipient, and the Certificate Verifier. Certificate Issuer(CI) issues a certificate or contract in the name of Certificate Recipient(CR). The issued certificate data is added to the blockchain by mining a block in the blockchain. DLT[5] implemented using IPFS[4] or Hyperledger[7] or Ethereum[9] that distributes the newly constructed blockchain to all the nodes in the blockchain[1] network. Each node verifies that authenticity of new chain and accepts or rejects it. When a Certificate Verifier (CV) wants to verify the data of the Certificate Recipient, CV computes the hash of recipient's data and compares with the hash in the blockchain. Also Asymmetric key encryption and decryption techniques are used to encrypt and decrypt the data present in the blockchain to safeguard it from eavesdropping in other nodes.

Revised Manuscript Received on October 12, 2019.

Dr.V Brindha Devi,, B.E.,M.E.Ph.D, is currently working as Associate Professor & Head of Information Technology at Sri Sairam Institute of Technology,Chennai ..

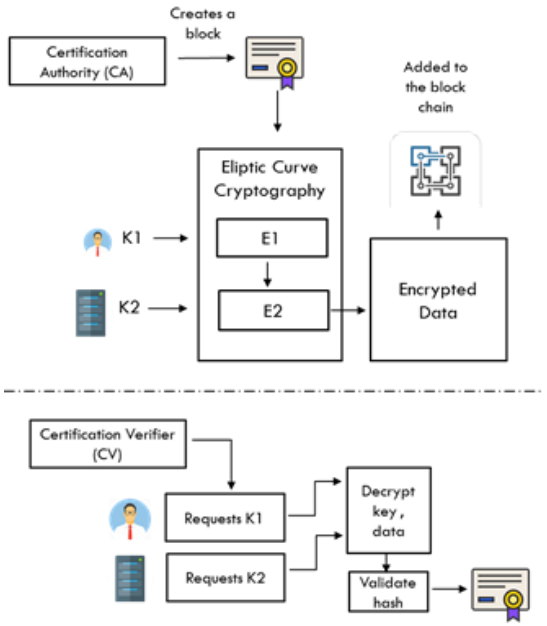
R. Skanda Gurunathan is currently working in Prodapt Solutions Pvt Ltd,. He is an alumni of Sri Sairam Institute of Technology.

N. Keerthivasan is alumni of Sri Sairam Institute of Technology. He is an active member of Institute of Engineers India (IEI).

II. EXISTING SYSTEM

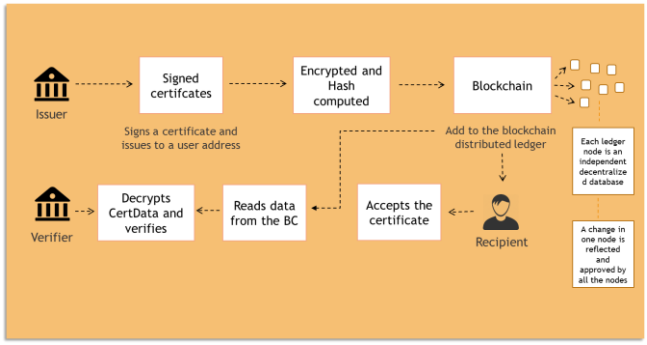
In existing system, the verification of documents is manual and data is fetched and verification is done from a centralised server. It requires lot of efforts to maintain a centralised server and at the time of verification, the server may become unavailable. Thus, relying on a centralized server for documents such as certificates doesn't guarantee availability and integrity. This we propose a system that uses a distributed system to ensure availability and blockchain is used to ensure the integrity of the documents. Also the system uses asymmetric key encryption mechanism to provide confidentiality to the data stored in the blockchain.

In Chapter III we propose DCCVuB Structure. In Chapter IV we propose methods creating user and records (mining) in the blockchain. In chapter V we propose methods for distributing the blockchain over the network using IPFS. In chapter VI we propose the process of verification of records in the blockchain. In Chapter VII we propose encryption and decryption mechanisms to ensure security of the system. In Chapter VIII we propose the implementation details of the system. In Chapter IX we propose the future work and conclusion of DCCVuB



III. DCCVUB STRUCTURE

In our system, a user is generated with a unique address and this is used for all transactions with the user. The Certificate Issuer (CI) creates a digital certificate block, this block is digitally signed by the CI using its private key. Then, a notification is sent to the Certificate Recipient(CR). This signed certificate is added to the Smart contract[2] in the block chain. CR accepts the certificate and encrypts the certificate using Eliptic Curve Cryptography(ECC) using Key K1 to obtain E1. The encrypted data is again encrypted using another key K2 which is stored and served by a MultiSig[3] key service provider servers to form E2. The Encrypted certificate's hash is computed and then added to the blockchain, computed for the proof of work and distributed to all the nodes by Distributed Ledger. At the time of verification, the keys K1 and K2 from the user and the service provider is requested and the certificate is decrypted and the hash is verified. The system uses a mobile interface to request the Key K1 and API call to request the key K2. Thus the system also provides two step verification by implementing MultiSig concepts.



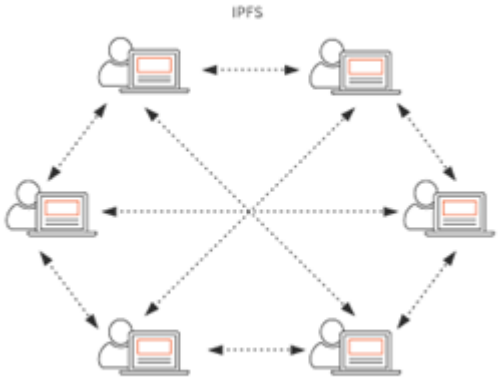
IV. MINING RECORDS TO BLOCKCHAIN

At first the user creates an account with the system. Each user is created a unique 256bit address that is computed based on the IP address of the user and the current timestamp. Each user resides in a unique IP address at a time. The system uses this point of uniqueness to create a unique hash which represents the address of the user. Also ethereum address can be used in case of using an ethereum blockchain network.

The transaction records are computed for Proof of Work, which has 5 zeros preceding the hash produced with the nounce and encrypted data E2. Once the required Proof of Work is computed, the block is ready to be mined to the blockchain[6][8].

V. DISTRIBUTING BLOCKCHAIN OVER NETWORK USING IPFS

To verify the new block and to authorize it to be added to the blockchain, the block has to be mined by 51% of nodes in the blockchain. To distribute the block and mine the data IPFS can be used. IPFS Nodes are created and each node connects with the other IPFS node, With help of this, not only the blockchain is decentralised, also the DApp the UserInterface is also decentralised. The whole system including all the decryption and hashing code is decentralised using IPFS and concurrency is achieved by replacing the old data with new mined data when it is approved by 51% of nodes in the blockchain. The user database is directly shared and for each new user added to the system, all subscribed nodes are published with new data of the users.



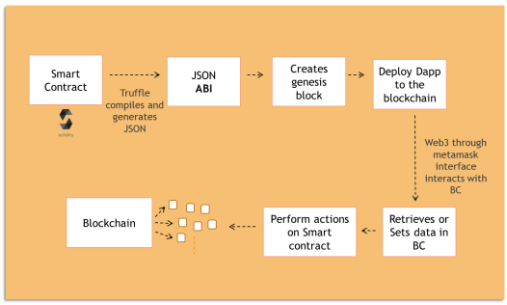
VI. VERIFICATION OF DOCUMENTS

The user is provided with a QR code that represents the block in the block chain uniquely identified by transaction hash which is produced by the mined block. Also a link is provided to access the blockchain in the DApp built over IPFS[10]. The block is retrieved from the blockchain by the Certificate Verifier portal embedded with the DApp interface and decrypted for verification. The decryption requires two public keys K1 and K2. The user is requested for the key K1 using the DApp interface and K2 is obtained from the service provider using an API call. Once the two keys are obtained the block is decrypted and the digital signature is obtained. The digital signature is verified by using the public key of the Certificate Issuer. Once the block is decrypted, the certificate data is displayed to the Certificate Verifier(CV).

VII. SECURITY

More the technology growth, more the vulnerability. Since the system is completely implemented using Blockchain and IPFS there is no possibility of altering the documents online. All the data is distributed to all the nodes and there is no centralised database to store the users too. The user database and the DApp is also distributed using IPFS. Hence to change any data in the network it requires more than 51% of nodes to be faked around the globe. Also using MultiSig concepts two step verification is enforced which requires Document Holder's knowledge to view any document.

VIII. IMPLEMENTATION DETAILS



The following are functionalities of De-centralized Certificate Digitization and Verification System:

1) Create Account on blockchain network of VerifyCert
2) Connect with the public blockchain network ethereum[9]
3) Login to the account in blockchain
4) Verify the issuer's domain
5) Create Certificate – Add to blockchain
6) List certificates from blockchain
7) Accept certificate on the Recipient's side
8) Allow a verifying user to access the certificates
9) Verify the certificate in bulk – All certificates in one frame

A. *Creating account*

Create account button on Main Screen is clicked to navigate to create account page. Automatically the address of the user is retrieved from metamask. User enters the name and password. If the user is an issuer of certificate, then the issuer name which is printed on certificate is entered. Also domain of the company is entered which requires one-time verification. A file is generated upon creation of account which must be hosted on the root directory of the domain and called using a browser. This verifies the domain ensuring there is no fake users.

Address: User address in ethereum network retrieved from MetaMask.

Name: Full name of the user. This is the name displayed on the certificate in case of certificate recipient.

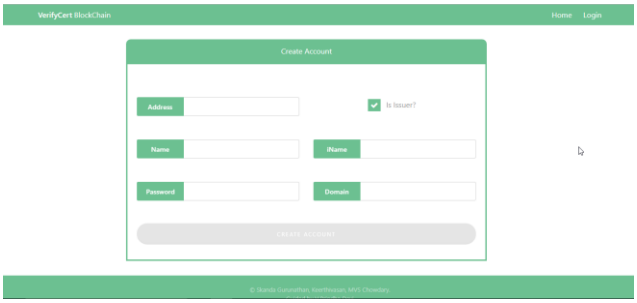
Password: To login into the system. Stored as md5 in the smart contract residing in blockchain.

Is issuer? Checked if the user is an issuer.

Issuer Name: The certificate is issued in name of this issuer name.

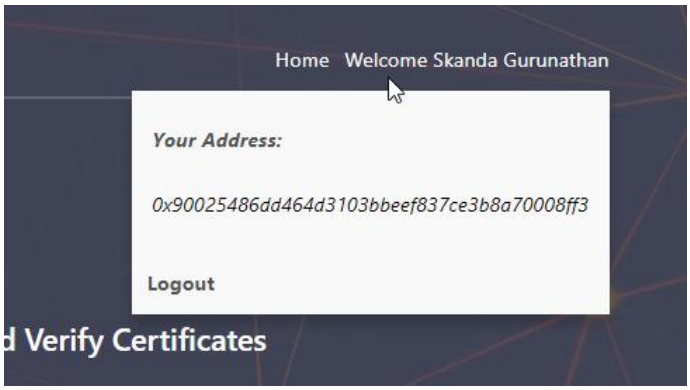
Domain: Displayed on the certificate as domain. To avoid duplication and fake users.

Creating an account executes the smart contract on the blockchain and adds the user to the blockchain.

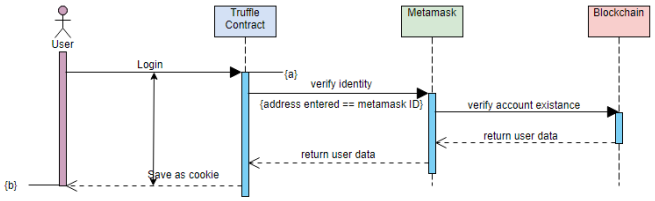


B. *Login authentication*

User enters their credentials and login to the system. Authentication occurs by a function call in the smart contract and verifies the credentials.



Once logged in the header changes to the username and address of the user is displayed. This address can be shared to issue certificate.



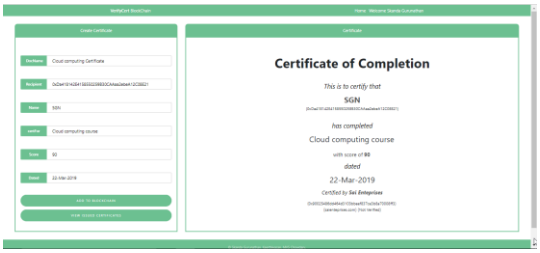
C. *Issue certificate*

This section is used to issue the certificate for a user already present in the system.

- Enter certificate details
- Enter certificate recipient address
- Add certificate to blockchain
- Track issued certificate

D. *Inputting certificate data*

The certificate data is entered into the system in the left pane. As the data is entered the preview of the certificate is displayed in the right.



E. *Adding certificate to blockchain*

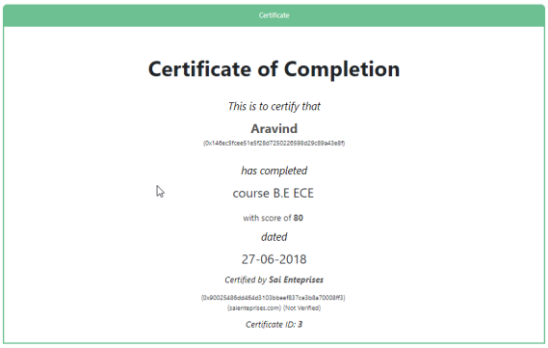
Clicking on the add certificate button calls a function on instance of the truffle contract and adds the certificate to the blockchain through MetaMask interface. For every transaction, truffle creates a truffle contract. The contract object is encapsulated with all the functions of smart contract. This is a paid transaction which consumes small amount of gas fee from the transaction account.

F. *Viewing issued certificates*

View issued certificates button opens a page for viewing list of issued certificates. Clicking on Get list button retrieves the list of Certificate issued by the current user. Selected certificate opens the certificate on preview.

G. *Certificate Preview*

Certificate states the name and course with dated. “Certified by” retrieves the Issuer name from the blockchain and displays it with domain and its verification status. Certificates with Verified domain holds more trust than untrusted certificates.



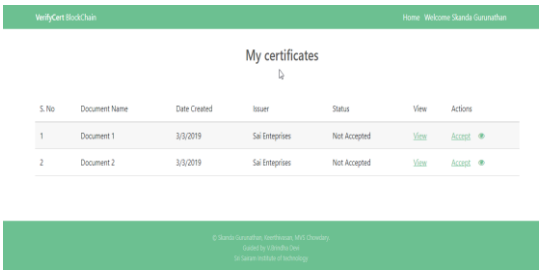
H. *My certificates*

“My certificates” page holds list of certificates that are issued to a user. The user here is the recipient of the certificate. Clicking on get certificate button executes a list function in the smart contract through TruffleContract instance and lists all the certificates issued to the current user from the blockchain.

1) View my certificates
2) Accept issued certificates
3) Approve visitors to view my certificates
4) Generate QR to view my certificates

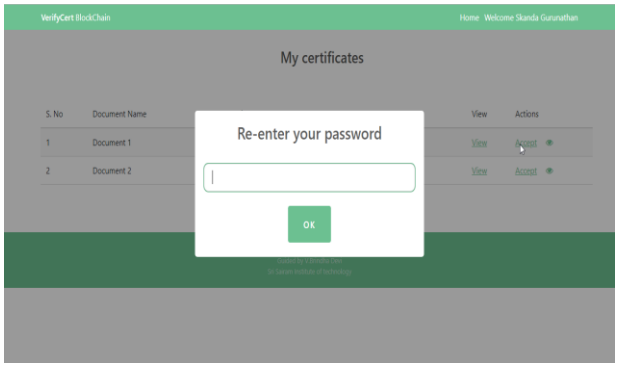
I. *Listing all certificates*

The list is presented in a data table. It contains 5 fields with header data of the certificates. And two fields for viewing and actions on the certificates. The action includes accepting the certificates and preview certificates.



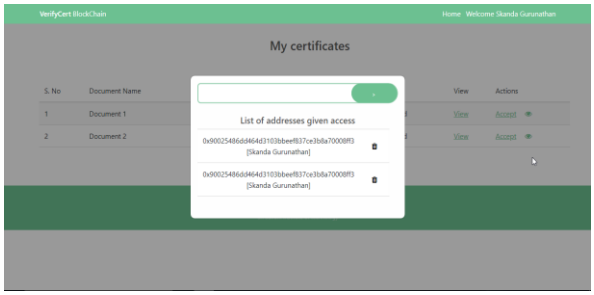
J. *Accepting certificates*

Clicking on Accept link asks the user to re-enter the password of the user. Once right password is entered the smart contract is altered through instance of truffle contract and new block with change is added to the blockchain. This is a paid transaction which causes some amount of gas fee paid from wallet.



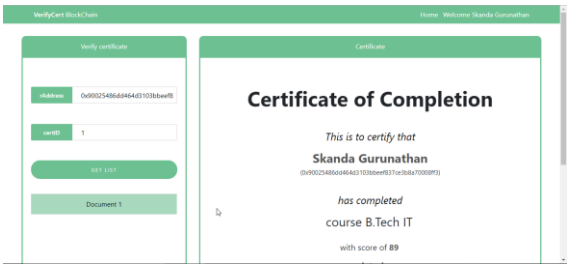
K. *User access to view and verify certificates*

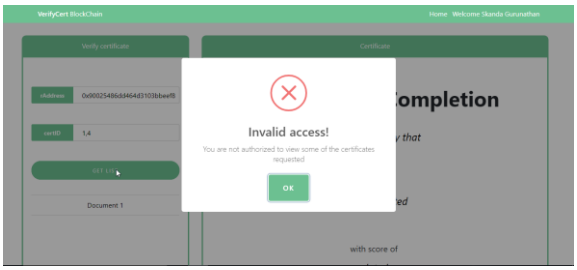
Clicking on the eye icon lists all the users who have access to view and validate the certificate on the blockchain. By default, the issuer and recipient will have the access. More users can be added to the list by typing the verifier address to the text box.



L. Verify certificates

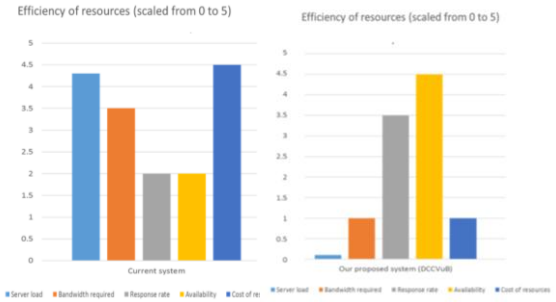
The recipient address and certificate ID is entered. If the user has the access to view the certificate the certificate list is updated and it can be clicked to view the preview. If there is no access or the certificate ID and recipient doesn't match the system throws an error. The errors and success dialogs are displayed using SweetAlert Node Plugin. It's a modal used to display alert messages with better UserInterface.





IX. EFFICIENCY CHART

This system is more efficient, takes less network bandwidth and dependability of centralized server is completely eradicated in this our system. Since it's a peer to peer system it doesn't require maintenance and load balancing. The following graph demonstrates how effectively the system balances resources.



X. CONCLUSION AND FUTURE WORK

This system provides a more secured way of storing documents and verifying it through blockchain. It uses IPFS to distribute the DApp and MultiSig[3] to provide two factor authentication. In future, when the WWW is extended with IPFS, the DApp can be easily accessed and more nodes can be accommodated.

REFERENCES

1. S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, pp. 9, 2008.
2. L. Luu, D.-H. Chu, H. Olickel, P. Saxena, A. Hobor, "Making smart contracts smarter", Proceedings of the 2016 ACM SIGSAC Conference on Computerand Communications Security, pp. 254-269, 2016.
3. Okamoto, T., "A digital multisignature scheme using bijective public-key cryptosystems", ACM Trans. Computer Systems, 1988, 6, (8), pp.432–441
4. Juan Benet, "IPFS - Content Addressed, Versioned, P2P File System", ArXiv,2014
5. Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi, Jameela Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms", Second Symposium on Network Cloud Computing and Applications, 2012
6. "Blockchain", Wikidia, [online] Available: https://en.wikipedia.org/wiki/Blockchain#cite_note-te20151031-1.
7. "AboutHypededger", Hyperledger, [online] Available: https://www.hypededger.org/about.
8. C. Christian, A. Elli, Angelo De Caro, K. Andreas, O. Mike, S. Simon, S. Alessandro, V,. Marko et al., "Blockchain cryptography and consensus", IBM Research Zurich, June 2017.
9. Vitalik Buterin, "Ethereum and The Decentralized Future", Future Thinkers Podcast. 2015-04-21, 05 2016.
10. Zheng, Y. Li, P. Chen and X. Dong, "An Innovative IPFS-Based Storage Model for Blockchain," 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Santiago, 2018, pp. 704-708J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: http://www.atm.com

AUTHORS PROFILE

Dr.V Brindha Devi, B.E.,M.E.Ph.D, is currently working as Associate Professor & Head of Information Technology at Sri Sairam Institute of Technology,Chennai .She has more than 18 years of teaching experience and her areas of specializations are Wireless Networks, Fuzzy computation, Artificial Intelligence.

R. Skanda Gurunathan is currently working in Prodapt Solutions Pvt Ltd,. He is an alumni of Sri Sairam Institute of Technology. He is an active member of Institute of Engineers India (IEI) and also received Best Student award for academic year 2018 – 19. He has published 2 journals in field of automation. His areas of specializations are Data analytics, Machine Learning, Web technology.

N. Keerthivasan is alumni of Sri Sairam Institute of Technology. He is an active member of Institute of Engineers India (IEI). His area of specializations are DBMS, Data Mining, Algorithms.