ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803/tg-20200811113729

Preliminary communication

Blockchain-Based Application for Certification Management

Jovan Karamachoski*, Ninoslav Marina, Pavel Taskov

Abstract: Blockchain technology will bring a disruption in plenty of industries and businesses. Recently it proved the robustness, immutability, auditability, in many crucial practical applications. The blockchain structure offers traceability of actions, alterations, alerts, which is an important property of a system needed for development of sustainable technologies. A crucial part of the blockchain technology regarding the optimization of the processes is the smart contract. It is a self-executable computer code, open and transparent, encoding the terms of a regular contract. It is able to automate the processes, thus decreasing the human-factor mistakes or counterfeits. In this paper, we are presenting the feasibility of the blockchain technology in the certification processes, with an application developed for university diploma certification. The example is easily transferable in other areas and business models such as logistics, supply chain management, or other segments where certification is essential.

Keywords: certification; cryptographic hash; distributed ledger technology; Ethereum Blockchain; smart contract; smart record

1 INTRODUCTION

The certification procedures are part of our everyday life. A certificate is a verification of existence or possession of declared characteristics or acquired competences. In most of the cases the certificate is issued by an institution and handed in paper form to the holder. These bureaucratic procedures are time consuming, expensive and leave plenty of opportunities to issue fake documents. The advancement of technology offers a plethora of tools to the scammers to falsify the paper certificates, hence having a technology that offers protection against these malicious activities is quite beneficial.

Blockchain technology shows great potential to avert the corruption of certifications. The characteristics of the blockchain technology, like, auditability, immutability, non-repudiation, transparency, verifiability and irrevocability, make it a perfect candidate for enhancement of the traditional certification procedures. These characteristics make the blockchain technology suitable for any type of applications where certification is needed. This includes the certification of origin, possession, quality level, class, properties, measured parameters, some features, location while tracing the movement, or similar.

This paper introduces one of the pioneering works in the field of blockchain usage as certificate storage. The proposed application shows potential to overcome the long bureaucratic procedures and prevent fraudulent activities during the certification procedures. It uses the blockchain technology, also known as Distributed Ledger Technology (DLT), which is transforming the activities in a trustless environment and still keeps a single truth in the whole system.

The implementation of blockchain technology for building on-line certificate database will increase the commodity of living and ease up the administrative procedures to issue and verify the certificates.

The problem with the fake education diplomas is present in many countries. Especially it is tricky to overcome during the process of mutual recognition of foreign diplomas. In [1], Sayed points out the crucial characteristics of the blockchain technology to overcome the fake diploma problem and mentions few projects related to the application of blockchain technology. Furthermore, the author analyzes the fraudulent activities regarding the fake diplomas and proposes structure for a concrete blockchain-based application to overcome the problems. Besides the potential to overcome the fraudulent activities over the certification and validation process, the implementation of blockchain-based application for certification will also have a financial impact. The financial impact and potential business model are analyzed in [2].

Tariq et al. in [3] are developing a blockchain-based accreditation and degree verification system by the use of Ethereum Blockchain [4]. The uniqueness of the proposal is the implementation of the private version of the Ethereum Blockchain in order to keep the system under Proof-of-Authority consensus mechanism. Similar approach of implementation private type of blockchain, is described in [5]. They are using the Hyperledger Sawtooth [6] enterprise blockchain in order to manage the credentials and privileges in a system. In contrast to these approaches, the application presented here is implemented on a public blockchain. It offers a complete decentralization of the database, while keeping the robustness of the application and managing the credentials of the users.

In addition to the use case of blockchain-based diploma certificate management, there are use-cases where the blockchain technology is used as certificate management mechanism for a birth certification [7], certificate revocation lists [8], green certification in energy sector [9], product compliance and assurance in the construction industry [10], endorsement and forestry certification [11].

2 BLOCKCHAIN-BASED CERTIFICATE STORAGE

The blockchain technology is decentralized data storage structure, capable to operate in trustless community, to track record modifications and to reduce the need of third-parties. Moreover, the blockchain database structure, also known as the distributed ledger, offers liveness, immutability, redundancy and non-repudiation of the records. The database

structure model of the blockchain technology is presented in Fig. 1.

The records in the blockchain database are organized in blocks, where the blocks are generated in predefined time intervals. All the information generated in one blockchain network is stored in every participating node, thus creating a complete copy of the common database of the system, in every participating node. This property makes the database structure redundant, reliable and very robust.

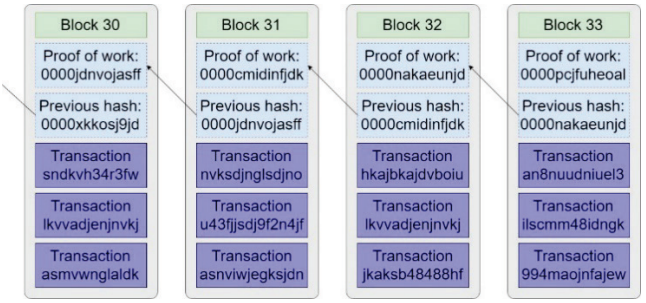

Figure 1 The structure of the blockchain

As presented in Fig. 1, the blocks are connected in a single-chain formation. The connection between two consecutive blocks is through the common data field known as block hash. The block hash is calculated by use of a cryptographic hash function, which is one-way function that creates fingerprint of the block of records, that is unique and irreversible. The block hash is calculated for the current data block and it is set in the header data of the next data block. The implementation of hashing function in the process of block creation and linking the data blocks in this way, makes the record in the database tamper-proof. Practically even the smallest change in a data record will significantly change the block hash, which will represent an attempt for modification of the database. Regarding the type of hash functions, the Bitcoin [12] network uses SHA-256 and RIPEMD-160 algorithms [13], while Ethereum uses the KECCAK-256 algorithm, which is not following exactly the FIPS 202 standard, also known as SHA-3 algorithm [14].

Besides the hash function, the blockchain technology deploys encryption functions to provide security to the user wallet and digital signature for the transactions. The main encryption algorithm is the Elliptic-curve encryption algorithm, using the elliptic curve (EC) $y^2 = x^3 + ax + b$ over a finite Galois field (GF) defined with a prime number p. Encryption algorithms over the elliptic curve work in a way that an algebra over the elliptic curve plus a neutral point at infinity is defined. Using this algebra means that any sum of two different or two same points will give again a point on the same elliptic curve. Adding the point to itself, which is multiplying by two, will again give another point on the curve. Hence we can add the same point to itself multiple times, which defines the operation of multiplication as in the classical arithmetic. Elliptic curve cryptography (ECC) algorithms start from a *base point* G and this point is multiplied by a number that represents the private key k. Using the addition rules of the ECC algebra, one can get a point that is equal to the point $K = kG = G + G + ... + G$. This will represent the public key K that corresponds to the private key k, using the given curve. Bitcoin and Ethereum use the Elliptic-curve Digital Signature Algorithm (ECDSA), with the elliptic curve secp256k1 [15]. This curve has the following parameters:

- $p = 2^{256} – 2^{32} – 2^9 – 2^8 – 2^7 – 2^6 – 2^4 – 1$
- $a = 0$, $b = 7$, making the curve $y^2 = x^3 + 7 \bmod p$
- The coordinate x of the base point G is $G_x =$ 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798
- The order of the base point G is n = FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141
- The cofactor is $h = 01$.

For protecting the data confidentiality in the digital wallet and transaction maintenance, Ethereum additionally uses the Advanced Encryption Standard (AES) [16]. The digital wallet is a file which stores user credentials, basically the private and public keys. The private key is used in the procedures to digitally sign user transactions and decrypt messages, while the public key is practically the wallet address of the user and it is used to encrypt messages sent to another user.

Another important mechanism is the consensus algorithm which helps the network to coordinate the state of the common database. Due to trust-less approach of building the network and possible malicious users in the network, the consensus mechanism offers procedures for decentralized synchronization of the databases owned by the network users. Three most common consensus algorithms are:

- Proof-of-Work (PoW) [12],
- Proof-of-Stake (PoS) [17], and
- Practical Brzantine Fault Tolerance (PBFT) [18].

Every algorithm has its own advantages and disadvantages. PoW has the most reliable properties for an anonymous usage of the network but it is the most energy inefficient algorithm due to heavy computational processes. PoS has less reliable properties but it is more energy efficient than PoW because it implements moderate computational processes. PBFT is energy efficient but it is not feasible for pure public blockchain technologies because it is designed to work in closed networks.

With the deployment of the Ethereum Blockchain technology, the feature called *smart contract* was introduced. Smart contract is a self-executable and transparent code, stored in the immutable ledger of the Ethereum Blockchain. The last property makes the smart contract code impossible to be modified. This code, or more precisely functions from this code, can be invoked by those who have credentials. Often, smart contracts have properties to follow the terms of the regular (legal) paper contracts. In that way it can often replicate and enforce the legal procedures into an automatic machine code. The implementation of smart contracts will provide unambiguous and automated procedures, transparent, reusable and publicly accessible by everyone. By the development of smart contract, that follow the regular

procedures, it is possible to store, check or revoke any information. A function in the smart contract will record the information on the blockchain, thus creating immutable evidence. Additionally, the smart contract can manage the credentials of the application users.

3 APPLICATION STRUCTURE AND FUNCTIONALITIES

3.1 Tools and Platforms

To develop our application, we use the Embark framework (v3.2) [19], which supports the whole eco-system for building a Distributed Ethereum Blockchain application. The Embark framework offers complete environment configuration for the Storage part, blockchain part, Front-end part and Back-end part of the application. Smart contracts are coded into a text editor by the use of the Ethereum custom made object-oriented programming language called Solidity (v0.4.11) [20]. The unique address of the smart contract developed for the application is publically visible in a bytecode on the Internet. The format of an Ethereum address looks like the following 40 hexadecimal character (20 bytes) sequence:

0x79Df2bf2891Be327FC859189c9d1D44eC33Df3d0.

It is obtained by hashing the public key (that corresponds to the user's private key) using Keccak–256 hash function, taking the last 40 hexadecimal characters and adding a '0x' as a prefix. In this particular case we get

0x79df2bf2891be327fc859189c9d1d44eC33df3d0.

This address has to pass the checksum process which is quite simple. First you calculate the Keccak–256 hash of the obtained hexadecimal address without the prefix 0x. If the character number *i* of the Keccak–256 hash is greater than or equal to 8, you convert the *i*-th character of the address to uppercase, otherwise you leave it lowercase. Finally, you add 0x back at the start of the resulting string. The checksum address is the same as the initial one if you ignore the case of the letters a, b, c, d, e, and f.

The application uses the Infura development suite that provides instant and scalable Application Programming Interface (API) access to the Ethereum network and the Internet Planetary File System (IPFS) [21]. By the use of Infura service, the application attaches to the Ethereum network without the use of a fully functional Ethereum node. That is a gateway service for easier deployment of distributed applications (DApps). The current version of the application is deployed on the Ropsten Ethereum test network. Moreover, Infura offers a nice gateway to the IPFS [22] storage, to access the application data. IPFS is protocol for peer-to-peer storage and sharing of files in a distributed fashion. The application content is recorded and hosted on the IPFS network, except for the IPFS record addresses, which are kept on the Ethereum Blockchain. The content on the IPFS storage cannot be erased or modified with the current version of IPFS protocol. This property complemented with the immutability of the Ethereum Blockchain records creates a tamper-proof system. The application address on the IPFS network is as follows:

https://ipfs.infura.io/ipfs/QmTryMjQRq8hVtVDyjTS2ixSa3AGynsKbHCGVNr6VxUqS7/

To access and utilize the Ethereum network every user needs a wallet. The most intuitive way to use the application is in a standard web browser, such a Google Chrome or Mozilla Firefox, with an installed and enabled Metamask extension [23]. Metamask is an application for wallet management where the users have the balance information, address credentials and functionalities for interaction with application located in the Ethereum Blockchain.

3.2 Application Design

The application consists of two functional parts: one part is the certificate identification (CID) number storage procedure on the Ethereum Blockchain via Smart contract and the other segment is the front-end and the back-end of the application stored on the IPFS network. The CID number is paired to the address of the certificate record on the IPFS network.

Our application is purposely developed for university diploma certification process but can very easily be adapted to any other certification process, such as certification of origin of goods, item ownership, or feature possession, among others. The application implements four roles with different read/write credentials:

1. Visitor,
2. Administrator,
3. Staff, and
4. Rector.

The Visitor role is the default role of the application, which is given to every guest on the web site. The other roles: Administrator, Staff and Rector, have several elevated privileges. The Administrator role will have elevated privileges to manage the addresses on the page (associating the correct addresses to the Administrator, Staff or Rector role). The Staff role will have elevated privileges for insertion of newly graduated students and their credentials. The Rector role will have elevated privileges to validate and digitally sign the certificates of already inserted students. The distinct roles will be authenticated through the user wallet.

The current version of the application has five segments:

1. Student search,
2. Insert student,
3. Diploma validity,
4. Address management, and
5. Role panel.

The application organization regarding the credentials and roles is shown in Fig. 2. The Student search segment is used for searching of students by the use of the student CID number and also has verification function if the student was

previously inserted in the system, but still does not have verified the diploma by the Rector. The Insert student segment is used for insertion of students who have graduated. The Diploma verification segment is used to check the validity of the diploma for a given student. The Address management segment is used by the Administrator of the page to manage the addresses for the particular roles in the system. The Roles panel displays the addresses of the particular roles in the system. This panel gives total transparency of the certification process within the institution.

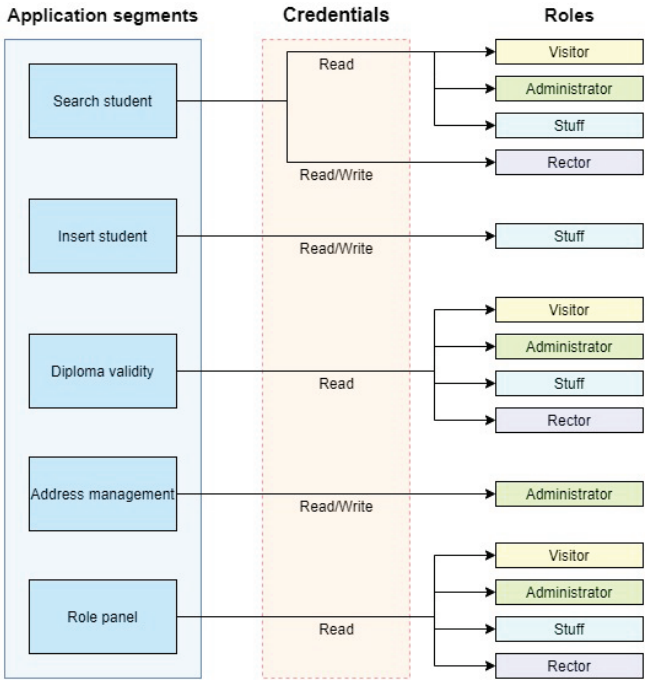

Figure 2 Credentials and roles

The smart contract of the application, beside the standard types of variables, uses variables of type *map* to store the dynamic information of the application. The map variable is a key-value type of variable where one variable can be reflected into another variable. The application implements two maps:

- one is to keep the inserted graduated students, and
- the other is to keep the verified diplomas.

```
if (search_student (CID_number) == Null)
    if (user_credentials == stuff)
        IPFS_address = insert_student (student_information)
        Graduation_record = graduated_student_map(IPFS_Address=> CID_number)
        store_data(Graduation_record)
    end
else
        show_student_information(CID_number)
        if (user_credentials == rector)
        verify_diploma(CID_number)
        Verification_record = verified_diploma_map(IPFS_Address => CID_number)
        store_data(Verification_record)
        end
end
```

Figure 3 Pseudo-code for diploma insertion and verification

The first action is taken over the map for inserting graduated students where the CID of the students is mapped with the IPFS record address, and the second map is populated by reflecting the same IPFS address record from the graduated students map in the verified map. To illustrate the procedure for diploma verification, see the pseudo-code in Fig. 3.

The front-end of the application is presented in Fig. 4. The figure displays the information for a test student, previously inserted in the system. In addition, the graduation of the student is previously verified by the Rector, which is visible in the same figure.

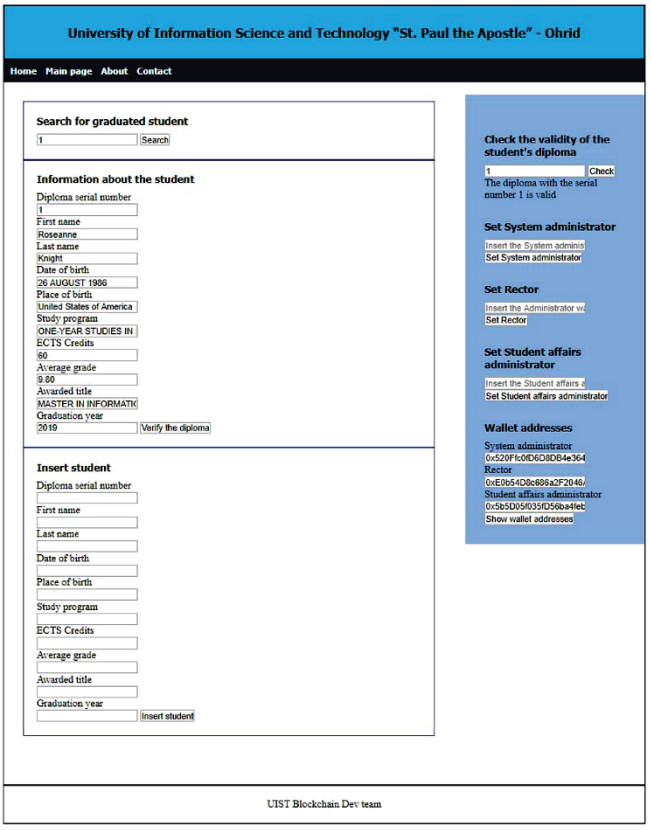

Figure 4 Front-end of the application

Besides the back-end segment for CID storage on the Ethereum Blockchain, the application implements back-end segment for storing the student information on the IPFS network. The current version stores the student information in plain text JSON format. The improvement of the application will implement encryption mechanism to protect the student information and enhance the privacy of the data.

4 FUTURE WORK

The current prototype of the application implements the basic functionalities, sufficient to test the feasibility of the solution and the potential of the application for mass use. Due to circulation of private data in the application, and exposure of these data in the public Internet, the application will be upgraded with encryption module to ensure privacy of the data stored in the application (either on the blockchain or on the IPFS). Also, the application will implement QR codes to make it more interactive and user friendly.

The final version of the application with extended privacy and security will be published on the Main Ethereum network and linked to the official University web site, where the graduated students will get the digital certificate.

5 CONCLUSION

It is crucial to enhance and simplify the administrative procedures. The current certification procedures are prone to falsification, and are very time consuming and expensive. The use of the blockchain technology will have immense impact on the certification processes by decreasing the bureaucratic procedures, shorten the time for certificate verification and skip the third-parties in the processes. This application is proof for the feasibility of the blockchain technology and the certification procedures, by offering transparent, reliable and robust mechanism to prevent the malicious activities.

Notice

The paper was presented at MOTSP 2020 – International Conference Management of Technology – Step to Sustainable Production, which took place from 30th September – 2nd October 2020 in Bol, island Brač (Croatia). The paper is not and will not be published anywhere else.

6 REFERENCES

[1] Sayed, R. H. (2019). *Potential of blockchain technology to solve fake diploma problem*. University of Jyväskylä, JYX Digital Repository.
[2] Oliver, M., Moreno, J., Prieto, G., & Benitez, D. (2018). Using blockchain as a tool for tracking and verification of official degrees: business model.
[3] Tariq, A., Haq, H. B., & Ali, S. T. (2019). Cerberus: A blockchain-Based Accreditation and Degree Verification System. arXiv preprint arXiv:1912.06812.
[4] Buterin, V. et al. (2014). Ethereum white paper: A next-generation smart contract and decentralized application platform. http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf, 2014.
[5] Wegelid, F. (2019). *Storing digital certificates using blockchain*. Lund University.
[6] Olson, K., Bowman, M., Mitchell, J., Amundson, S., Middleton, D., & Montgomery, C. (2018). *Sawtooth: An Introduction*. The Linux Foundation.
[7] Shah, M. & Kumar, P. (2019). Tamper proof birth certificate using blockchain technology. *Int. J. Recent Technol. Eng. (IJRTE)*, 7.
[8] Baldi, M., Chiaraluce, F., Frontoni, E., Gottardi, G., Sciarroni, D., & Spalazzi, L. (2017). Certificate Validation through Public Ledgers and blockchains. *ITASEC*, 156-165.
[9] Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., McCallum, P., & Peacock, A. (2019). Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews*, 100, 143-174. https://doi.org/10.1016/j.rser.2018.10.014
[10] Allison, N. & Warren, M. (2019). Applying blockchain to product compliance and assurance in the construction industry. *Building Research Levy, External Research Report*.
[11] Sylvester, G. (2019). *E-agriculture in action: blockchain for Agriculture Opportunities and Challenges*. The Food and Agriculture Organization of United Nations.
[12] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Available: https://bitcoin.org/bitcoin.pdf
[13] Nist, U. (2001). *Descriptions of SHA-256, SHA-384 and SHA-512*. Available: http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf
[14] Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2011). The keccak sha-3 submission. Submission to *NIST (Round 3)*, 6(7), p. 16.
[15] Certicom, S. E. C. (2000). Sec 2: Recommended elliptic curve domain parameters. *Proceeding of Standards for Efficient Cryptography, Version*, vol. 1.
[16] Standard, N.-F. (2001). Announcing the advanced encryption standard (aes). *Federal Information Processing Standards Publication, 197*(1-51), 3-3.
[17] Thin, W. Y. M. M., Dong, N., Bai, G., & Dong, J. S. (2018). Formal analysis of a proof-of-stake blockchain. in the *23rd International Conference on Engineering of Complex Computer Systems (ICECCS)*, 197-200.
[18] Buchman, E. (2016). Tendermint: Byzantine fault tolerance in the age of blockchains. Available: https://allquantor.at/blockchainbib/pdf/buchman2016tendermint.pdf
[19] *Embark Framework by Status home page*. Available: https://framework.embarklabs.io/. (Accessed: 02-2020)
[20] *Solidity home page*. Available: https://solidity.readthedocs.io/en/v0.4.11/. (Accessed: 02-2020)
[21] *Infura home page*. Available: https://infura.io/. (Accessed: 02-2020)
[22] Benet, J. (2014). *IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3)*. https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf.
[23] *Metamask home page*. Available: https://metamask.io/. (Accessed: 02-2020)

Authors' contacts:

Jovan Karamachoski, MSc
(Corresponding author)
University of Information Science and Technology "St. Paul the Apostle",
Partizanska bb, 6000 Ohrid, North Macedonia
jovan.karamachoski@uist.edu.mk

Ninoslav Marina, PhD
University of Information Science and Technology "St. Paul the Apostle",
Partizanska bb, 6000 Ohrid, North Macedonia
ninoslav.marina@uist.edu.mk

Pavel Taskov
University of Information Science and Technology "St. Paul the Apostle",
Partizanska bb, 6000 Ohrid, North Macedonia
pavel.taskov@uist.edu.mk