2023-03

Educational Certification on Ethereum blockchain: Analysis on How BCert and UZHBC differ

Aadeef, Tanvir Ahmed

Independent University, Bangladesh

Educational Certification on Ethereum blockchain: Analysis on How BCert and UZHBC differ

Tanvir Ahmed Aadeef
Dept. of Computer Science and Engineering
Independent University Bangladesh
Dhaka, Bangladesh
1910090@iub.edu.bd

Md Fahad Monir
Dept. of Computer Science and Engineering
Independent University Bangladesh
Dhaka, Bangladesh
fahad.monir@iub.edu.bd

Sanzar Adnan Alam
Dept. of Computer Science and Engineering
Independent University Bangladesh
Dhaka, Bangladesh
sanzar@iub.edu.bd

***Abstract*—Counterfeit certificates and diplomas are a common issue. Blockchain technology offers a potential solution through its decentralized ledger system, which uses cryptographic techniques and incentives to verify and store transactions in blocks that are added to the public blockchain. This secure and transparent method of storing data makes it ideal for Industry 4.0 and beyond as it is difficult for information to be stolen or for corrupt institutions to sell unauthenticated certificates. In this research paper, we compare and analyze existing two different methods of certification issuance and management which is BCert, and University of Zurich Blockchain. We also discuss the differences in algorithms for hashing and how we can improve the existing code for each method. Our aim is to thoroughly examine these approaches and contribute to the field of certification through our analysis and conclusions.**

Keywords—Blockchain, Ethereum, Cryptography, Ether, Hash, Smart Contract, Solidity

I. INTRODUCTION

In today's modern era, counterfeit methods of forging certifications are available all across the globe. It is widely seen that factories and mills specifically catering to the print and sale of diplomas and certificates for money is fast becoming a tier 1 industry, with a market cap theoretically in the billions. An emergence of counterfeits and replicas has led to special departments being employed to verify whether an educational or skills degree is authentic or not. This has created a large industry who's difficulty can be fixed simply using existing technology [1].

Enter Blockchain technology: It is a decentralized and distributed cryptographic ledger system where computers or nodes make up a network. The system incentivize work done of transactions to be verified and stored in a block, which is approved again and stored in the blockchain as a small chain of code.

The first blockchain to be created was Bitcoin and it uses scripting using a stack-based language called Script to encode transactions. Ethereum was developed by Vitalik Buterin and is a second-generation blockchain that was designed to address the limitations of Bitcoin's script language. It uses a Turing Complete programming language called Solidity, which allows users to execute any program, called a Smart contract, in addition to exchanging money. Smart contracts require a fee, called a gas fee [2], to be paid in order to be executed. The gas fee is measured in gwei, a denomination of ETH [3]. Ethereum is essentially a decentralized application (DApp) network that consists of accounts and uses Ether as a means of signed data transaction, with Smart Contracts being used to execute various functions [2].

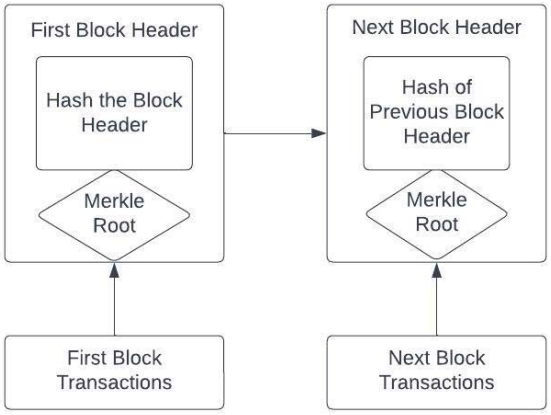

Fig. 1. Architecture of block chaining on blockchain networks

Blockchain allows for the secure, decentralized storage of data through the use of cryptographic techniques to link blocks together. When a block is verified and deemed valid by other nodes in the network, it is added to the public network and a copy is kept for reference. As per figure 1 it is detailed that when a block is created, the hash of the first block is time-stamped and recorded and passed on to a new chain in the network. This is so that the transactions of the first block can be authorized and validated and saved on the network forever. Since authorization is under the approval of groups of nodes in the network, there is no scope for conflict regarding transactions [4]. This decentralized, transparent approach to data storage makes it difficult for information to be stolen or for corrupt institutions to sell unauthenticated certificates [5]. Due to reverse cryptographic hash being an impossibility [6], it determines the efficiency and speed of issuing certificates, while the consensus algorithm used in blockchain technology helps to prevent network attacks and validate work [7]. Blockchain can therefore be used for the management and authentication of certificates as it is resistant to duplication and reproduction [8],[9].

The main objective of this paper is to focus on two different types of certification issuance and management methods. BCert is a decentralized certification system that utilizes Ethereum Smart contracts that act as a ledger for documentation. The other is University of Zurich Blockchain (UZHBC) which is used consistently for the issuance of diplomas. In essence this paper goes into a diverse and minute way where, divided into sections, defined are how verification of certificates work in a blockchain network, related works on the field of certification, how algorithms in hashing differ, and our analysis on two certification methods where we draw a conclusion on the performance of our

methods. We also add our take on how we improved the existing code of each verification method and how it is improved compared to the existing code; and this work is done to further benefit future research work on differing certification methods on the blockchain network.

II. Related works

Several works have been published on the topic of Blockchain certification using different methods, some of which we have mentioned to provide different perspective on how certification can differ.

EduCTX is a certification method that focuses on integration into a more unified homogenous system in which tokens or coins can be accredited but students must maintain a cryptographic credential in the system always, a system of grading founded on the European Credit Transfer and Accumulation System (ECTS) [10]. There is also BCDiploma which is also a Decentralized App (DApp) where the encrypted data is stored on a secure register and AES 256 is used to manage issues regarding access of data and confidentiality [11].

Institutions like MIT have their own certification method, called Blockcerts that create an open standard entirely using a DApp that enables students to share their credentials with employers and such using the app itself [12]. University of Nicosia (UNIC) is also implementing a way to record certifications of it's students. UNIC's method is used for issuance of certificates, diplomas, and fee payments. It uses SHA 256 hash algorithm for this purpose but there are no clear methods for any employer to verify the adequacy of the skills that are in the data chain [13].

In contrast some institutions also can pool their resources to create a centralized service for certification like "My eQuals" that allows students to access their certificates from a single location by paying a fee. It is a collaboration between several universities [14], but it's centralization can potentially create privacy issues and make it vulnerable to security breaches, as any leak or hack into the network could expose everyone's data. While a centralized approach may be efficient in the short term, it presents a danger for long-term storage of certificates and diplomas due to possible future exploits in the system. In contrast, this research aims to examine the key differences between two approaches to certification on a granular level, including their pseudocode, issuance of certificates, and suitability for different purposes. Comparison of how hashing differ in terms of encryption and decryption of two approaches is made and evaluate how well-defined each method is for its intended purpose.

III. Verification methods

Blockchain network works by verifying the work done by nodes using concepts defined under a consensus algorithm.

The first is proof of work which is a consensus algorithm that requires a large problem-solving calculation to be done in order for work to be verified and is therefore very computational heavy and so is expensive for data verification. It is how first-generation bitcoin was authenticated on the network, and most certification methods that employ proof of work do so with a small transfer of bitcoin from an issuer's address as the transfer fee. As such both in terms of electricity cost and computational cost, proof of work is undesirable for issuance of certificates for students around the globe [7].

Proof of stake meanwhile is an alternative algorithm that has recently overtaken proof of work and is simple: where the probability to generate a block is proportional to the owner's stake in the structured system with users with the highest stakes having the most incentive to maintain a secure network. Delegated Proof of Stake is a derivative in which delegates process in two ways:

- Building a block of transactions to occur
- Digitally signing said block to verify it for use [7]

IV. Methodology

Several steps have been conformed to evaluate the differences between BCert and UZHBC. This involves illustrating how certificates are issued for each use case, how the Smart contract code of each method differs and how we have improved on their existing code, the encryption algorithm employed for hashing purposes, the gas fees associated with issuance of certificates as well as the scope and general useability of certification methods in question.

It has also been noted the recent event in the Blockchain world, which is the Ethereum Merge. It is the merger of the Ethereum Mainnet (which was the main execution layer since conception) and Beacon Chain. Ethereum network did not strictly follow proof of stake from genesis, but after the merge the systems came together with accounts, balances, and Smart contracts under a unified method, where proof of work is permanently replaced by proof of stake [3]. It is important for differentiating one certification method from the other in terms of certificate issuance cost. Figure 2 disseminates how analysis of both methods are carried out:

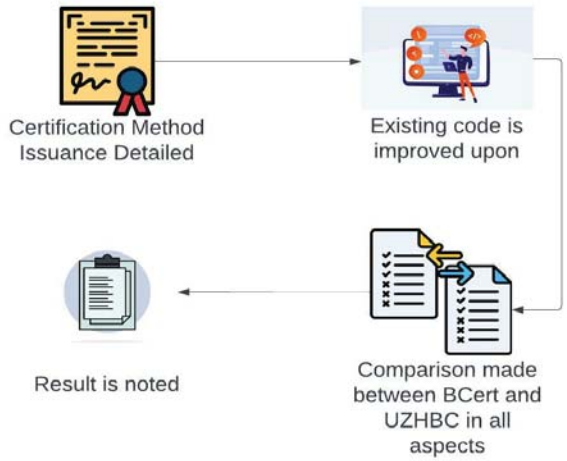

Fig. 2. Flowchart of Methodology for analysis of certification methods

Adhering to the results of the analysis made in this paper, an assertion is made on why leveraging blockchain technology is the future of certification in Industry 4.0 in this generation and beyond.

V. Analysis

In this section we dissect how each certification method differs, starting of with how they are issued to students and

elaborate the way in which they are used for accreditation; then analyze how Smart contracts work for each method work and improve on their approach with code of our own. Following up, an analysis on a more monetary nature is prepared for gas fee and estimate how much each certificate would cost to produce now, after which each certification methods' hashing algorithm and versatility is compared.

A. Issuance Process

The process of how students acquire certificates at the end of their career and in what way these credentials are distributed to employers are detailed int the following figures that elaborate in a concise way the flow of exactly how certificates go from the accreditation body to the employers. The first method for analysis is BCert and is as follows:

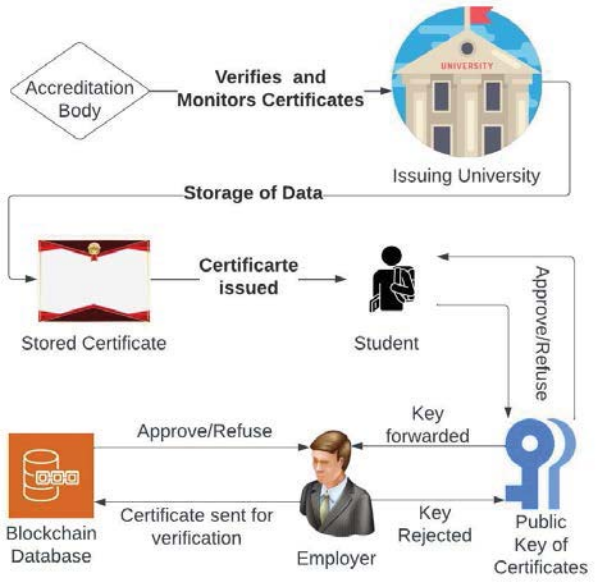

Fig. 3. Architecture of flow of issuance of certificate using BCert method

As seen from figure 3, BCert issues certificates from the universities in the form of handing a public key to the student in question who in turn hands over the public key to an employer who can verify whether or not the certificate is valid from the Ethereum Mainnet [15]. Next is the flow of certificate issuance of UZHBC and analysis on the process.

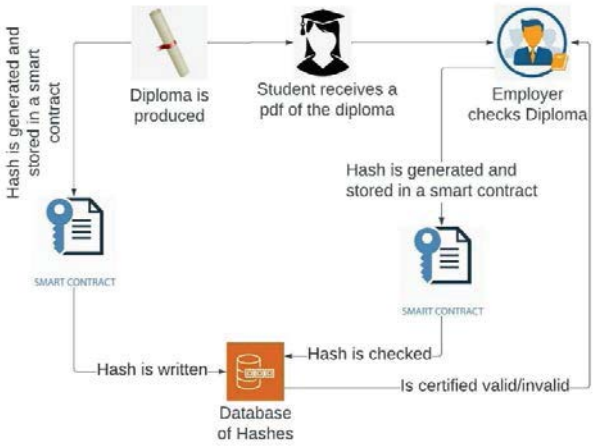

Fig. 4. Architecture of flow of issuance of certificate using UZHBC method

As we see in figure 4, there is a lot more privatized checking of certificates in UZHBC method where the hashes of issued certificates are stored on a database and is used for verification purposes. Due to the immutability of deploying Smart contracts on the Ethereum Network, this is a deletion-proof method of making sure data stays on the net and is not taken down in any instance. As only the hash is stored, this creates a much more server-free cost-effective scenario for storage of diplomas [16] as opposed to BCert which needs to store student ID among other stuff and focuses on verification of the accreditation body first rather than authenticity of certificate issued [15].

B. Smart Contract

Smart contracts are effectively an account controlled by code. Smart contracts can interact with other accounts or hold funds. Solidity programming language is utilized to write Smart contracts. The Solidity code that BCert and UZHBC writes in their Smart contract determines how certification will behave on deployment and how effectively it will allow implementation of user-defined operations that cannot be handled through cryptographic protocols.

To start off with BCert from [15] in the given code, every certificate can be accessed from certCount() function which works as the certificate serial number, and thereby it's index number.

Custom functions for different parameters include addCert() and getCertById(). To sign the transaction, addCert() gets 4 parameters of StudentID, address and state and the encrypted data for the function to occur. Various checks are made throughout the process to make sure the appropriate data in filled in. If the private key linked to the address is invalid, the transaction is not signed. If the encryption key is valid, the data is decrypted, so the function getCert() depends on the specified index of the certificates issued [15].

Our improved code for BCert is as follows:

```
PRAGMA SOLIDITY ^0.4.18;

CONTRACT bCERT {
  //......
  function getCert(string memory _SID) public view returns (
    string memory SID, string memory encData, uint256 created, State state
    ) {
  // Find the certificate with the given SID
  for (uint i = 0; i <= certCount; i++) {
    if (keccak256(bytes(certs[i].SID)) == keccak256(bytes(_SID))) {
      // If the certificate is found, return its information
      SID = certs[i].SID;
      encData = certs[i].encData;
      created = certs[i].created;
      state = certs[i].state;
      return;
    }
  }
  //......
}
```

Fig. 5. Improved Certificate Code of BCert

Instead of returning a tuple of variables, it would be more readable to use named return variables. For example, the getCert() function can be modified to return the variable as it is shown in the improved code in figure 5.

Moving onto UZHBC, from [16] in the given code, the UZHBC Smart contract used as a test purpose on Rinkeby network has two functions: issueCertificate() and verifyCertificate(). The former stores hashes in the Smart

contract where a single hash is passed as a string to keep costs of data storage low. The function issueCertificate() is only owner accessible, and the hashes stored by the university are on their end so cannot be accessed by others. Solidity's bytes32 data type is used to avoid storage space wastage and bytes and integer arrays are passed as parameters only. The latter code verifyCertificate() simply gauges the hash value that was initially issued to the received certificate request hash to verify if they are one and the same[16].

Our enhanced code for UZHBC is as follows:

```
1   PRAGMA SOLIDITY ^0.4.18;
2
3   CONTRACT UZHBC {
4      address public owner = msg.sender;
5      bytes32[] public diplomaHashes;
6      // updated "issueCertificate" function
7      function issueCertificate(bytes32[] memory byteHashes) public {
8         require(
9            msg.sender == owner,
10           "Only the contract owner can issue certificates."
11           );
12        diplomaHashes.push(byteHashes);
13     }
14     // updated "verifyCertificate" function
15     function verifyCertificate(bytes32 byteHash) public view returns (bool) {
16        return diplomaHashes.contains(byteHash);
17     }
18  }
```

Fig. 6. Enhanced Certificate Code of UZHBC

From line 1 of figure 6, the issueCertificate() function is much more optimized as the need for a for loop in the code is eliminated. The "push" function can accept an array as an argument and add all the elements of the array to the end of the "diplomaHashes" array in one call. This reduces the number of calls to the contract's storage and can improve the overall simplicity and efficiency of the contract. The "require" function is used to check the contract owner which throws an exception and reverts the transaction if the condition is not met rather than returning from the function, which eliminates the need for an if statement making the code much more concise.

Progressing onto line 9 of the same figure the verifyCertificate() function is optimized similarly as it uses built-in "contains" function of solidity's "bytes32[]" type which searches for a value in the array and returns a Boolean value if the value is found. This eliminates the need for a loop and counter variables and is marked as "view" as it only reads data and does not modify it, which makes it cheaper to execute.

C. Gas Fee

Gas fee determines the basic issuance cost of each certificate deployment, and each certification method has different ways to mitigate this cost and maximize efficiency in deployment. Gas fee is difficult to estimate due to data transfer involved upon which relies the fee of the transmission itself. For specific gas costs, repos are available that give a rough estimate as to how much issuance of diplomas/certificates can be depending on a variety of circumstances [17].

This rough guideline to determine cost from data coded to a certificate suggests that for BCert a transaction of 170 bytes of data costs around 725714 GAS [15] which is around $0.96 [18], with an initial deployment of a Smart contract costing around $20 [15].

UZHBC has a base gas fee of 21000 gwei for each execution of a transaction, where even no interaction with a Smart contract costs 21000 gas. The higher the gas the faster the transaction will be mined into the network. Around 6000 graduate from University of Zurich each year, and the nature of this method necessitates batch execution due to high gas fees, where a 1000 diplomas in a batch of 270 costs around $45 while for 6000 it costs around $273. As such, UZHBC deploys Smart contracts to issue diplomas annually and puts as much diploma hashes as possible into one transaction to mitigate costs [16].

Due to the recent Ethereum merge, prices of issuances of certificates are diminished as electricity costs are reduced by 99.5%, which makes issuance prices less of an issue now [3].

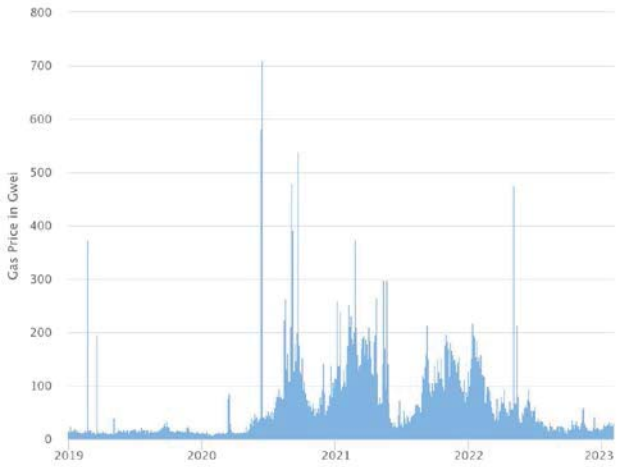

Fig. 7. Average Ethereum Gas price chart [22]

As seen in figure 7, Ethereum Gas price is very volatile and unpredictable so predicting how the network might charge can be difficult, which is also highly affected by the Ethereum Merge.

D. Certification Encryption Algorithm (Hashing)

The Algorithm utilized for each certification method achieves different goals and has their use cases, but the underlying encryption method of this algorithm is what distinguishes one from the other and is detailed in the following table.

TABLE I. COMPARISON OF CERTIFICATION ALGORITHMS

Parameters of Comparison	**SHA**	***AES***
Objective	Secure Verification of data	Secure transmission of data
Use case	For small and important index data for files and searching	Transmission of data for business, and other sensitive data
Nature	Unidirectional and irreversible	Bi-directional and reversible
Result	Initial message cannot be recovered	Original message can be retrieved using the decryption key
Security	More secure	Less secure
Approach	One-way hashing	Data Scrambling
Encryption	3 types	6 types

BCert relies on AES algorithm to provide confidentiality for transactions and offers real time online verification and revocation among others [19],[20]. UZHBC meanwhile uses a hash function called SHA-3 with a length of 256 bits which is collision resistant and software artifacts are mitigated with greater ease [16].

The information collected and presented from [21] in Table 1 shows that BCert employs AES as it is symmetric in nature, so it uses the same key for encryption and decryption and protects the data by scrambling. Leveraging AES as the core encryption algorithm has the upside in certification being that the data in them while immutable can be revoked incase of a breach in security in the physical medium of the certificate uploading institute. Any sort of egress can therefore be nullified and can also be traced back to the time it was changed on the network.

UZHBC meanwhile uses SHA as it is one-way function of a hashing algorithm, where diplomas are encrypted once and the data on them stays confidential always, as even a small change to the document will change the hash completely. It is also used primarily for indexing small portions of important data on the network therefore decreasing the overall logical footprint and cost of issuance of certificates. Therefore, this algorithm is particularly suited for usage in the blockchain network as diplomas need not much data to be uploaded and can be authenticated effortlessly.

E. Versatility

A comparison has been made between each certification method upon the following criteria:

1. Support any kind of certificate.
2. Accredited institutes.
3. Certificate cancellation.
4. Privacy of personal data.
5. No cryptocurrency.
6. Verifiable certificate information is stored solely on the Blockchain.
7. Student's certificates are collected in single digital wallet.

TABLE II. COMPARISON OF CERTIFICATION ALGORITHMS

Scope	**1**	**2**	**3**	**4**	**5**	**6**	**7**
BCert	✓	×	✓	×	×	✓	×
UZHBC	×	×	×	✓	×	×	×

Exists ✓; Does not exist ×

From all the information gathered and analysis on them [15], BCert supports any kind of certification as long as the user can code in solidity and get indexes for different parameters. As AES is used, certificates can be revoked as they are stored on the blockchain network, though is less secure than in the case of UZHBC. And lastly since BCert itself is not sanctioned by any educational body and is therefore public hence, information is stored solely on the Blockchain database publicly and can be referenced to by any student and issued by any university.

UZHBC however being under an institution has more secure privacy of personal information in terms of both the storage and issuance process of certificates . It tailors to their students' needs so can only issues diplomas in batches, with a combined storage space taking up around 192 kilobytes on the network [16]. Due to the nature of how the Smart contract is coded, only diplomas can be issued which is an issue for many educational institutions that may want to distribute certificates for students earning their bachelors and above.

F. Industry 4.0 and Beyond

Industry 4.0 is the automation and data and information transfer across applications and systems alike. Paramount to the ecosystem is the preservation of one's privacy and security as information leakage can cost huge financial losses for individuals and corporations. Blockchain technology nullifies these concerns as DDoS, spoofing and other data alteration attacks are prevented outright [23]. Certification is greatly affected by theft of data which is why both BCert and UZHBC make virtual security an essential feature as cyber-attacks are on the rise in today's day and time.

VI. CONCLUSION AND FUTURE WORK

Certification for the next generation is a billion-dollar industry which will inevitably lead to all certification standards to be in the cloud to decrease discrepancies in issuance of certificates of any kind. In this paper we compared two different certification methods that are proposed to see how methods can differ while being on the same network. Ethereum or rather Blockchain as a whole can be leveraged to great effect to impact our educational industry with digitalization of processes and cryptographically encrypting data to ensure anonymity and security fast becoming a necessity. Systems such as these can be implemented simply with the use of Smart contracts and using Solidity to code out data while leveraging the preexisting network, and since Ethereum Merge has occurred, there is very little cost in the issuance of certificates this way. Beyond direct beneficiaries depending on the use case, the ministry for education can increase quality of education and shorten the process of identification and authentication for open competition on a level playing field for all; all while being under a standardized banner under which skills are treated as per pen and paper. We have improved on the existing code and for future implementations, functions of each certification methods can be further enhanced.

For example, for BCert we can:

- Use safemath library to avoid potential vulnerabilities related to integer overflows/underflows
- Add access control methods for certain functions like addCert() and getCert() to segregate different functions according to access of certain parties
- Add error handling of any sort as it is missing so that cases where SID is not found, the code provides feedback to the caller to be corrected

And for UZHBC we can for instance:

- Add a function to revoke certificates as it is possible in the way UZHBC algorithm and it's issuance process works. This is in the scenario the owner's credentials are no longer valid
- Add error handling so that in cases where hash is not found in the "diplomaHashes" array. Currently the function will return false in these cases but a specific error message could be returned to be more useful
- Add a function to update owner's info as the original owner may no longer be responsible for managing the contract
- Add testing and security considerations to test the contract with a variety of inputs with the use of a security analysis tool like Mythril for vulnerabilities

In this way we can build on the existing information for future research work and show how specialized certification cases like ones that require only diplomas for example can leverage a system similar to UZHBC or take the BCert approach and implement their certification method as they fit; with any implementation of any certification method being very simply improved upon as we have shown.

REFERENCES

[1] G. Grolleau, T. Lakhal, and N. Mzoughi, "An introduction to the economics of fake degrees," *Journal of Economic Issues*, vol. 42, no. 3, pp. 673–693, 2008.

[2] D. Vujičić, D. Jagodić and S. Ranđić, "Blockchain technology, bitcoin, and Ethereum: A brief overview," 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2018, pp. 1-6, doi: 10.1109/INFOTEH.2018.8345547.

[3] "The Merge," *ethereum.org*. [Online]. Available: https://ethereum.org/en/upgrades/merge/. [Accessed: 22-Jan-2023].

[4] E. Karafiloski and A. Mishev, "Blockchain solutions for big data challenges: A literature review," *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, Ohrid, Macedonia, 2017, pp. 763-768, doi: 10.1109/EUROCON.2017.8011213.

[5] J. Hallak and M. Poisson, "Corrupt schools, corrupt universities: What can be done?," in *Paris: International Institute for Education Planning*, 2007.

[6] R. H. Sayed, "Potential of blockchain technology to solve fake diploma problem," 2019.

[7] O. Vashchuk and R. Shuwar, "Pros and cons of consensus algorithm proof of stake. Difference in the network safety in proof of work and proof of stake," *Electron. Inf. Technol.*, vol. 9, no. 10, 2018.

[8] C. Holotescu, "Understanding blockchain opportunities and challenges," *eLearning Softw. Educ.*, vol. 14, no. 04, pp. 275–283, 2018.

[9] M. Jirgensons and J. Kapenieks, "Blockchain and the future of digital learning credential assessment and management," *J. Teach. Educ. Sustain.*, vol. 20, no. 1, pp. 145–156, 2018.

[10] M. Turkanović, M. Hölbl, K. Košič, M. Heričko and A. Kamišalić, "EduCTX: A Blockchain-Based Higher Education Credit Platform," in IEEE Access, vol. 6, pp. 5112-5127, 2018, doi: 10.1109/ACCESS.2018.2789929.

[11] BCdiploma and Blockchain digital credentials, "Diplomas and badges on the blockchain," *BCdiploma | Blockchain digital credentials*. [Online]. Available: https://www.bcdiploma.com/en. [Accessed: 22-Jan-2023].

[12] Blockcerts, "Blockchain Credentials," Blockcerts. [Online]. Available: https://www.blockcerts.org/. [Accessed: 22-Jan-2023].

[13] "Free MOOC - UNIC," *UNIC | Blockchain Programs*, 10-Feb-2019. [Online]. Available: https://www.unic.ac.cy/blockchain/free-mooc/. [Accessed: 22-Jan-2023].

[14] "My eQuals New Zealand," My eQuals New Zealand. [Online]. Available: https://www.myequals.ac.nz/. [Accessed: 22-Jan-2023].

[15] E. Leka, B. Selimi, and N. Macedonia, "Bcert – a decentralized academic certificate system distribution using blockchain technology," *Ijits-bg.com*. [Online]. Available: https://ijits-bg.com/contents/IJITS-N4-2020/2020-N4-09.pdf. [Accessed: 22-Jan-2023].

[16] J. Gresch, "An Educational Blockchain for the," Uzh.ch. [Online]. Available: https://www.merlin.uzh.ch/contributionDocument/download/11140. [Accessed: 22-Jan-2023].

[17] D. Ryan, evm-opcode-gas-costs: Gas Costs from Ethereum Yellow Paper.

[18] "Gwei to USD Ethereum gas fee calculator," *Automated Web Tools*, 05-May-2021. [Online]. Available: https://automatedwebtools.com/usd-eth-gas-fee/. [Accessed: 22-Jan-2023].

[19] A. M. Abdullah, "Advanced encryption standard (AES) algorithm to encrypt and decrypt data," *Cryptography and Network Security*, vol. 16, pp. 1–11, 2017.

[20] P. Nivethini, S. Meena, V. Krithikaa, and G. Prethija, "Data security using blockchain technology," *International Journal of Advanced and Applications (IJANA), Special Issue*, pp. 279–282, 2019.

[21] S. Bhandari, "Difference between AES and SHA," *Ask Any Difference*, 16-Jan-2022.

[22] *Etherscan.io*. [Online]. Available: https://etherscan.io/. [Accessed: 22-Jan-2023].

[23] U. Bodkhe et al., "Blockchain for Industry 4.0: A Comprehensive Review," in IEEE Access, vol. 8, pp. 79764-79800, 2020, doi: 10.1109/ACCESS.2020.2988579.