

PROVIDING AUTHENTICATION AND PRIVACY FOR UNIVERSITY CERTIFICATES USING SMART CONTRACTS IN BLOCKCHAIN TECHNOLOGY

Gururaj Harinahalli Lokesh^{1,*}, Uvais Mon Valyagath Vadake Nalagath¹,
Vignesh Vijay Kuamr¹, Francesco Flammini²,
Rakesh Kirugaval Ramaswamyaradhya¹ and Harshitha Kamal Kannan¹

¹Vidyavardhaka College of Engineering
Mysuru, India

²University of Applied Sciences and Arts of Southern Switzerland IDSIA USI-SUPSI
Manno, Switzerland

DOI: 10.7906/indecs.20.4.7
Regular article

Received: 17 March 2022.
Accepted: 7 April 2022.

ABSTRACT

Traditional ways of distributing and verifying academic certificates are not efficient. Certificates are distributed as hard copy. Verifying the integrity of the certificate is a time and resource consuming process. As a result, forged certificates have become common. It is very difficult to differentiate between a real and a forged certificate. Through our system, we intend to make the certificate generation, distribution, and verification process seamless. Any student can enter his or her personal details, academic coursework details, and the university code, and thus submit a certificate request to the university. University admins can verify the certificate requests, and approve or reject the requests as per their policy. Any student or third party could verify the integrity of the certificate by entering the details of the certificate under scrutiny into the system. Data required to verify the integrity of the certificate will be stored on a blockchain. All verification data will be stored on the blockchain, so it will be tamper proof. This eliminates or limits the case of forged certificates to a large extend.

KEY WORDS

blockchain, smart contract, decentralised app

CLASSIFICATION

JEL: O33

*Corresponding author, *✉*: gururaj1711@vvce.ac.in;
Vidyavardhaka College of Engineering, P.B. No.206, Gokulam III Stage, Mysuru – 570 002,
Karnataka, India

INTRODUCTION

Digital technology plays a vital role across the globe. Nowadays the majority of things are converting from offline to online. Privacy and authentication are also an important factor. Transition from offline to online makes any process highly efficient. It is environmentally friendly. Digital assets are long lasting, portable and easy to store. Digital assets are also secure and private.

Since the emergence of blockchain, digital assets could now be decentralised. This means that users can now store their data online, forever, without worrying about any one failure point taking down the entire data and services. Blockchain has changed the way the internet functions. User data could now be distributed across multiple participating nodes, yet completely secure and private. This ensures that digital assets are persistent.

This transition from offline to online has opened up many opportunities to make processes more secure and efficient. One among them is digital academic certificates. Today most of the universities distribute academic certificates as a hard copy. They collect the data, generate the certificate, print it and manually distribute it to all the students. Once distributed, it is hard to verify the integrity of the certificate. It is due to this reason that cases of forged certificates have risen. Employers are finding it time consuming and resource intensive to verify the validity of a candidate.

This article focuses on a blockchain based approach to solve the above-mentioned issues. The system generates the certificate and stores the data required to verify the integrity of the certificate on blockchain. This provides zero knowledge proof. So no private data is made public, yet the validity of the certificate is public. This provides an effortless mechanism for the employers to verify the integrity of any certificates generated through the system.

BLOCKCHAIN

Blockchain has emerged as a trending technology in recent years. It entered the market as an infrastructure for crypto currency. Today, with the help of smart contracts, blockchain can be used for storing and distributing data throughout the blockchain network. The data written to the blockchain are immutable. New blocks are created for any update, the previous data logs will be still available. This ensures that data on blockchain cannot be tampered. Since blockchain has a distributed network, it avoids a single point of failure. Today, every industry is trying to transition their business process to blockchain. Increasing number of users of crypto currencies is an indicative factor of the growth of blockchain technology, Figure 1.

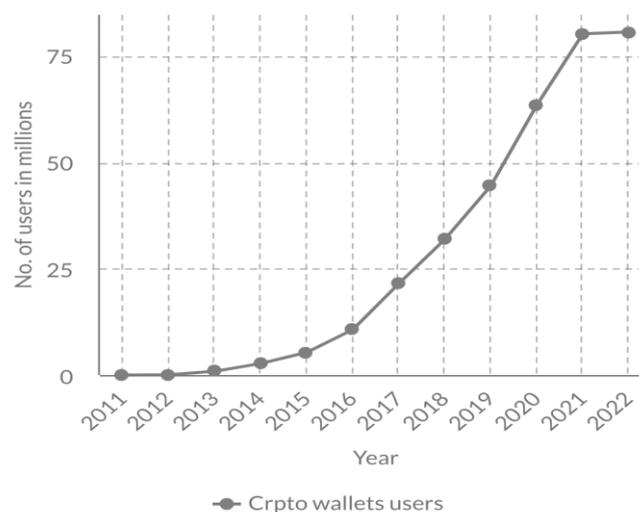


Figure 1. Number of crypto wallet users [1].

UNIVERSITY CERTIFICATES

Academic certificates are crucial for a graduate. These certificates convey their academic achievements and abilities. Employers depend on these certificates as one of the evaluating factors of a graduate. Currently, universities provide certificates as a hard copy. These are prone to damage over years or could be lost. Reapplying for a certificate is a cumbersome process. It is also hard to verify the integrity of the certificate. The cases of tampered or fake certificates still exist. The advancement in blockchain technology could be used to resolve this issue. Blockchain could give zero knowledge proof for the integrity of a certificate.

ETHEREUM

Ethereum is an open source blockchain technology. It supports smart contracts. Transactions on Ethereum are persistent and immutable. Ether is the native cryptocurrency of the Ethereum network. Users will have to pay a small gas fee for any transaction on Ethereum. The fee also depends on how much data has to be written to the contract. It is therefore ideal to store minimum data on the blockchain and achieve the desired outcome. The proposed system will only store the SHA256 hash of the certificate file.

Ethereum uses Proof of Work as a consensus mechanism. Proof of work requires a node to solve a complicated mathematical puzzle. Finding the solution of the puzzle should be hard, but validating the given solution should be easy. The puzzle should not be too hard that it makes it practically inefficient, nor too easy to allow DOS attack. Proof of Work has the following mathematical model.

An irreversible function f is defined for the entire network. For any transaction t , y is defined. A node is expected to find x such that,

$$f(x) = y. \quad (1)$$

where $x \rightarrow$ *Nonce* and $y \rightarrow$ *Required hash prefix*.

Now, consider 3 nodes that provide the answers as x_1 , x_2 and x_3 respectively. Any participating node can check which answer is correct by applying the function on the answers as follows:

$$f(x_1) = y_1, f(x_2) = y_2, f(x_3) = y_3.$$

Then they compare y_1 , y_2 and y_3 with y . The corresponding value of x for which $f(x)$ matches y is selected as the answer and the node who generated x is rewarded. The approval of the solution has to be done by 51 % of the participating nodes. Finding the value of $f(x)$ is a rather effortless process. So evaluation of the provided answer is easy.

ETHEREUM VIRTUAL MACHINE

Ethereum Virtual Machine (EVM) is a software platform based on blockchain. EVM helps developers to deploy smart contract bytecode. It is a Turing complete machine. Every full node of Ethereum blockchain will have an EVM implementation that runs the smart contract. EVM is more like a state machine. It will have a current state, take input and transition to another state:

$$F(S, T) \rightarrow S'. \quad (2)$$

where S is an old state, T is a valid transaction, F is an Ethereum state transition function, and S' is the new state.

SMART CONTRACT

Smart contract is a revolutionary technology that allows developers to deploy programs to blockchain to be run on EVM. Smart contracts were introduced in blockchain 2.0. This led to the emergence of distributed apps. The proposed system will use smart contracts to read and write data to the blockchain and to provide data privacy and security.

SOLIDITY

Solidity is the programming language used to write smart contracts for Ethereum. The smart contract for the system logic will be written in Solidity. A solidity program will consist of compiler version declaration followed by imports and contract definitions. A contract will consist of state variables, modifiers and functions. State variables track the current state of the contract. The function helps the contract to modify the state variable, access the state variable data or perform other transactions. State variables are changed by transactions. Once the smart contract is written, it is compiled using Solidity compiler. The bytecode generated is deployed on the EVM. All interaction of the system with the blockchain will be through this smart contract.

The main motivation to build this system is the rising cases of forged certificates. Academic certificates are one of the deciding factors for employment and higher studies. According to the current system, it is easy to forge a certificate or edit an existing certificate. It is practically impossible to check the integrity of all the certificates by an employer. So the employer will have to depend on the moral ethics of the candidate to trust the integrity of the certificate presented. Using forged certificates, candidates manage to secure undeserving benefits. The proposed system provides a trustworthy portal to the employers for dependable student certificates. The system also makes the certificate generation and distribution process more efficient and error free. It automates the process of distributing the certificate to the students. This saves a lot of time and effort for the university. This article focuses on building a decentralised app that could generate student certificates. The generated certificate is stored and distributed to the students. Our system also provides a portal to verify the integrity of the certificate by comparing the hash stored on the blockchain against the hash of the certificate under test.

The rest of the article is structured as follows: section two contains details of literature survey, detailed methodology is explained in section three, results and discussions are explained in section four, and at last conclusions are drawn in section five.

LITERATURE SURVEY

Significant research has been done so far towards certificate generation and maintenance with and without blockchain technology. Some of such important and recent works were reviewed and mentioned as follows.

Blockchain is an emerging technology. It is finding its way into the digital world quickly. Today, all the sectors are trying to implement blockchain in their business process. One of the main reasons blockchain is emerging as a candidate solution to all real world problems is trust. Blockchain is highly trustable. Authors in [2] talk about how blockchain's immutable nature helps people develop trust in decentralised apps. This helps blockchain to be used in many crucial applications. In [2], the authors have developed a voting system using blockchain. Since this system is decentralised and distributed over blockchain, it is impossible to tamper the result. Hence, it makes the election process more trust-able, transparent and efficient.

Blockchain can be used in machine learning and IOT based applications as well [3]. This manages the trust element of the system. Another application of blockchain technology in IoT devices is studied in [4]. Here, the authors use blockchain for tamper proof authentication and authorization. The botnet detection system mentioned in [5] proves that blockchain solution could be integrated with a smart city network to derive the best possible system. Article [6] shows how blockchain could be used as a supporting service for IoT and Deep Learning applications to detect BotNet attacks.

Smart contract is the programmable part of the blockchain [7]. Smart contract lets developers implement custom logic, store it on the blockchain, and run it on Ethereum Virtual Machine. The authors in [7] talk about how smart contracts could be used to create a greater extent of trust between transacting parties. The emergence of smart contracts paved the way for contract automation, they are executed when certain conditions are met. Smart contracts ensure both the parties can depend on the successful completion of the contract transaction as per the condition in the contract.

All these features of blockchain could be used to successfully develop a system that could generate and distribute student certificates that are tamper proof. [8] shows how the potential of blockchain and smart contracts could be exploited for this application. Here [8], the authors have discussed the features of blockchain that helps universities to generate tamper proof and transparent student certificates.

In [9], the author has developed a decentralised app based on hyperledger. They have used chaincode to interact with the blockchain. They generate the certificate, encrypt it, and store it on IPFS. The hash of the certificate and the link to the certificate file stored on IPFS is stored on the transaction written to the hyperledger. Storing files on IPFS is one of the efficient ways to avoid single point of failure [10]. IPFS is a protocol used to distribute multimedia files across the nodes in the network. The main objective of [10] is to let the students request for the certificate, the University to approve the certificate and the employer to verify the certificate. Article [11] focuses mainly on automating the certificate generation process. It also uses a hyperledger system to store transaction details about certificate generation. It provides a way to transfer the ownership of the certificate to another institute, in case of certificate transfer.

Another concern of certificate forgery is regarding certificates provided for online courses [12]. Online course completion certificates could be forged easily. The author of [12] has proposed a way to address this issue using blockchain. They have also included a module to transfer the credits so obtained from online courses into other academic courses. All these certificate generation, validation and credit transfer are counterfeit proof. Meaning, nobody could generate a fake certificate and transfer non earned credits to other courses.

One of the economical yet tamper proof ways to generate certificates is to use a centralised database for certificate storage and store only the hash of the certificate on the blockchain [13]. This way, it is easy to find the hash of a particular certificate from the blockchain and compare it with the certificate under test to check its integrity. Using a centralised database might introduce a single point of failure, but it could be resolved by using a cloud storage that stores different copies of your database on many servers. This approach is more economical as writing to blockchain is a costly transaction.

Belurgikar et al. propose a blockchain system for identity management [14]. Through this system, students' profiles are stored on blockchain. The access to these profile data is protected. Also since it is on blockchain, the identity management process is transparent. Table 1 summarises literature survey, while Table 2 provides additional data from utilised literature sources.

METHODOLOGY

The system mainly consists of 3 modules. One module for the students to request for the transcript, one for the admins to verify and approve certificate requests and one for any 3rd party to verify the integrity of the certificate. There are 2 logical modules – the smart contract module and the web app module that interacts with the smart contract. Logical modules differentiate and divide the logical part of the system. This approach helps to build the system in a modular approach, which will be highly flexible and easy to scale up.

Table 1. Summaries of related articles.

Ref.	Objective	Proposal	Drawbacks
[8]	Certificate generation and distribution system using blockchain for verification.	Used smart contracts to store certificate details on blockchain.	Centralised database.
[9]	Tamper proof certificate management.	Used hyperledger for blockchain needs, IPFS for certificate storage and elliptic curve encryption for certificate encryption.	Chances of disappearing certificates from IPFS if no node has pinned it.
[10]	Secure certificate validation system.	Used smart contracts, remix online IDE, and IPFS to generate and store certificates.	Client users will require an Ethereum account and meta mask extension to run the software.
[11]	A secure system to transfer academic certificates from one university to another.	Uses hyperledger to transfer and verify certificates.	The system cannot generate the certificate nor guarantee the availability of the certificate.
[13]	Multisignature blockchain based certificate generation and storage.	Used Bitcoin transactions to store the certificate hash in blockchain.	Centralised database, no smart contract or chaincode.
[15]	Blockchain based app to distribute and validate certificates.	Used Unicoi to store certificate hash on blockchain for verification.	Uses a private/custom Unicoi network, availability of the certificate is not guaranteed.
[16]	Automated system to generate and verify certificates.	GUI based certificate designing, centralised database to store and verify certificates	Completely depend on the issuing authority for certificate availability and verification. Certificate manipulation is possible.
[17]	Certificate digitization and verification using blockchain.	Convert the physical certificate to digital certificate, store its hash on blockchain.	No guarantee of the availability of the certificate.
[18]	Certificate verification using smart cards.	Generate a UID for certificates, encrypt it and store it on a smart card. Use this UID for verification.	Depend on the availability and integrity of the university for certificate verification.
[19]	Store certificate details on a QR code to save space.	Generate the QR code for the certificate and encrypt it.	Depend on a centralised service for availability and verification of the certificate.

Table 2. Analyzing parameters implemented in related articles.

Reference	Document hash on blockchain	Generate document	Document verification	Decentralised document storage	Encryption	Centralised document storage
[8]	✓	✓	✓	X	X	✓
[9]	✓	✓	✓	✓	✓	X
[10]	✓	✓	✓	✓	X	X
[11]	✓	X	✓	X	X	✓
[13]	✓	✓	✓	X	X	✓
[15]	✓	✓	✓	X	X	✓
[16]	X	✓	✓	X	X	✓
[17]	✓	✓	✓	X	X	✓
[18]	X	X	✓	X	✓	✓
[19]	X	✓	✓	X	✓	X

SMART CONTRACT MODULE

Smart contracts are the backbone of this system's security. It interacts with the blockchain to read and write data related to the system. Here, the smart contract stores the hash of the certificate as a mapping, mapped from certificate id to certificate hash. It also stores the details of admins and the owner of the contract. Only the owner is allowed to add new admins. Only the admins can approve certificate requests. Owner is also an admin.

Moreover, it has 4 functions, `add_hash`, `get_hash`, `add_admin` and `remove_admin`. `add_hash` takes in 2 arguments, the certificate ID and its hash. Only admins can call this function. The smart contract also prevents overriding existing certificate hash. `get_hash` takes in 1 argument, the `certificate_ID`. If the hash of the given certificate exists, it returns the hash, else it returns null. `get_hash` is a public function that can be called by anybody. `add_admin` and `remove_admin` take in the address of the admin to be added to the list of admins or removed from the list of admins, respectively. These functions can only be called by the owner of the contract. The one who deploys the contract becomes the owner of the contract.

add_hash logic:

1. Input `certificate_id`, `certificate_hash`, `sender_address`.
2. Check if the sender is an admin.
3. If the sender is not an admin, revert the transaction.
4. Else, check if the `certificate_id` already exists.
5. If it already exists, then revert the transaction.
6. Else, add the hash to the mapping as below.

certificate_hash(certificate_id) → certificate_hash

get_hash logic:

1. Input certificate_id.
2. Fetch certificate_id from certificate_hash mapping.
3. Return fetched certificate_hash if it exists.
4. Else return an empty string.

The identity of the person interacting with the contract is authenticated using the private key of the user's Ethereum account.

There will be a contract helper program running on the backend web app server. This program establishes communication between the web app and the deployed contract. All requests to the contract from the webapp happens through this helper program.

WEB APP

Web App manages the front end interaction, centralised database handling, and the overall user experience of the system. Web app consists of 3 modules. Through these modules, users can request for certificates, generate certificates and verify the certificate.

Request Certificate Module

This module collects data from the students with the help of a HTML form. The data includes their personal details and academic course works. Then it commits these data to the centralised database. It will mark the certificate as yet to be approved in the database. This field helps differentiate uncertified certificates from certified certificates.

The database is private and cannot be accessed without authentication. So the system maintains user privacy. This database is common to the entire system. Any module within the system can access this database. Certificate generation module uses the data from this database to generate the certificate.

Approve Certificate Module

This module could only be accessed by university admins. They can open this module, view the certificate requests made, and approve the certificate requests if it is a valid request. When the admin logs in, the system collects the list of yet to be approved certificate requests from the database and displays them in the home screen. Admins can choose any certificate for verification and approval. Figures 2-4 depict described relations.

Verify Certificate Module

The integrity of the certificate is verified through this module. This module takes in the certificate(c') under test and its certificate ID. It calculates the hash(d') of c' and compares it with the hash present on the blockchain(d). If both the hash matches, it shows that the certificate is valid. If the hash does not match, it shows that the certificate is invalid, Figures 5-6.

$SHA256(c') \rightarrow h'$
 $hexdigest(h') \rightarrow d'$
If $d' = d$, then
 "The certificate is valid."
else
 "The certificate is invalid."

where $c \rightarrow$ Actual certificates, $c' \rightarrow$ Certificate under test, $h \rightarrow$ hash of actual certificates, $h' \rightarrow$ hash of certificate under test, $d \rightarrow$ hexdigest of h , $d' \rightarrow$ hexdigest of h' .

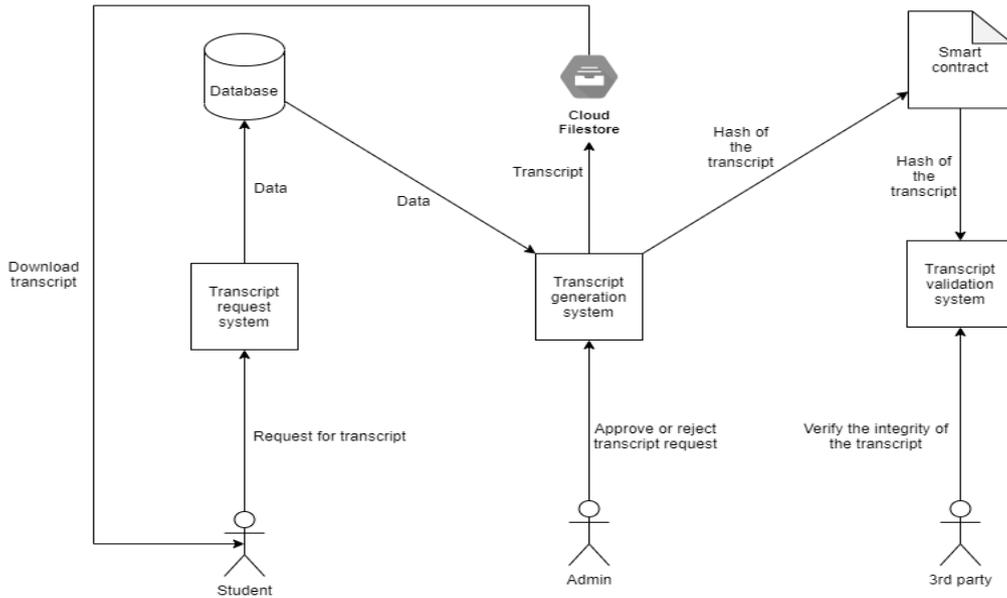


Figure 2. Interaction between different modules.

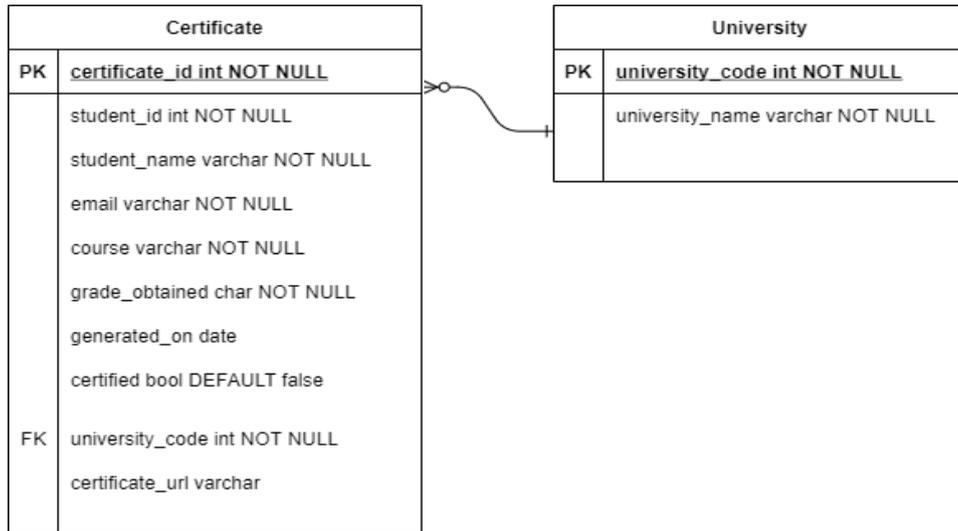


Figure 3. Database schema.

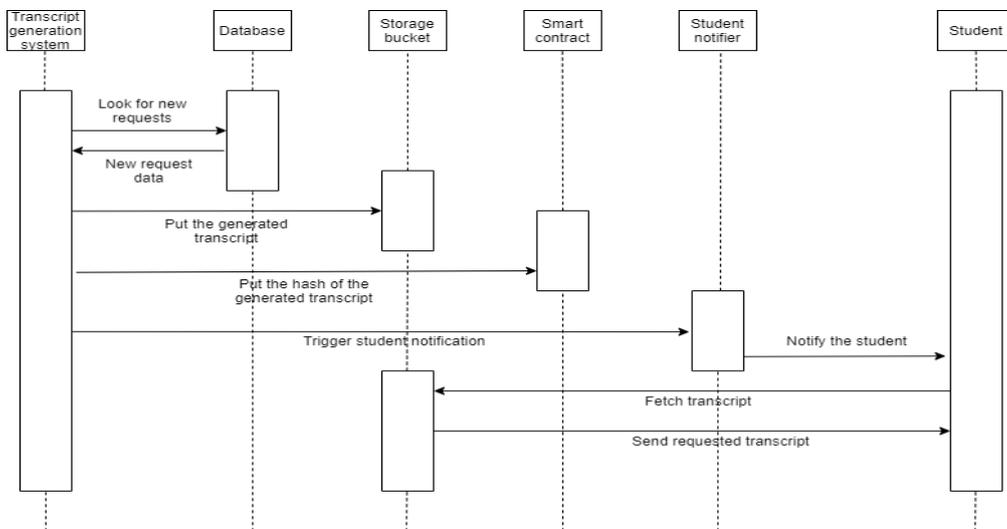


Figure 4. Certificate generation system sequence diagram.

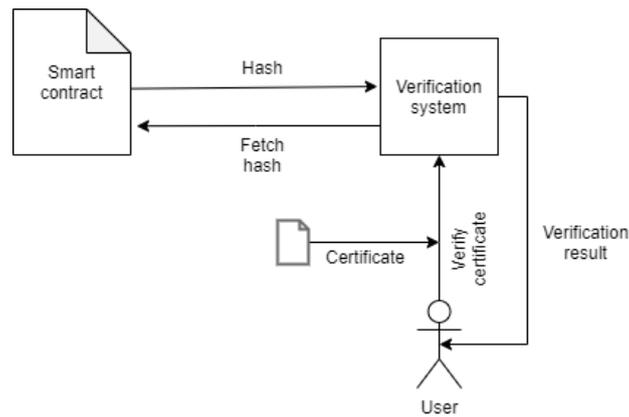


Figure 5. Hash Verification flow diagram.

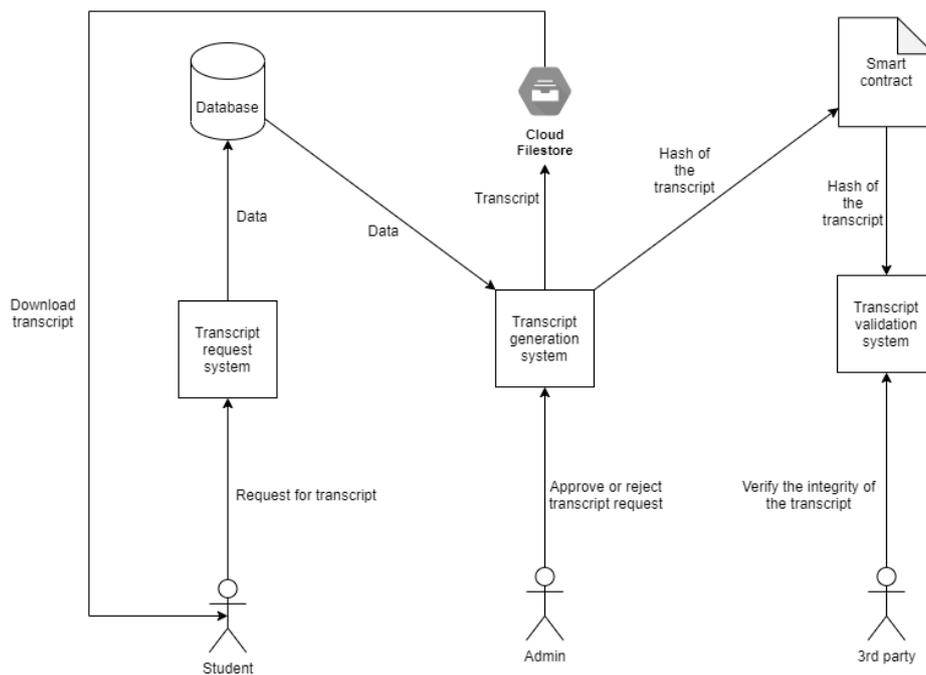


Figure 6. Interaction between different modules.

RESULTS

The end product of our system is a web app that can generate, distribute and verify certificates. The app can be deployed on any university network. Once deployed, students can start applying for certificates, admins can start approving certificate applications and employers can start verifying certificates. As students are filling out their certificate details, the chance of human error is less. Admins can log in, verify the details and approve the certificate. This procedure takes less time compared to manual processes.

The landing page of the app is shown in Figure 7. From there users can navigate to different pages as per their role.

A student can navigate to the request certificate page. Certificate request page consists of a form. Student has to fill out this form to provide his personal and academic details. Once he/she has filled the form, he/she can request for the certificate and wait for the approval by the university admins. The request certificate form is shown in Figure 8.

When the admins approve the certificate request, the student gets an email notification. The email will also contain the link to download the generated certificate from the storage bucket.

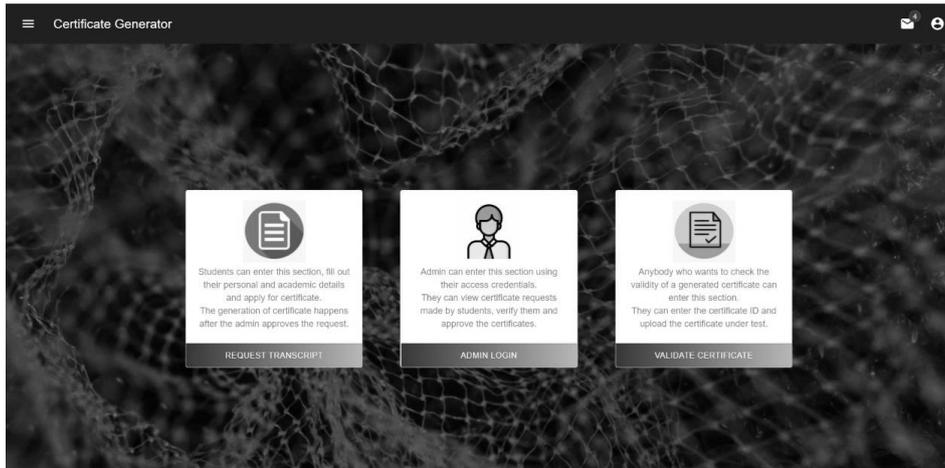


Figure 7. Home screen.

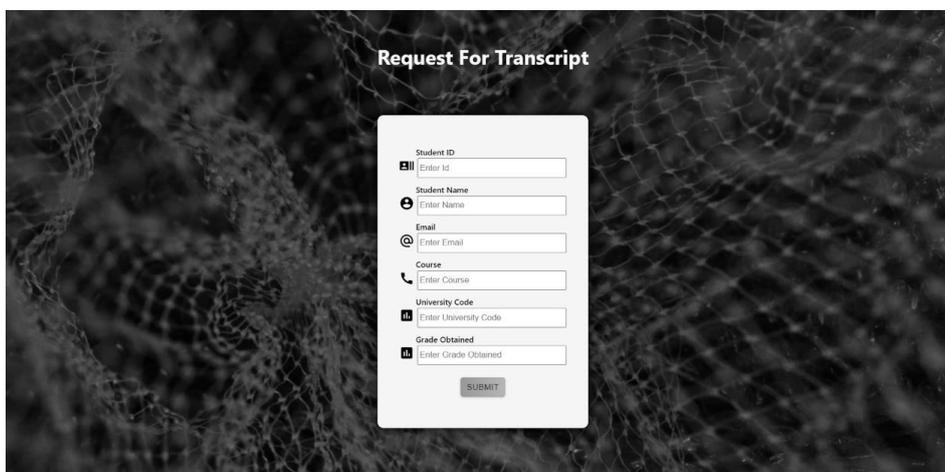


Figure 8. Request certificate page.

An admin can navigate to the admin login page from the landing page. Admins will have to enter their login credentials to access the admin page. Once the admins have logged in, they can view the certificate requests made and approve the requests accordingly. Admin page is shown in Figure 9.

When the certificate is approved, the generated certificate is stored in the Firebase storage bucket. The storage bucket is shown in Figure 10.

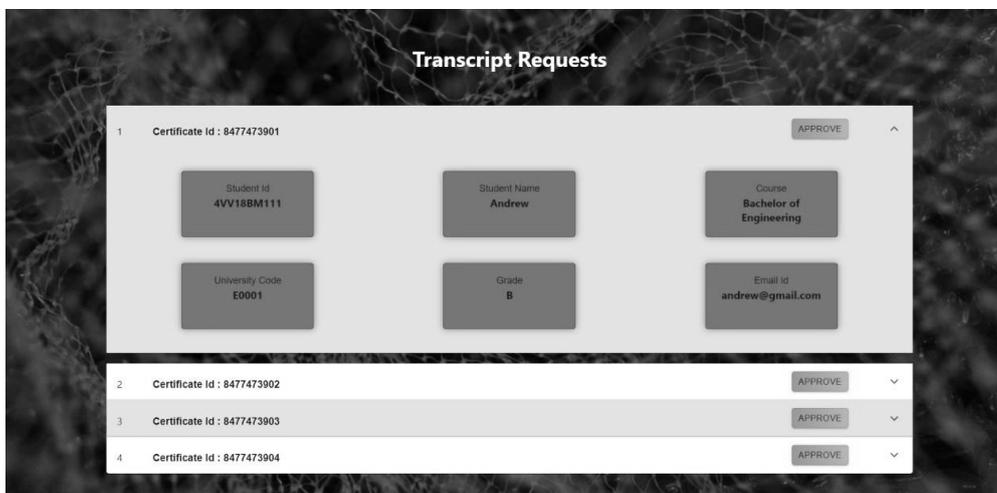
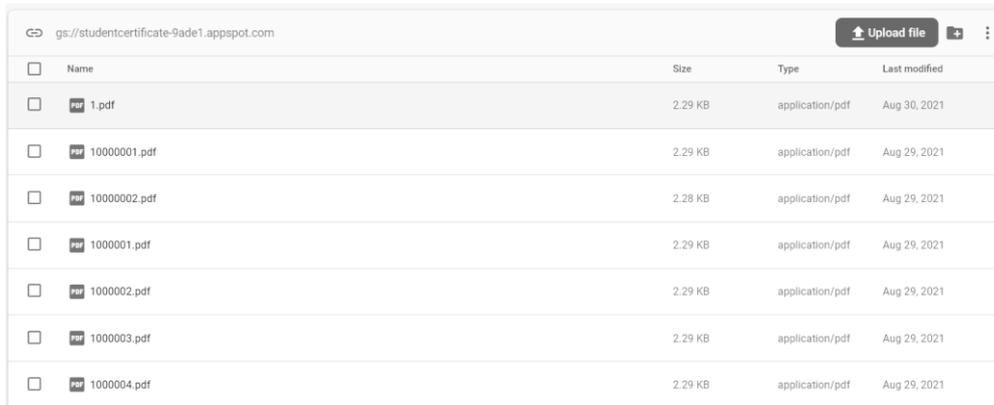


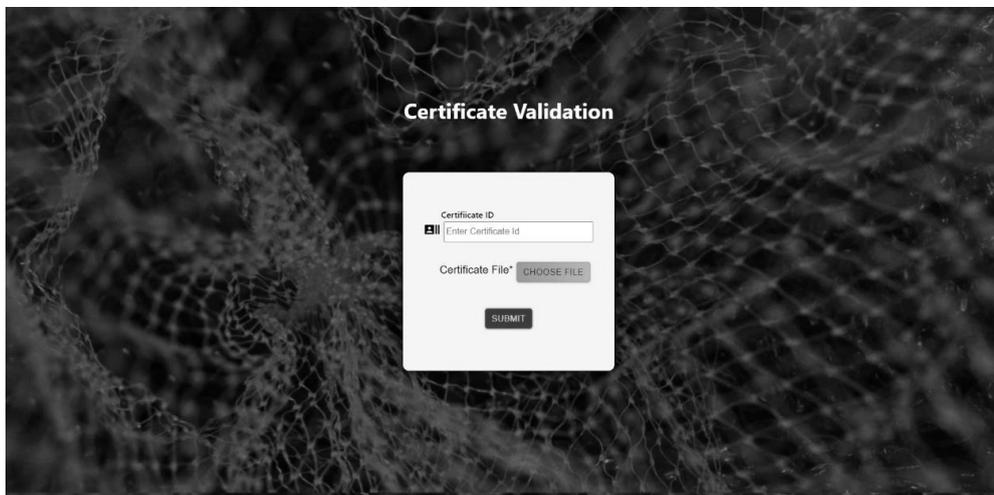
Figure 9. Certificate approval page.



Name	Size	Type	Last modified
1.pdf	2.29 KB	application/pdf	Aug 30, 2021
10000001.pdf	2.29 KB	application/pdf	Aug 29, 2021
10000002.pdf	2.28 KB	application/pdf	Aug 29, 2021
10000001.pdf	2.29 KB	application/pdf	Aug 29, 2021
10000002.pdf	2.29 KB	application/pdf	Aug 29, 2021
10000003.pdf	2.29 KB	application/pdf	Aug 29, 2021
10000004.pdf	2.29 KB	application/pdf	Aug 29, 2021

Figure 10. Firebase storage.

Anybody who wants to verify the validity of the certificate can go to the validation page from the landing page. Validation page consists of a form that takes in certificate ID and PDF certificate under test and informs the user if the certificate is valid or not. Validation page is shown in Figure 11.



Certificate Validation

Certificate ID
Enter Certificate Id

Certificate File* CHOOSE FILE

SUBMIT

Figure 11. Validation page.

The hash of the certificate is stored on the blockchain. The transaction details of storing a hash on the blockchain is shown in Figure 12.

Blockchain provides a decentralised database to store data. As blockchain is decentralised and involves a lot of individual parties who are spending their resources to mine the transaction, they will be charging clients who perform the transaction to compensate for the resources they spent. This charge is called gas fee. This gas fee could get high quickly. The gas fee depends on the gas spent for a transaction and the price of the gas at the time of transaction. On Ethereum mainnet, the gas fee is paid in ETH. Developers are required to pay attention to the data they store on blockchain to reduce the gas fee and make their system more cost efficient. This is one of the main concerns of decentralised applications.

Data chunks of different sizes are stored on the blockchain to study to dependence of cost of transaction on data size. The result of the analysis is shown in Figure 13. Our analysis shows that as the size of the data stored on the blockchain increases, the gas fee also increases. So, it is deemed necessary to find out a cost efficient amount of data to be stored on the blockchain such that the certificates are easy to verify and the gas fee is feasible, yet impossible to tamper the certificate. SHA256 hash of the certificate is found to be an ideal metadata to check the integrity of the certificate. SHA256 makes it impossible to modify the certificate, because a simple modification in the certificate will create a hash mismatch with the hash of

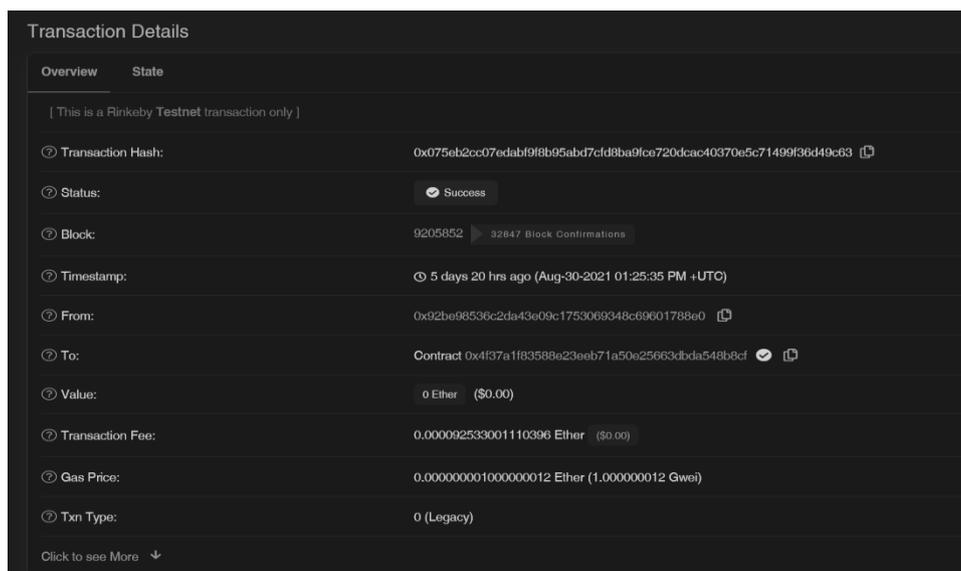


Figure 12. Transaction receipt.

the original certificate. By using SHA256 only 64 characters hexdigest of the hash is stored on the blockchain, which is cost efficient compared to storing the entire meta data of the certificate. It also speeds up the transaction process. This analysis is conducted on Ethereum Rinkeby test network as depicted in Figure 13.

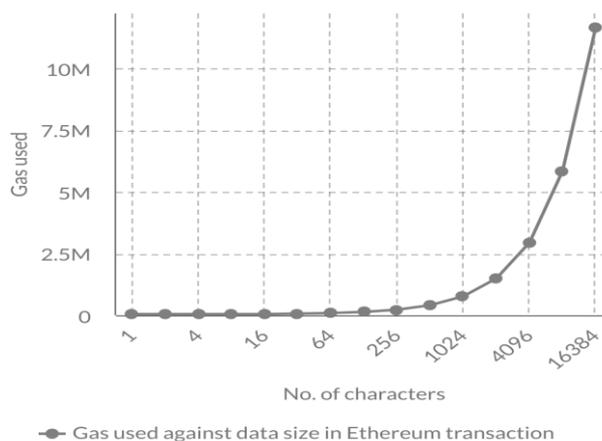


Figure 13. Transaction fee comparison.

CONCLUSION

The article mainly focuses on a system that can generate, distribute and validate academic certificates with the help of blockchain. The entire process is seamless. It saves a lot of effort and resources for the university. It also provides a tamper proof platform to verify the integrity of the certificate. The security of the system is maintained by Ethereum blockchain. In other words, to compromise our system, the attackers will have to compromise 51 % of the Ethereum nodes. It is practically infeasible. The end product is a certificate management system that is secure, tamper proof and highly dependable. Our system will make it nearly impossible to create forged certificates.

At present, our system has defined an admin role for the university. As future enhancement, more roles can be included. This enables the university to pass the certificates through different levels of approval. The certificate flows through the hierarchy of admins of the university before it is issued to the student. A decentralised storage can also be added to the

system. This ensures that the certificates are still available even if the university database goes down. Another improvement would be to automate and generate certificates in batches using the result database of the university. This enables generation of multiple certificates simultaneously without requiring individual approvals.

REFERENCES

- [1] Ştata: *Number of Bitcoin block explorer Blockchain.com wallet users worldwide from November 2011 to April 6, 2022*.
<https://www.statista.com/statistics/647374/worldwide-blockchain-wallet-users>, accessed 6th April, 2022,
- [2] Aroul Canessane, R., et al.: *Decentralised Applications Using Ethereum Blockchain*.
In: 5th International Conference on Science, Technology, Engineering and Mathematics. IEEE, pp.75-79, 2019,
<http://dx.doi.org/10.1109/ICONSTEM.2019.8918887>,
- [3] Vishwas, D.B., et al.: *Blockchain-Based Secure Method for Tiger Detection Using Machine Learning*.
In: Gururaj, H.L., et al., eds.: *Convergence of Internet of Things and Blockchain Technologies*. Springer, Cham, pp.221-242, 2021,
http://dx.doi.org/10.1007/978-3-030-76216-2_14,
- [4] Janardhana Swamy, G.B.; Janardhana, D.R.; Vijay, C.P. and Vinayakumar, R.: *Blockchain-Enabled IoT Integrated Autonomous Sewage Management System*.
In: Gururaj, H.L., et al., eds.: *Convergence of Internet of Things and Blockchain Technologies*. Springer, Cham, pp.41-56, 2021,
http://dx.doi.org/10.1007/978-3-030-76216-2_3,
- [5] Vinayakumar, R., et al.: *Visualized Botnet Detection System Based Deep Learning for the Internet of Things Networks of Smart Cities*.
IEEE Transactions on Industry Applications **56**(4), 4436-4456, 2020,
<http://dx.doi.org/10.1109/TIA.2020.2971952>,
- [6] Sriram, S.; Vinayakumar, R.; Alazab, M. and Soman, K.P.: *Network Flow Based IoT Botnet Attack Detection*.
In: IEEE Conference on Computer Communications Workshops. IEEE, pp.189-194, 2020,
<http://dx.doi.org/10.1109/INFOCOMWKSHP50562.2020.9162668>.
- [7] Wang, S., et al.: *An Overview of Smart Contract: Architecture, Application, and Future Trends*.
In: 2018 IEEE Intelligent Vehicles Symposium. IEEE, pp.108-113, 2018,
<http://dx.doi.org/10.1109/IVS.2018.8500488>,
- [8] Cheng, J.-C.; Lee, N.-Y.; Chi, C. and Chen, Y.-H.: *Blockchain and Smart Contract for Digital Certificate*.
In: IEEE International Conference on Applied System Innovation. IEEE, pp.1046-1051, 2018,
<http://dx.doi.org/10.1109/ICASI.2018.8394455>,
- [9] Raghav, et al.: *Tamper-Proof Certificate Management System*.
In: IEEE Conference on Information and communication technology. IEEE, pp.1-6, 2019,
<http://dx.doi.org/10.1109/CICT48419.2019.9066236>,
- [10] Padmavati, E., et al.: *Smart and Secure Certificate Validation System through Blockchain*.
In: Proceedings of 2nd International Conference on Inventive Research in Computing Applications. IEEE, pp.862-868, 2020,
<http://dx.doi.org/10.1109/ICIRCA48905.2020.9182975>,
- [11] Badr, A., et al.: *A Permissioned Blockchain-Based System for Verification of Academic Record*.
In: 10th IFIP International Conference on New Technologies, Mobility and Security. IEEE, pp.1-5, 2019,
<http://dx.doi.org/10.1109/NTMS.2019.8763831>,
- [12] Li, C., et al.: *A Blockchain System for E-Learning Assessment and Certification*.
In: 2019 IEEE International Conference on Smart Internet of Things. IEEE, pp.212-219, 2019,
<http://dx.doi.org/10.1109/SmartIoT.2019.00040>,

- [13] Li, R. and Wu, Y.: *Blockchain based Academic Authentication System Overview*.
Block Chain Laboratory, University of Birmingham, 2018,
<https://intranet.birmingham.ac.uk/it/innovation/documents/public/Experiments/Blockchain-based-Academic-Certificate-Authentication-System-Overview.pdf>,
- [14] Belurgikar, D.A.; Kanak Kshirsagar, J.; Dhananjaya, K.K. and Vineeth, N.: *Identity Solutions for Verification using Blockchain Technology*.
In: 2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE). IEEE, pp.121-126, 2019,
<http://dx.doi.org/10.1109/ICATIECE45860.2019.9063802>,
- [15] Huynh, T.T.; Huynh, T.T.; Pham, D.K. and Ngo, A.K.: *Issuing and Verifying Digital Certificates with Blockchain*.
In: 2018 International Conference on Advanced Technologies for Communications. IEEE, pp.332-336, 2018,
<http://dx.doi.org/10.1109/ATC.2018.8587428>,
- [16] Yusuf, A.D.; Boukar, M.M. and Shamiluulu, S.: *Automated Batch Certificate Generation and Verification System*.
In: 13th International Conference on Electronics, Computer and Computation (ICECCO). IEEE, pp.1-5, 2017,
<http://dx.doi.org/10.1109/ICECCO.2017.8333321>,
- [17] Gayathiri, A.; Jayachitra, J. and Matilda, S.: *Certificate validation using blockchain*.
In: 7th International Conference on Smart Structures and Systems (ICSSS). IEEE, pp.1-4, 2020,
<http://dx.doi.org/10.1109/ICSSS49621.2020.9201988>,
- [18] Lingampalli, J.R. and Namdeo, V.: *Unique Smart Card Verification System for Validating University Degree Certificates*.
In: 5th International Conference on Intelligent Computing and Control Systems. IEEE, pp.1574-1578, 2021,
<http://dx.doi.org/10.1109/ICICCS51141.2021.9432360>,
- [19] Somdip, D.: *New generation of digital academic-transcripts using encrypted QR code™: Use of encrypted QR code™ in mark-sheets (academic transcripts)*.
In: International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s). IEEE, pp.313-317, 2013,
<http://dx.doi.org/10.1109/iMac4s.2013.6526429>.