Certificate verification using Blockchain technology

Team members :

Kevin Samson - 2021A7PS0181U
Milan Sunil - 2021A7PS0009U

Contributions:

Abstract - Kevin
Introduction - Kevin
Literature review - Milan
Problem Formulation - Milan
Possible Solutions - Kevin & Milan
Problem with proposed solution - Milan
Simulation and implementation - Kevin & Milan
Conclusion - Kevin
Reference - Kevin & Milan

Table of Contents

Abstract

Blockchain is a digital ledger that is decentralized meaning that there is no one entity that is holding all the information. It is also immutable as each block contains a hash and changing the data within a block, modifies the hash of the entire block. This is mainly used in cryptocurrency like bitcoin but can also be used to track and record assets. This technology can be used to change the way academic data is stored as it is more secure. One important document for students is a degree certificate which is issued by universities and other educational institutions. This certificate is useful when applying for jobs, taking tests, and higher education. Because of this, there is a motivation to forge fake or counterfeit certificates. It is also hard to verify if a certificate is legitimate or not, as only professionals with special equipment can test the validity of certificates with special watermarks, etc. Blockchain can be used to combat this issue since it's decentralized and it is easy for anyone to verify the authenticity of the certificate. In this project we aim to fix this issue. We use Ethereum and smart contracts to deploy the app on the Ethereum network.The hash of the certificate is generated and the hash is digitally signed. This ensures that only the institutes are able to sign and upload certificates.

Introduction

As time moves on, more and more things are being digitized as technology is becoming more secure and accessible to everyone. This also applies to certificates as in the example of DigiLocker which is used to authenticate and share documents like CBSE board results. The problem with the current technology is that bad actors may be able to attack DigiLocker servers and modify data. Also DigiLocker servers may sometimes be down due to maintenance or other issues. This means that the certificates won't be accessible for an unknown amount of time. The use of Ethereum network and smart contracts mitigate this issue as they are multiple nodes and it is impossible to attack all the nodes.

Certificates are required while applying for jobs and people are required to send the physical certificate to be verified manually. This is expensive and time consuming. Thus the efficiency of verification of documents can be increased by Blockchain .

While verifying manually, a lot of time will be wasted contacting the respective universities and verifying the validity. This can be manageable for a small company but for a company with hundreds of applicants, this can be a very time consuming process.

Our objective is to create an application that can run on any platform and only requires a camera and an internet connection to verify certificates. The student or the person who wants to apply for the job can provide the certificate in both the physical copy as well as the hard copy. Even if there is no camera available, the certificate can still be verified by using the softcopy of it.

Literature review

Verification in blockchain

There has been a lot of analysis of different methods to identify duplicate documents or certificates as this is a very concerning issue. Even currently there is no ideal method to identify duplicate documents or keep them secured so we suggest a framework where it is managed by universities and higher education institutions where verification of the certificates can take place at one place.

It was implemented by Satoshi Nakamoto and was used as a distributed ledger where transactions were made using bitcoin. Basically what happens in the blockchain is that each block undergoes the process of calculation to attain its hash value. It considers the hash of the previous block concurrently. Blockchain is like a linear structure of blocks. It is a decentralized database and is more secure and faster. It is better than the current technologies. The academic certificate is very important to a student as it stands a proof that a particular student has graduated from this particular institution. The physical certificate which is a paper certificate is converted into a digital certificate and is stored in the blockchain. The presented algorithm will generate the hash code for the certificate. The issuing process of the certificate does not undergo perfect verification. This can be solved or atleast avoided by using blockchain technology which has better security and an unchangeable data storage. Blockchain technology can fight up against deceit or abuse of the documents.Ethereum comes under the blockchain network. The unique thing about it is that a user can develop his/her own application in the ethereum network. It is a decentralized network where there is a huge collection of nodes and it does not have to depend on a centralized server to process. Rather it is run by thousands and thousands of these nodes. No single node owns ethereum as it is publicly accessible and widely used by a lot of people around the world. As there is a huge issue dealing with the creation of many fake degrees so the design was refined with respect
to the algorithm of ethereum and put into the Unicoin network. This network deals with the global financial services and thus due to this process it verifies and stores the certificate related data thus leading to the creation of EUniCert which verifies digital certificates.Smart contracts are used in the ethereum network. They use the IFTTT logic which is extended as IF-THIS-THAT-THEN which basically means when one particular event takes place it initializes another event to take place. Since blockchain is comparatively transparent the employer, student, faculty all have access to the information all at the same time.It is permissionless which is a huge advantage the user does not have to wait to get permission to write their own smart contract and will be able to implement them as long as the respected user has enough ether. When building smart contracts solidity and LLL are typically the high-level scripting languages. Now solidity is widely used. Every transaction is stored in a smart contract and seen in the blockchain and promotes trust. Smart contracts are secured by cryptography making them avoid fraud. Good advantage is it reduces the risk of damaged or fraud digital degrees. Here they can verify the digital degrees thus reducing time consumption. There is full control for the graduates when it comes to who they want to share their degrees with. It gives an evident collection of academic records for all authorized parties.Solidity is used for the creation of smart contracts. After the completion of the creation of smart contracts a compiler is needed to translate the solidity program into bytecode as it can be used to store strings to prevent overflow. The ethereum virtual machine is used for interpreting the byte code. Merely by virtue of the advent of cryptocurrency like bitcoin and ethereum and the advancement of blockchain technology. Since the technology has a decentralized structure and the unchangeability of the data it provides various advantages that could change the way today's systems conduct business. Here we deal with the creation of an electronic file that operates similarly to a paper certificate figuring out its hash value and then dealing with the storage in the block. The original certificate is affixed with a QR code that is present in it. It can be used with a phone’s scanning abilities to verify if it is an original or a fraud certificate. Following the instructions are added to the ethereum blockchain. This is done from beginning to end. Transparency

in certificates entails keeping an open, append-only record of digital certificate issuance and revocation activities. The objectives of this transparent approach are to discourage fraudulent actions, promote accountability among certificate authority, and make it easier for relying parties to verify the authenticity of certificates. While it increases security and confidence in digital certificates, its adoption is hindered by issues with scalability, privacy, and regulatory compliance. In general, certificate transparency helps to create a more trustworthy and safe certificate environment, which in turn promotes confidence in online interactions and transactions.Using blockchain technology, the study on preventing counterfeiting with blockchain sought to solve the issue of fake academic credentials. They created a system that used smartphone apps for simple verification, safely stored certificate data on a blockchain, and gave certificates individual digital identification. The solution improved certificate security, discouraged fraud, and offered easily available verification methods by utilizing the immutability of blockchain technology and cryptographic techniques. For widespread acceptance, nevertheless, issues like privacy concerns and technical complexity had to be resolved. The study's overall findings demonstrated blockchain's potential to prevent certificate fraud and uphold the validity of academic qualifications.We utilize UNIC (University of Nicosia Certificate System where the student certificates are grouped and it saves these group reference hashes to minimize storage requirements on the blockchain and gives up some privacy and security in exchange for faster data access.BCDiploma for example also uses distinct data structures in smart contracts for the issuance and validation of diplomas and validates URL using the ethereum platform. Data size limitations that impact transaction costs are one of the challenges.

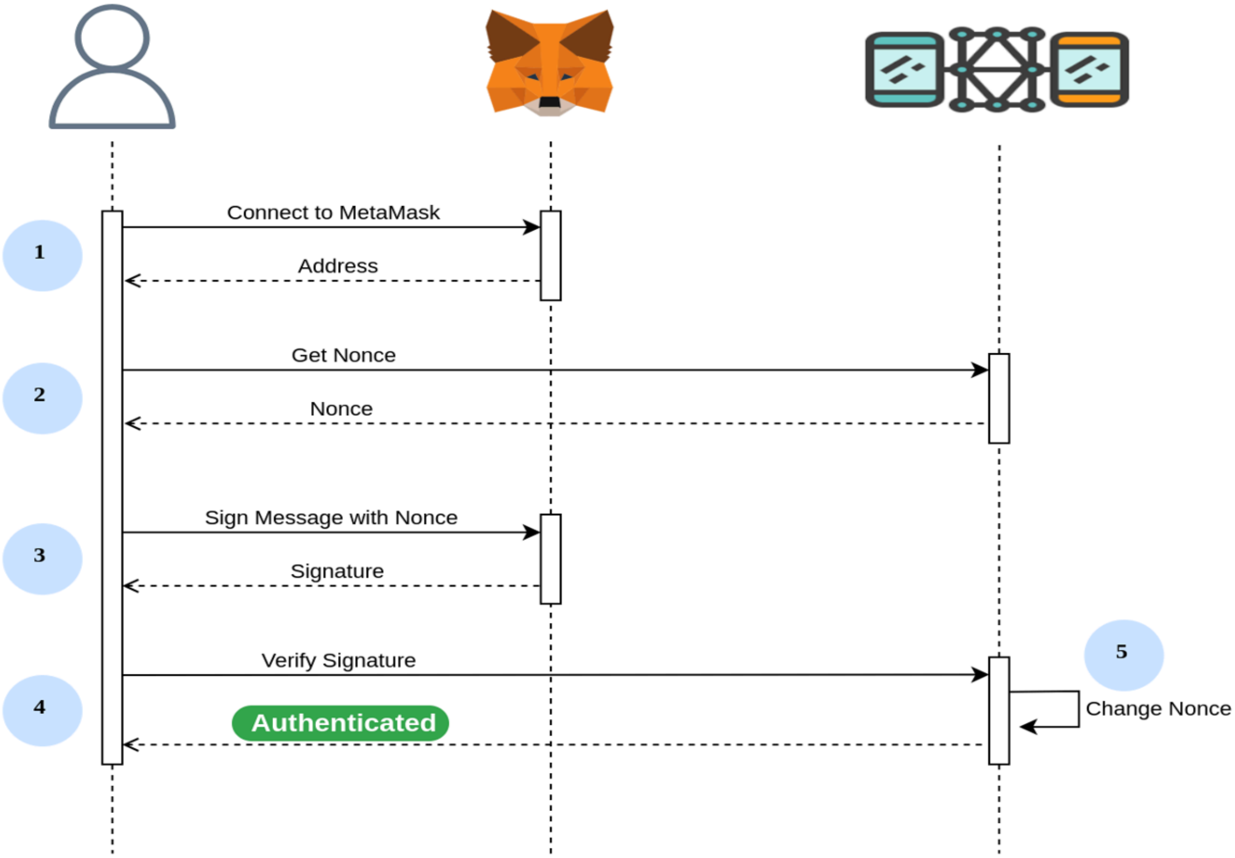


FIG 1.1 Metamask process

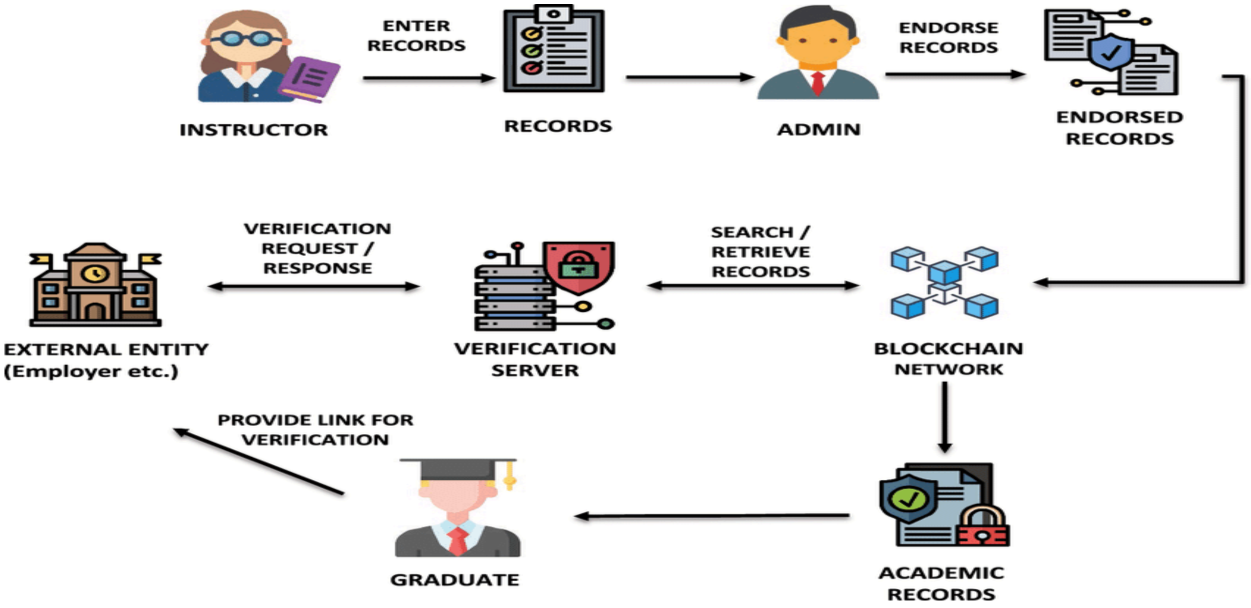


FIG 1.2 Process of how the digital degree certificates is transferred

Takeaways from literature review

With our survey, we can see that most solutions just store the hash of the certificate on the blockchain. Our method also used digital signatures to add an extra layer of security.This is done by signing the hash of the certificate with the private key of the issuer or the institute. The smart contract also has the address of the issuer hard coded in it. This means that only the issuer is able to send documents to the smart contract.

Most solutions use a meta mask account to login and verify the certificate.This is an extra step which is not necessary just for finding the hash of the certificate. Only a simple program is required for finding the hash of the certificate. Creating a meta mask account is not at all needed in our application. Only the issuer or the institute must create a wallet.

Verifying the validity is free and can be done unlimited times. Adding a new document to the blockchain is paid and it is called gas charges. This charge is done using ethereum, so the issuer must ensure that he/she has enough funds in her wallet to perform the transaction.

Most implementations use a web app so people who want to verify certificates must do so only using their website. Their website may be offline or hacked by malicious actors, rendering the website useless. We are developing an API using python meaning that different apps can be created using this such as web apps, mobile apps, etc.
The user can also easily run the code by just downloading python and running the code themselves.

We are using the Ethereum mainnet to deploy our smart contract. This network has more than 269M users which means the data stored on this is verified as compared to having a private blockchain which has only a limited amount of users.

Most implementations either focus on being fully digital or fully physical(with QR codes). Our system is flexible and can implement both physical as well as digital.We find the hash of the whole certificate image itself as compared to finding the hash of only the data inside it.

With the help of the private key of the institute as stored in the blockchain system we can ensure if the respective certificate has come from the specific university or not thus avoiding taking any further step with a forged certificate.

After checking the validity of the document, it is then stored in the blockchain which is a decentralized system thus providing secure storage and data exchange.

We are majorly using the smart contracts because it reduces the risk of moving forward with forged certificates, reducing time consumption and not needing an intermediary third party thus leading us to the major implementation of smart contracts.

The use of ethereum is implemented as it acts as an immutable ledger because the transactions are not temporary and can be used as tamper proof. After validation it is then stored in the blockchain thus ensuring more security.

Problem formulation

In the current way of how all academic certificates are handed, the copies of the certificates are directly given to students in the form of hard copies and no digital verification method exists.

Problems with physical copy

Most certificates have very basic security features like watermark, special paper etc. This can be easily forged. There is also more incentive to forge a certificate as it enables them to get into jobs which they are not trained for. This will impact job opportunities as someone who got the certificate in a legitimate manner may not get the job as compared to the one who got it in an illegitimate way. This will also affect the credibility of the company as the work provided with these illegitimate employees may not be up to standards.

Paper certificates can be damaged easily as a student may apply to many jobs and there might be wear and tear when the certificate is transferred from one person to another. There is also a chance that the certificate may get lost during transportation , misplaced in a different location etc. Natural problems like floods or fire may also physically damage the certificate.When it is not used for a long time, the paper might decay and the print may become impossible to read. Replacement of these certificates is a long and lengthy process.

They are inconvenient to carry around and share electronically. There may also be a chance where the certificate can be photoshopped and sent digitally, this makes it very difficult for the HR team to verify if it's fake or not. Especially when the university has been shut down and there is no way to verify the certificate by calling the university.

Problem with digital copy

When a university provides a digital certificate it is usually stored in a centralized server which is controlled by the university or third party. The servers may have a chance of being hacked. This can result in two scenarios, one being a data breach where sensitive information like academic record is leaked. Another scenario is that the data can be modified and tampered with. It is difficult to verify the authenticity of the real and tempered versions. Blockchain includes the timestamp in the hash value calculation so it is impossible to do so when using blockchain.

The university may also use its own protocol in storing data which is not compatible with what someone else might use. So the HR team must include this new system to view the certificates which can be difficult as there are many different universities which implies that there may be many different protocols as well.

The file format used to store the certificate may become obsolete and cannot be used in newer versions of programs as it will become incompatible.

From application to verification: The Journey of a student’s credentials

1. Admission: Students applied for university and the university the either accepts or declines the student
2. Academic journey: Students complete courses, complete assignments and take exams.
3. Graduation: Upon successful completion of all program requirements, the student graduates and a paper certificate is given to the student.
4. Verification: When the graduates apply for jobs, they might submit their transcripts to employers. The employees may contact the issuing university to verify the authenticity of the credentials .

Possible solution

We can solve this problem with the help of IPFS and smart contracts. They provide a powerful method for storing and verifying academic credentials in a secure, efficient and transparent way. A website is made where universities are able to enter the student details. This application is made on the Ethereum mainnet.

IPFS - Decentralized Storage

IPFS stands for InterPlanetary File system. It is a protocol used for storing and sharing data in a decentralized way. Unlike cloud storage which is stored in a centralized server which is controlled by a single company or entity, IPFS distributes data across a network of computers.

Anyone can join the network by running the IPFS software, contributing storage space to the network. They then are given Filecoin for contributing.

When a file is added to the IPFS system, the network creates a unique hash which is computed from the data in the file. This hash acts as a file address. If the data is modified, the hash of the file is also changed, hence this ensures data integrity

When a file is uploaded to IPFS, it gets broken down into smaller chunks. These chunks are replicated and distributed across multiple nodes in the network. This makes IPFS resistant to data loss as even if some of the nodes go down, the data remains accessible on other nodes as they have copies of the chunks as well. The data cannot be accessed by the node that is holding the file as only chunks of the file are present

To access a file stored on IPFS, a unique hash is required. When this hash is entered into the client, the network searches for nodes that possess the requested date chunks. It then combines them together and delivers the complete file.

Smart Contracts - Programmable Logic:

They are self-executing programs stored on blockchain like ethereum networks. The code runs when a set of predefined conditions are met.

Smart contracts are written in a programming language specifically designed for blockchain. For example in the case of ethereum, Solidity is used. This language provides functions to interact with blockchain and its features. The code is then compiled to byte-code and deployed to the blockchain network. The code is then deployed to the blockchain network creating a transaction that includes the compiled code and the fee. Miners on the network process this transaction and add the smart contract to the blockchain ledger.

Etherum supports virtual machines. This VM acts as a secure sandbox environment where the bytecode of the smart contract is executed. This VM runs in isolation and prevents malicious code from affecting other parts of the blockchain. To execute the code, the user needs to pay a fee called gas. The unit represent the computational power required to execute a specific operation

The process of generating a degree certificate :

Once a student is ready to graduate, the university collects all the necessary information for the certificate. The details are then entered into the website. When the submit button is clicked, it triggers the smart contract to run and execute the program. The program calculates the hash values and uploads the generated hash into the IPFS server. The hash value is calculated by taking the data of the certificate and combining it with the Aadhar card number. A QR code is generated which is attached to the certificate. The QR code contains a link which redirects the user to the website and runs a search query which finds the hash value of the certificate.

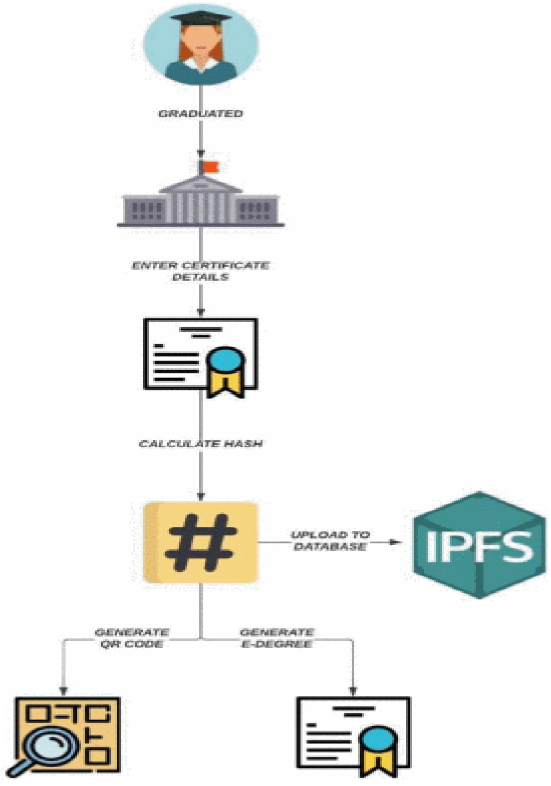


FIG 1.3 Process of generating a digital degree certificate

Verifying a Degree Certificate:

The person who wants to verify if the certificate is valid or not will first need to scan the QR code which is present on the certificate. The app then runs a search query which finds the hash value of the certificate. The app will then compare the hash of the certificate and the hash returned from the IPFS server. If the values are equal, The certificate is displayed along with the Aadhar card number. If the values are not equal, an error message is displayed.

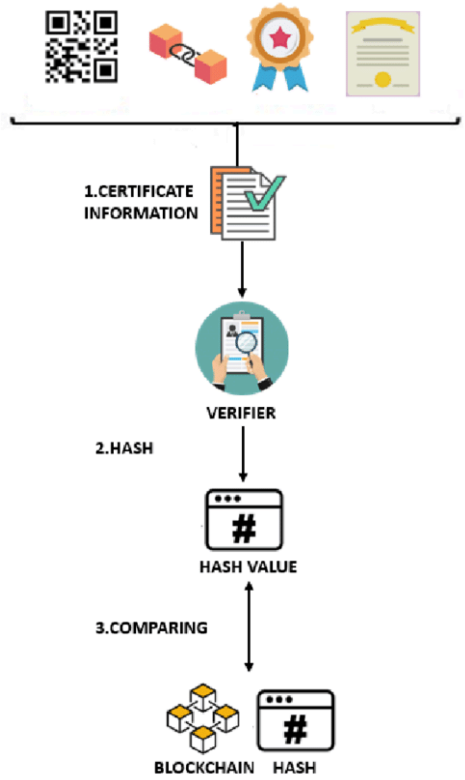


FIG 1.4 Verification of digital degree certificate

Problem with proposed solution

IPFS

- If some file is not broadly attained by some nodes then it might not be available if storing of the chunks in the nodes go offline. Unpopular data can get abandoned in the huge network.
- It does not straightforwardly benefit the users for data storage. There is no assurance that the nodes will keep storing a file for an elongated term mainly when its barely getting accessed.
- Finding the exact nodes with the wanted data chunks can be more difficult and time consuming than inquiring about a single server.
- Similarly receiving a file from the IPFS includes bringing all the data chunks from various locations throughout the network. It can be more time consuming when compared to one source.

- As it is decentralized it does not have an inbuilt component for the moderation of content. It can lead to dangerous content than can have a gateway to the network which can be worrying.

Generation of degree certificate

- Invalidation issues can be of a concern as the system does not specify on how to deny a fraud or forged certificate. If a certificate is to be withdrawn or denied it is very doubtful on how the blockchain system would apply this change.
- If in case of a centralized website the verification process can face difficulties as if a failure occurs thus if the web browser is down or denied then it makes the verification process more complicated.
- Even though the blockchain is a decentralized system sometimes used for hashing, storing and verifying it still relies on a centralized web page.

Aadhar card

- It contains highly important information like address, iris, fingerprint etc. The use of blockchain can be implemented as providing security and storing the hash value but issues can occur with respect to the privacy as a data breakage or leak can result in serious issues for the respective user.
- Students coming from a foreign country would not have an aadhar card as it is not an international card which can thus delay or prolong the verification process.
- If an issue with the system arises then it can lead to identity theft or illegal access to certain government activities that have a connection to the aadhar card.
- People within the country also do not have aadhar card, mainly the people below poverty line or who are not of the best financial status thus creating a wall in getting access to services and creating complications in certificate verification.
- Aadhar card is a centralized system and blockchain is a decentralized system thus bringing in technical obstacles because maintaining the security of the data exchanged and stabilizing the data integrity can be a hard task and thus needs precise design and execution.

Solution and Implementation

Our solution involves finding the hash of the certificate and digitally signing the hash with the private key of the university or institution. We use a programming language called solidity to write out smart contracts. We also use ganache and truffle to test and deploy our smart contract locally. Python along with the web3 library is used to interact with the smart contract.The university is also expected to have a wallet with some ethereum stored in it.

Block Diagram

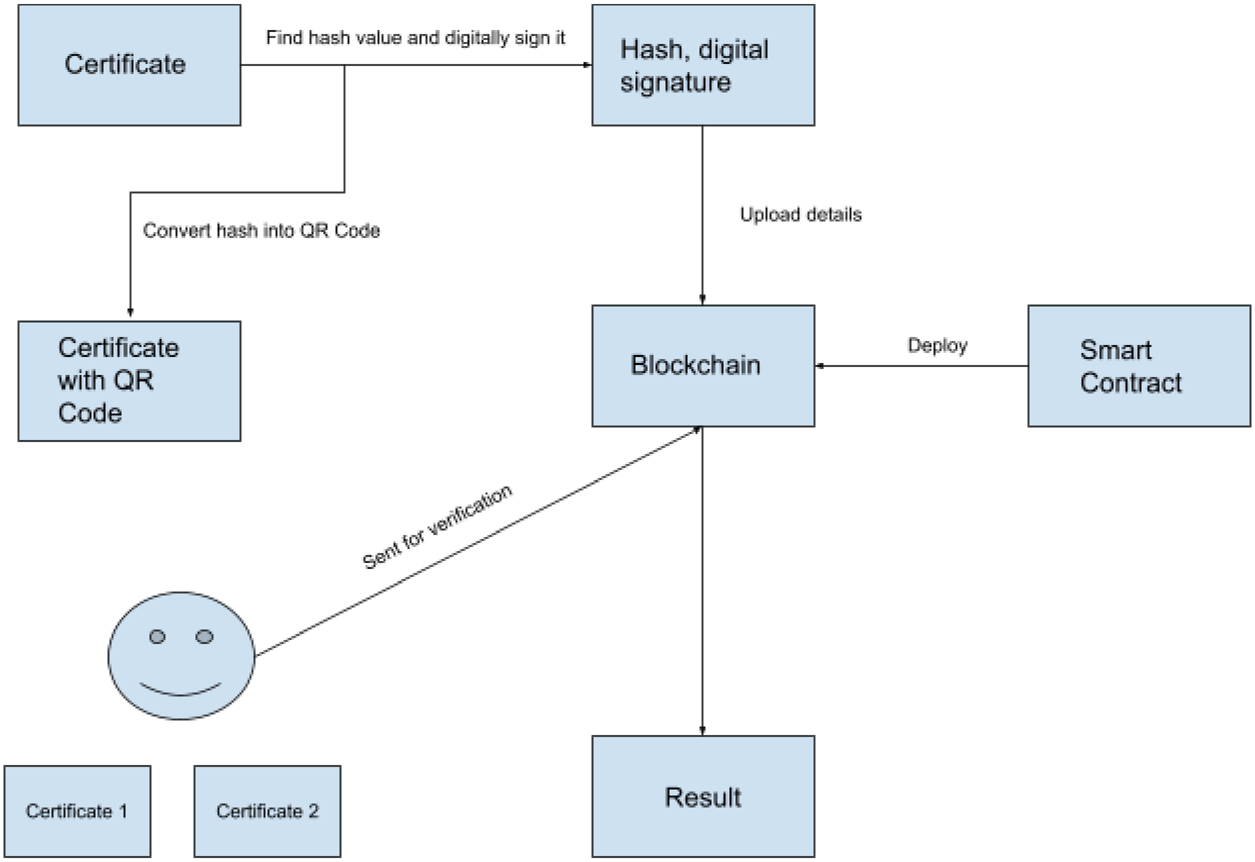
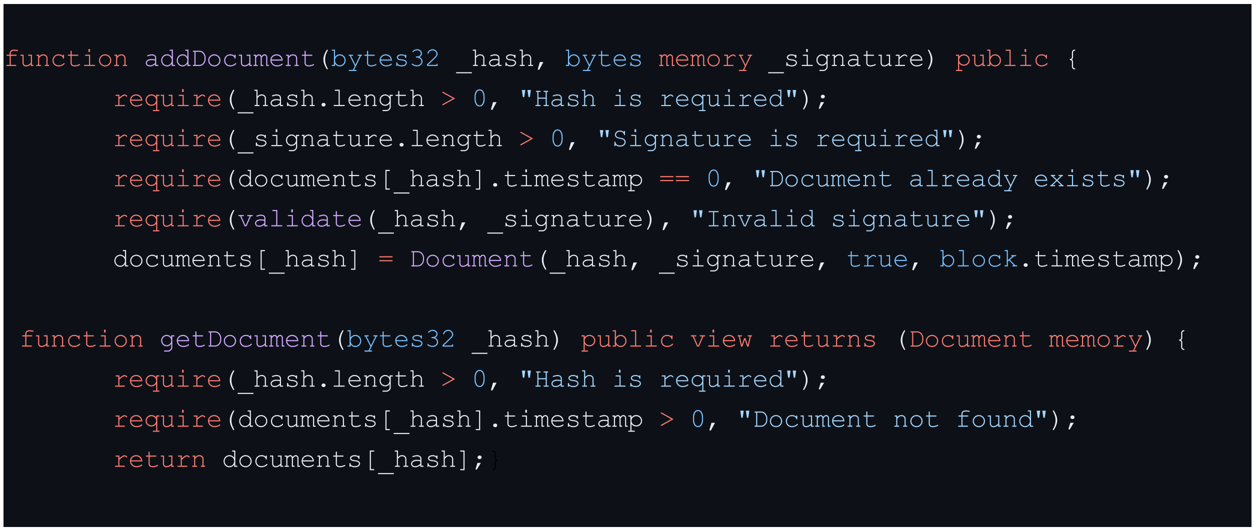


FIG 2.1 Block Diagram

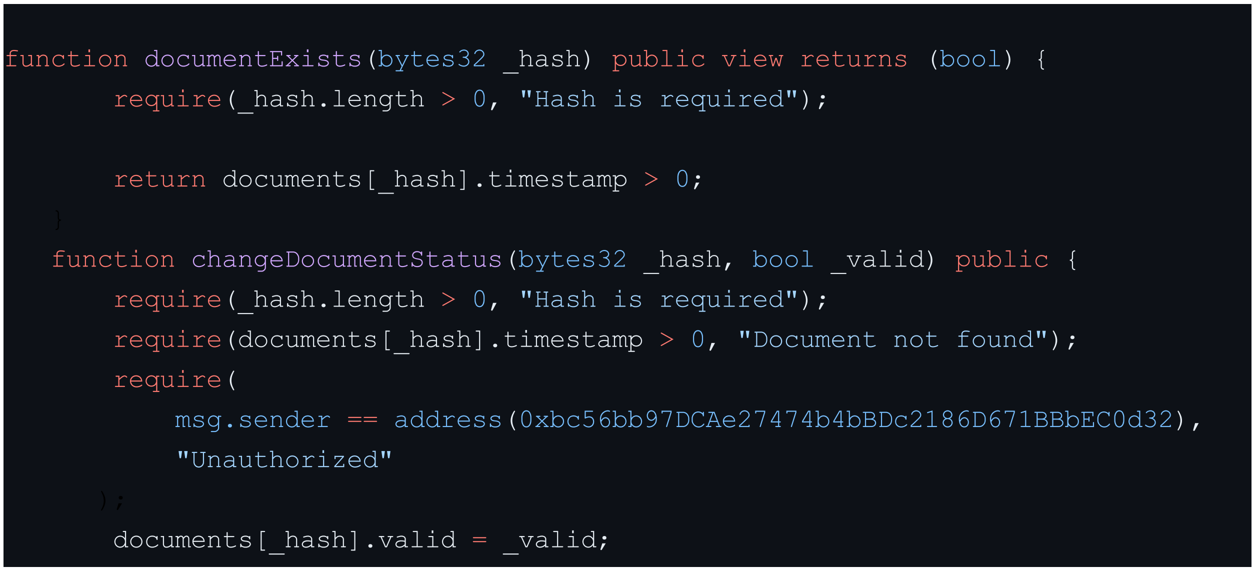
The block diagram illustrates a system for secure certificate verification using a blockchain network. It starts with certificate creation.The hash and the digital signature is uploaded to the blockchain. For verification, the certificate is submitted for verification using this smart contract. Finally, the blockchain network returns a result indicating the certificate's validity.

Smart Contract



```
function addDocument(bytes32 _hash, bytes memory _signature) public {
       require(_hash.length > 0, "Hash is required");
       require(_signature.length > 0, "Signature is required");
       require(documents[_hash].timestamp == 0, "Document already exists");
       require(validate(_hash, _signature), "Invalid signature");
       documents[_hash] = Document(_hash, _signature, true, block.timestamp);

 function getDocument(bytes32 _hash) public view returns (Document memory) {
       require(_hash.length > 0, "Hash is required");
       require(documents[_hash].timestamp > 0, "Document not found");
       return documents[_hash];}
```



```
function documentExists(bytes32 _hash) public view returns (bool) {
       require(_hash.length > 0, "Hash is required");

       return documents[_hash].timestamp > 0;
   }
   function changeDocumentStatus(bytes32 _hash, bool _valid) public {
       require(_hash.length > 0, "Hash is required");
       require(documents[_hash].timestamp > 0, "Document not found");
       require(
           msg.sender == address(0xbc56bb97DCAe27474b4bBDc2186D671BBbEC0d32),
           "Unauthorized"
      );
       documents[_hash].valid = _valid;
```

This code implements a smart contract for managing documents on the blockchain. It provides functionalities to securely add new documents with unique identifiers and signatures, retrieve existing documents, and verify their existence. Additionally, it allows authorized users to control the validity status of documents.

Adding Documents:

- addDocument function allows adding a new document.
- It takes two arguments:
  - _hash: A unique identifier (bytes32) for the document.
  - _signature: A signature (bytes) presumably used to verify document authenticity.
- It performs several checks before adding:
  - Ensures both hash and signature are provided.
  - Checks if a document with the same hash already exists (to prevent duplicates).
  - Calls a validate function (not shown in this code) to verify the signature is valid for the provided hash.
- If all checks pass, it creates a Document struct containing the hash, signature, validity status (initially true), and a timestamp of when it was added.
- This Document struct is stored in a mapping called documents using the hash as the key.

Retrieving Documents:

- getDocument function allows retrieving a document by its hash.
- It checks if the provided hash has a value in the document mapping (indicating the document exists).
- If the document exists, it returns a copy of the entire Document struct associated with the hash.

Checking Document Existence:

- documentExists function allows checking if a document with a specific hash exists in the contract.
- It simply checks if the document's timestamp in the document mapping is greater than zero (indicating a document exists for that hash).
- It returns a boolean value (true if document exists, false otherwise).

Changing Document Status:

- changeDocumentStatus allows changing the validity status of a document.
- It takes two arguments:
  - _hash: The unique identifier of the document.
  - _valid: A boolean value representing the new validity status (true or false).
- It performs several checks before updating:
  - Ensures the hash is provided.
  - Checks if the document exists.
  - Restricts access to this function by requiring the sender's address to match a specific authorized address (presumably an administrator).
- If all checks pass, it updates the valid field of the Document struct associated with the hash to the provided _valid value.

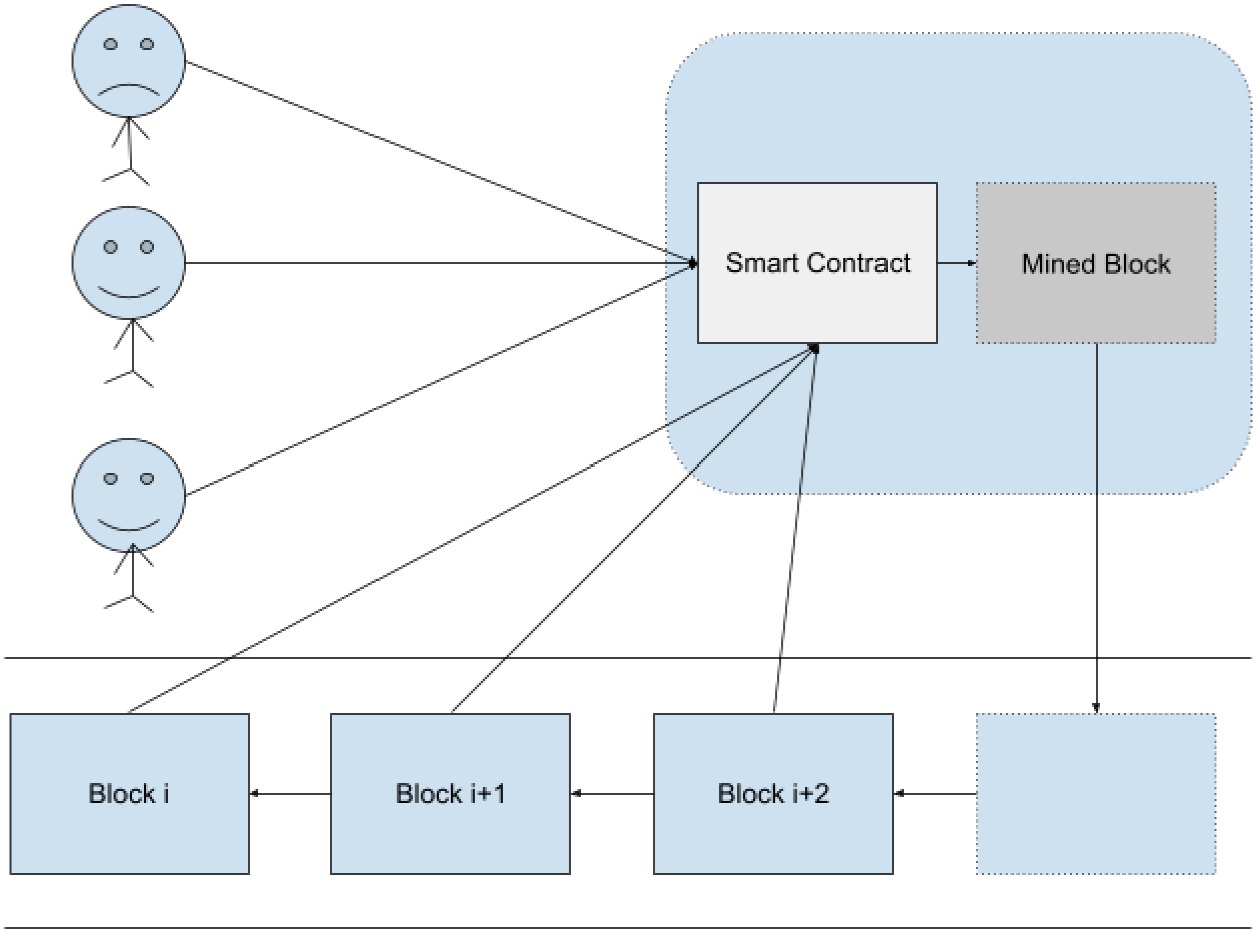


FIG 2.2 User and blockchain interaction

The data from the smart contract is made into a block which is mined by miners. The smart contract itself sits at the top, while below it a "Mined Block" represents a validated block added to the blockchain ledger

Implementation

- This may be done with any software. The first step is to create a soft copy of the certificate. Reading a CSV file with a list of students and the courses they are receiving certificates for is how we are going about this. A template certificate is also made with blank spaces that will later be filled with the names of the student and their course they are completing. The program reads the csv file and uses the python image library to paste the names on the certificate.

```
      DECLARE a file object `file`
      OPEN file named 'students.csv' in read mode and assign it to `file`
      WHILE there are lines to read in `file`
            READ a line from `file`
            REMOVE leading/trailing whitespaces from the line and store
            it in `line`
            SPLIT `line` using comma (',') as delimiter and store parts
            in variables `name` and `cert`
            CALL `certificate_generator.generate_certificate` function
      with arguments:
            - `name` (student name)
            - 'utils/template.png' (path to certificate template)
            - f'gen_certs/{name}.png' (output path for generated
      certificate)
            - `cert` (certificate information)
      PRINT a message "Certificate generated for " followed by `name`
CLOSE the file `file`
```

- The university then connects to the contract which is stored in the blockchain using metamask or using their private key. It also verifies if the wallet exists or not.

- We then calculate the hash of the certificate and digitally sign it with the private key. If metamask is used, they will be redirected to the metamask page to confirm that they are going to sign the certificate

```
      DECLARE a variable `certificates` to store certificates data
      FOR each certificate filename `cert` in the directory 'gen_certs'
              DECLARE a variable `cert_path` and construct the full path
      by joining 'gen_certs' and `cert`
            OPEN the file at `cert_path` in read binary mode and assign
      it to `file`
            READ the content of the certificate file into a variable
      `cert_data` using `file.read()`
            CALL function `crt.sign_message` with `cert_data` as argument
      and store the result in variables `cert_hash` and `cert_sign`
            OPEN the file "certificates.csv" in append mode and assign it
      to `output_file`
            EXTRACT the student name (`name`) from `cert` by splitting at
      '.' and taking the first element
            WRITE a line to the `output_file` with the format
      "{name},{cert_hash},{cert_sign}\n"
            PRINT a message "Certificate signed for " followed by `name`
            CLOSE both files `file` and `output_file`
```

- Once the hash and the digital signature is saved, we read the hash from that file and make a QR code with it. This QR code is then pasted on the certificate using PIL(Python Image Library)

- The smart contract is then developed. This smart contract has functions for Adding documents, verifying if the document exists, and changing the validity of the document. All the functions have a requirement that verifies that the function is called only by the university. It does this by comparing the sender's address and the hardcoded value of the address which is stored as a variable in the contract.

- Once the contract is made, it is then compiled and uploaded or deployed to the blockchain.After getting uploaded successfully the contract address is returned.

- The ABI (Application Binary Interface) is used to communicate with the smart contract. The ABI contains the different functions of the smart contract along with their respective inputs and outputs.

- Next the hash of the document and the digital signature is uploaded to the blockchain by calling the add document function. The transaction is then signed with the private key of the university and sent.A receipt is sent back when the transaction is successful.

```
OPEN the file "certificates.csv" in read mode and assign it to `file`
FOR each line `line` in the file
    REMOVE leading/trailing whitespaces from the line and store it in `line`
    SPLIT `line` using comma (',') as delimiter and store parts in variables `name`, `cert_hash`, and `cert_sign`
     CALL function `crt.add_document` with arguments: - `cert_hash` (certificate hash) - `cert_sign` (certificate signature) - `contract_address` (address of the smart contract) - `abi` (application binary interface of the smart contract)
     PRINT a message "Document added successfully for " followed by `name`
CLOSE the file `file`
```

- Now anyone who wants to verify the certificate can call the document exists function from the smart contract which returns true if the document can be found or else return false.

Verifying a Hard Copy

- The basic thing to do to verify the certificate is to extract the hash one way or another. We propose two easy solutions. One is by the use of QR code and the other is by the use of NFC.

- The hash of the certificate can be encoded into a QR code and can be scanned when anyone wants to verify it.

- The hash can also be stored in a NFC tag. The NFC tag can also be made read-only so the information stored on it cannot be modified. Also NFC tags can store more information so additional information like the name of the student, the wallet address of the university can also be stored on it.

Verifying a Digital Copy

- The hash of the digital document can be calculated by using any open source tool or a trusted website. The returned hash can be pasted into the verifying software which returns if the certificate is valid or not.

Combining Hard Copy and Digital Copy

The digital version can be saved in a central location. When the QR code is scanned, the certificate can be downloaded and can be passed to hash functions that find the hash. That hash can be compared with the hash found on the QR code to ensure that the certificate that is downloaded is not tampered with in any way.

Conclusion

Blockchain is one of the most popular technologies in the 21st century. It has provided solutions for many modern day problems. In this paper we were able to show how blockchain can be used to verify documents and ensure their security. This model avoids counterfeiting and forgery and ensures recruiters are getting trusted information from the blockchain network. This process can also be implemented to secure other types of documents as well. Since the method to verify a certificate only involves the use of hash, and the code to verify the certificate is open sourced, it is very easy for developers to make apps for web,desktop or mobile.

References

Roshani S. Bele and Jayant P. Mehare, "A review on digital degree certificate using blockchain technology", IJCRT, vol. 9, no. 2, pp. 2320-2882, 2021.

Nishant Anand, "New principles for governing Aadhaar: Improving access and inclusion privacy security and identity management", Journal of Science Policy Governance, vol. 18, no. 01, pp. 1-14, 2021.

T. Aditya Sai Srinivas, Ramasubbareddy Somula and K. Govinda, "Privacy and security in Aadhaar" in Smart Intelligent Computing and Applications, Singapore:Springer, pp. 405-410, 2020.

Harshita Khandelwal et al., "Certificate verification system using blockchain" in Advances in Cybernetics Cognition and Machine Learning for Communication Technologies, Singapore:Springer, pp. 251-257, 2020.

Omar S. Saleh, Osman Ghazali and Muhammad Ehsan Rana, "Blockchain based framework for educational certificates verification", Journal of critical reviews, vol. 7, no. 03, pp. 79-84, 2020.

K. Kumutha and S. Jayalakshmi, "Blockchain Technology and Academic Certificate Authenticity—A Review", Expert Clouds and Applications, pp. 321-334, 2022.

A. Gayathiri, J. Jayachitra and S. Matilda, "Certificate validation using blockchain", 2020 7th International Conference on Smart Structures and Systems (ICSSS), 2020.

Untung Rahardja et al., "Immutable Ubiquitous Digital Certificate Authentication Using Blockchain Protocol", Journal of applied research and technology, vol. 19, no. 4, pp. 308-321, 2021.

Binh Minh Nguyen, Thanh-Chung Dao and Ba-Lam Do, "Towards a blockchain-based certificate authentication system in Vietnam", PeerJ Computer Science, vol. 6, pp. e266, 2020.

Neethu Gopal and Vani V. Prakash, "Survey on blockchain based digital certificate system", International Research Journal of Engineering and Technology (IRJET), vol. 5, no. 11, 2018.

Ninoslav Marina and Pavel Taskov, "Blockchain-based application for certification management", Tehnički glasnik, vol. 14, no. 4, pp. 488-492, 2020.

S. R. Reeja and N. P. Kavya, "Real time video denoising", 2012 IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA), pp. 1-5, 2012.

J. Alanya-Beltran, S. R. Reeja, T. S. Rajeswari, C. Valderrama-Zapata, S. Akram and D. Kapila, "Smart Loading System of Bi-Directional Wireless Network", 2022 International Conference on Innovative Computing Intelligent Communication and Smart Electrical Systems (ICSES), pp. 1-7, 2022.

S. Terumalasetti and D. R. S. R, "A Comprehensive Study on Review of AI Techniques to Provide Security in the Digital World", 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), pp. 407-416, 2022.

https://peerj.com/articles/cs-266.pdf

Rama Reddy T, Prasad Reddy P, Srinivas R, Raghavendran C V, Lalitha RVS, Annapurna B. Proposing a reliable method of securing and verifying the credentials of graduates through blockchain. EURASIP J Inf Secur 2021;2021:1–9.

Dumpeti NK, Kavuri R. A framework to manage smart educational certificates and thwart forgery on a permissioned blockchain. Mater Today Proc 2021.

Shahnaz A, Qamar U, Khalid A. Using blockchain for electronic health records. IEEE Access 2019;7:147782–95.

Bahrami M, Movahedian A, Deldari A. A Comprehensive Blockchainbased solution For Academic Certificates Management Using Smart
Contracts. 2020 10th Int. Conf. Comput. Knowl. Eng., IEEE; 2020, p. 573–8.

Wang Z, Lin J, Cai Q, Wang Q, Zha D, Jing J. Blockchain-based certificate transparency and revocation transparency. IEEE Trans Dependable Secur Comput 2020.

Cheng J-C, Lee N-Y, Chi C, Chen Y-H. Blockchain and smart contract for digital certificate. 2018 IEEE Int. Conf. Appl. Syst. Invent., IEEE; 2018, p. 1046–51.

Chavan DP, Raut ST, Waghmode SU. QR Code Based Digitized Marksheet System. Int J Adv Res Comput Sci 2014;5.

Madala DS V, Jhanwar MP, Chattopadhyay A. Certificate transparency using blockchain. 2018 IEEE Int. Conf. Data Min. Work., IEEE; 2018, p. 71–80.

Hasan M, Rahman A, Islam MJ. Distb-cvs: a distributed secure blockchain based online certificate verification system from bangladesh perspective. 2020 2nd Int. Conf. Adv. Inf. Commun. Technol., IEEE; 2020, p. 460–5.
blockcert, 2020 n.d. https://www.blockcerts.org/.

Castor A. Cardano blockchain's first use case: Proof of university diplomas in greece. Bitcoin Mag 2018.

Turkanović M, Hölbl M, Košič K, Heričko M, Kamišalić A. EduCTX: A blockchain-based higher education credit platform. IEEE Access 2018;6:5112–27.

BCDiploma, 2018 n.d. https://www.bcdiploma.com/en.Daraghmi E-Y, Daraghmi Y-A, Yuan S-M. UniChain: a design of

blockchain-based system for electronic academic
https://blog.dock.io/instant-verification-announcement/

https://ieeexplore.ieee.org/document/10128289