ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #27872

BLOCKSCRIPTS – A BLOCKCHAIN SYSTEM FOR UNIVERSITY TRANSCRIPTS

Dr. Ronald P. Uhlig, National University

From 2010-2014, Dr. Ronald P. Uhlig was Dean, School of Business and Management, National University, La Jolla, CA. He returned to the faculty of the School of Engineering and Computing in 2014 as Lead Faculty for the Bachelor of Science in Computer Science program. During 2005-2010 he served the School of Engineering and Technology in multiple positions including Chair of the Department of Computer Science and Information Systems, and Lead Faculty of the Master of Science in Wireless Communications. From 2000-2005, he was President/CEO, SegWave, Inc., an educational technology systems company he founded.

Previous positions include Vice President for Russia and Eastern Europe, Qualcomm Inc., 1995-99, with offices in San Diego and Moscow, Russia and multiple positions with Northern Telecom and Bell-Northern Research in Ottawa, Canada and Richardson, TX during 1978-1995, including Director, Intelligent Network Solutions and Director, Asia/Pacific Strategic Marketing. He is one of several ”Fathers of email”; based on work he did with the US Army and DARPA in the 1970s and several international committees he chaired during 1979-91. Those committees took him to nearly 100 countries globally. He had nationwide responsibility for US Army Materiel Command scientific & engineering computing, 1969-78, pioneering many applications in what has become today’s Internet, and he served as a US Army Officer in the Office of the Chief of Staff, in the Pentagon, 1966-1968.

He holds a B.Sc. in Physics from the Massachusetts Institute of Technology, and a Ph.D. in Physics from the University of Maryland. In 2016 he received the Distinguished Teaching Award from National University. He is the recipient of a Gold Medal from the International Telecommunications Academy, Moscow, Russia, for sustained contributions to telecommunications; the Silver Core from the International Federation for Information Processing; and the Founders Award from the International Council for Computer Communications.

Rich Yonts, Teradata

Rich Yonts is a professional software engineer developing applications in modern C++, Java, and Python. He holds three degrees, two of which are in computer science including a master’s degree. His professional contributions include client/server, API, database, and Apache module development; this work has led to four US patents. Interests outside of work include advanced computing, algorithm and data structure development, and operating systems.

Benjamin W Cashman, National University

Ben Cashman holds a masters degree in computer science from National University and is living in Northern California. In his professional career Ben has worked as a freelance web developer. Additionally he has worked in education both as a teacher in special education and as a university partner. In his time working in education he has worked to invent and simplify the processes in education to enhance the way teachers are able to interact with students using technology. He also has a great passion for music and enjoys sharing it with others.

Richard S. Clark, National University
Brett Nieman

BLOCKSCRIPTS - A UNIVERSITY TRANSCRIPT BLOCKCHAIN

Abstract

Our team created a blockchain solution named Blockscripts to hold a student's transcript information, and or diplomas. The information is stored in a digital format, which can be retrieved easily by the student, faculty, or an interested third party i.e. a university with a prospective doctoral or master's program. Having records in a digital format allows a student to have an official record that cannot be changed, yet be publicly available for authorized viewers, eliminating the need for intermediaries such as the university registrar's office. This is beneficial for the student, the university, and any prospective employers as the process takes less time for verification. Creating and storing records in this way also ensures the digital transcripts and diplomas cannot be repudiated or altered in any way.

Additionally, blockchain technology provides a secure platform for storing transcripts due to the nature of decentralized data storage. As such, if one node in the blockchain network goes offline, other nodes in the network can be accessed for their persistent and accurate copies of the transcript ledger data. This negates the risk associated with a data center, especially in regard to disaster recovery (i.e. fires or earthquakes) which could potentially lose or destroy localized data. Another benefit to be realized from storing transcripts and diplomas on a blockchain is that of the processing time saved. This savings in time for the person requesting the transcript, the registrar's office in producing it, and the third party receiving the transcript can be significant. It has been documented that a typical transcript request can take up to three weeks; by using blockchain technology it is entirely feasible that requesting parties will be provided the requested transcript the same day.

For the Blockscripts Proof of Concept (POC) prototype, we created a User Interface (UI) that allows a student to request a transcript from a particular educational institution, typically a university. The student's transcripts/original data is owned and initially published to a blockchain network by the institution. It is made selectively viewable by encrypting the transcript with either the student's public key or the public key of a third-party entity; the student is able to decrypt his/her data with their private key and a third party is able to decrypt the transcript data with their private key. This ensures that the transcript will be publicly available, yet the contents will remain private and available only to authorized parties. If the student wants the transcripts to be made available to additional other institutions, the request should include each party's public keys in the transcript request. Each transcript request includes a monetary affidavit for the sum of a fixed fee for institutional processing and fixed fees for mining nodes; the institution's administration will receive their portion and the validating network node (miner) takes their portion after providing the service of validation, data storage, and transmission. Use of existing blockchain networks eliminates the need to create another one, one that may take a long time to gain critical mass to guarantee reliability.

Introduction

This team created a blockchain that can hold a student's transcript information and/or diplomas. The information is stored in a digital format that is immutable and can be retrieved easily by the

student, faculty, or interested third party. Having immutable records in a digital format will allow a student to have an official record that cannot be changed yet be available for authorized viewers [1]. This is great for the student, the university, and any prospective employers. Creating and storing records in this way also ensures the digital transcripts and diplomas cannot be altered in any way. Additionally, Blockscripts provides a secure means for storing transcripts due to the nature of decentralized, redundant data storage. As such, if one node in the blockchain network goes down, other nodes in the network which have received the pertinent data block persist and maintain accurate copies of the transcript.

This approach also negates the risk associated with a data center unavailability, especially in regard to disaster recovery (i.e. fires or earthquakes) which would have the capacity to destroy localized data [2]. Another benefit to be realized from storing transcripts and diplomas on a blockchain is the time saved in requesting, processing, and viewing the data. This reduces the processing latency for the requester, institution administration, and any interested parties authorized to read the data. It has been documented that a transcript request can take up to three weeks [3]; using Blockscripts with blockchain technology, we argue that the complete process can occur within the same day, and likely within a few hours.

There is much current interest in the media about blockchain technology. Our working with this technology revealed a significant difference between cryptocurrency and blockchain technology. Many popular articles suggest that these are synonymous. We found it to be true that cryptocurrencies are implemented using blockchains, but this represents only one possible use of the technology. The POC did not depend in any way on cryptocurrency and remains completely orthogonal to any such implementation.

Our Proof of Concept (POC) used a User Interface (UI) that allows a student to request a transcript from an educational institution and a third-party entity who has viewing rights to the data. Blockscripts enables agencies or universities to verify a degree or degrees that are held by a student or graduate. The transcripts once generated are owned and published to the blockchain network by the institution. It is made viewable by encrypting it with either the student's public key or the public key of the interested third-party. In the case of a third-party, the request will contain either the reader's public key, or a site from which to obtain it. The student will be able to decrypt the transcript data with their private key; third-parties will be able to decrypt the transcript data with their private key. Obviously, multiple readers will require multiple encoding actions by the institution.

This approach ensures that the transcript will be publicly available via the blockchain network, yet the contents will remain private and available only to authorized parties. Each transcript request will include a fee for processing the entire transaction; the administration will take its part for handling the request, and at each insertion of a new transaction into the blockchain network, the validating node (miner) takes its fixed fee for providing the service of validation, data storage, and transmission. The single greatest cost involved in this system is associated with the mining of transaction data [4].

The impetus for the work discussed in this paper stems from a project in an online Cybersecurity course in the National University MS in Computer Science Program. The course was

supplemented several years ago with a series of projects in which small groups of students are required to explore key "hot topics" related to cybersecurity. This includes topics such as blockchains, quantum computing, connected vehicles, Internet of Things, and more, all of which have important cybersecurity concerns. The groups are given two weeks to research their topic and produce a 15-minute recorded presentation of their findings which are shared with the whole class. Details of the approach have been discussed by Uhlig et al.[5,6]

The overarching nature of the topics leads some groups to want to pursue further research in their topic. The work discussed in this paper is the result of one such group's Master's Capstone project.

System Characteristics

Blockchain technology should be viewed as a group of computers (nodes) that are networked to each other over the Internet, and not linked to any single central server. Computers within this network work together to define and also agree on the shared state of data which they all individually hold. These computers adhere to the constraints put forth by the majority of the system which lends to strength-in-numbers for security of the data within the system. The shared state between the nodes can be thought of a distributed state machine where every new block added to the network creates a change to the known and current shared state of the network [7]. When a user accesses a network node, they will be unaware of the backend network supporting the blockchain or any particular implementation details of the accessed node. That is to say when they make a request on the blockchain they will access the network using an interface which is analogous to using a browser. Requests are sent in transactions to the node, bundled into potential blocks, mining is performed, and if successful, a new block is created and linked into the existing chain. Duplicated work (duplicated blocks) are arbitrated by the majority of the blockchain network to determine which node is considered the winner. Winning nodes will collect fees associated with the set of transactions within the new block.

The system proposed by our team is a software solution using blockchain technology of an existing network to store student transcripts provided by academic institutions. The transcripts stored in this network are made available upon request of the person for whom the transcript applies. Our prototype was built using our university's transcript system as the primary administrative source. A proof of concept network was developed to ensure the efficacy of the system, but without the complexities of actual fees or other remunerative aspects. However, in an actual implementation, these issues must be considered.

The use of an existing network is recommended in order to guarantee reliability. A network of too few nodes could easily be attacked by a small set of hostile nodes, gaining 51% or more of the network and creating havoc. Existing public networks are sufficiently large to ensure securely storing and distributing a student's transcript such as Ethereum [8]. Additionally, this system will effectively reduce the cost of obtaining transcripts while also increasing speed of delivery to the requesting party [9]. This system eases the burden of furnishing transcripts to students, generating a more reliable, speedy, and cost-effective solution. The system also provides an easy to use portal for third-party agencies (e.g., potential employers of the student).

In the case of diplomas, where PII (personally identifiable information) is not a concern, the data can be stored in the open in a public block along with any other non-secure information. This allows for a quick fact check, without necessitating a full transcript request.

A blockchain is a public ledger, storing information in a secure manner. After data is entered into a network node, after sufficient data arrives or at a set interval, multiple nodes begin the process of mining. This is typically done by generating a checksum, typically using a SHA-256 algorithm[16], of all user and some blockchain specific data—these data plus checksum becomes a new block. The new block is linked to the newest, already stored block by its checksum value. The computational effort of the new block, from which the mining effort expends costed resources, is to find a given nonce value that makes the new block's checksum match a pattern. This pattern is arbitrarily difficult and is set by the network. Nodes race to be the first to compute the complete block, and after which, the block is published across the network. If other nodes receive the block and verify the checksum (of the data and the nonce), they agree that the block is valid, and the block is then added to the chain. If multiple nodes believe they are the first to create a block, the tie is broken whenever the node owning the longest chain is discovered. As such shorter chains are rejected in favor of the longer one [10].

Some desirable characteristics of a blockchain implementation include the following: *Transaction correctness*: requests for transcripts fit a well-defined pattern of using encrypted sender identification, request type, monetary based information, and any third-parties. *Non-repudiation*: given that the network is public, it is untrusted. The block data therefore necessitates a hard-to-compute mining operation, with its complementary easy-to-verify validation. Arbitrarily altering information and attempting to push a falsified block into the chain generally costs more than the benefit gained. This makes denial of a transaction unwieldy and highly expensive. *Audit tracking*: unlike many current computing systems, Blockscripts uses the public ledgers to ensure a constant audit tracking—the data itself in the chain is the audit trail. *External resources*: the entirety of Blockscripts' interface is public facing, using public nodes. Therefore, the system runs piggyback to existing infrastructure [11]. *Scalability*: the scale of the network system is assured by being a distributed set of nodes that, while they do not trust each other, are certain to arrive at a consensus simply by statistics. *Capacity*: mining nodes need be only large enough to fit a working set of blocks in memory and others in any supporting storage mechanism. Compression techniques are known for minimizing storage requirements of the chain [12] while preserving data integrity. *Availability*: the network is resilient, it is designed to support nodes dropping at any time and in any quantity, as long as users can access at least one node. Theoretically, one node could be the entire network system, although as the number of nodes decreases, the reliability of consensus drops proportionally. *Reliability*: given the nature of the multi-node network, no specific node need remain immediately consistent. Missed blocks, correct but not accepted blocks, too-long hash computations, or fraudulent blocks will be detected and corrected in an eventually consistent basis. *Data integrity*: the nature of sealed blocks using checksums prevents fraud by making detection of single- or multiple-bit changes obvious and is easily verified. The expense and unlikelihood of successful injection attacks of altered blocks is the primary deterrent of fraud [13]. Put simply, it is more profitable to be honest.

Basic Operation

Some current blockchain networks provide anonymity to all parties by providing one-time use access points for senders and receivers. This prevents discovering a particular party's identity and eliminates attack vectors. However, for publicly accessible data like transcripts, this requirement is largely unnecessary. Our proposal is to have all institutional parties wishing to be part of the Blockscripts network register for and provide to students a unique id. Many third-party organizations would probably do this also. The benefit of using ids is that messages can be sent to mailboxes associated with the id so that institutions can implement event driven processing. Without the mailbox concept, there would remain some manual steps in handling requests, obviating the value of full automation. Students could also register for an id, either temporary (duration of the transaction from request to third-party receipt) or permanent.

The student will negotiate with a third-party for employment or admission which begins the process of requesting a transcript. The student will need the third-party's id or public key. With this information, he accesses the academic institution's web page pertaining to requesting a transcript. This interaction determines the information necessary for a properly formed transcript request.

Using these data, the student creates a transaction targeted to the blockchain network. The student-initiated transcript request is received by the academic institution which will produce a copy and send it to the interested third-party, whether employer or other academic institution. Contained in the request is the institution's set of required information such as student name, id, affidavit of funds, third-party id or public key. The request is created and packaged as a transaction. The transaction is sent to some node in the blockchain network, where it is built into a valid block. This block is broadcast to a subset of the network known to the node. Upon the block's acceptance at the various nodes, based on their consensus algorithms, it becomes a permanent public record. After the block is accepted, notifications of the block sequence number and transaction id is sent to the student and the institution's inbox.

It is assumed that academic institutions would have to retool some of their existing processes to handle transcript requests in an event-driven model from existing batch processing model; this is essential for the near real-time handling of these requests. Upon receipt of the transcript request information (block number and transaction id), the administrative process creates a transcript for the student. The block number and transaction id information contained within the record in its inbox determines the proper work flow and initiates the building or retrieval of the proper transcript for the student. Based on the recipients, the transcript is encrypted with the proper public keys, either supplied in the request or obtained from the party's registration record. Financial handling occurs to verify the funds and take a proper portion. It is assumed that financial records are handled by an independent agency, either a bank or a different blockchain dedicated to financial transfers.

The transcript is encrypted with the registered third party's public key which is obtained by requesting it from the Blockchain network. The Administrator builds a transaction request and adds the third party's id (recipient), the Student's id (interested party), and the encrypted

transcript. Upon publishing and acceptance, the Blockchain network informs the recipient and all interested parties of the block and transaction id.

After the institution has prepared the transcript and received funding (either directly or via promissory note), it creates a new transaction per interested party containing the proper encrypted transcript and sends it out to the blockchain network. Notification is sent back to the student's mailbox, updating the progress of the original request. The blockchain network receives the transaction, and upon bundling in a new block and its approval, appends the block in the public record. Upon publishing and acceptance, the blockchain network informs the recipient and all interested parties of the new block number and transaction id. The third party receives notification in its inbox and requests the proper block from the blockchain network. Based on the administrative details of the request, the third-party determines the proper processing action, retrieves the encrypted transcript from the block by its transaction id, decrypts it using its paired private key, and then forwards the transcript to the proper person or group. The transcript is now available to the third-party, bypassing much or all of the normal manual process.

Typically, within a network of mining nodes, different nodes compete to be the first to discover the proper POW. Different strategies exist for finding the proper nonce value, the simplest is to increment it after each failed attempt. Another is to randomly pick values; other, more sophisticated approaches could be used, but the more sophisticated, the more time and compute intensive they become. Therefore, just about any node has a reasonable chance of being first to successfully compute the proper POW. The first node to discover a correct value publishes the block along with its POW. Other nodes will concede if they determine that the checksum accurately describes the block and that it matches the network's specified pattern.

Mining operations are expensive. Therefore, miners expect some return on their computational investment. Two schemes in current use are currency generation [13] and transaction fees [14]. While the prototype used neither, it is conceived that the transaction fee scheme is the most likely to be used in a commercial enterprise where requesters are doing business with servicers since government backed financial instruments are accepted more widely than unbacked cryptocurrencies. A third option, unpublished but conceived by the authoring team, was to have requesting entities generate good will by providing evidence that they have mined (spent computational resources) prior to being able to make their own requests—a hybrid approach. This could blend the ideas behind cryptocurrencies without depending on financially backed instruments.

Prototype

Our team implemented an experimental blockchain network as a POC based on published work[15], we developed a Python 3 application using the Flask framework to process ReST interfaces.

The main algorithm used for encoding the Blockscripts information application is SHA-256 [16]. Each block, except the initial genesis block has a link to the previous block by using its

checksum. The genesis block has a sequence number of zero and a checksum computed over a limited set of data. The only information available from the genesis block is the checksum, the rest is well controlled by the network node which prevents its exposure.

A block uses the SHA-256 hash of the payload data (initial request or transcript data), the block's sequence number, and the nonce value. The timestamp data of the block is not hashed as the validator would have no way to know when the block was created. The checksum function is a one-way function; its computation is easy and inexpensive, but its reversal is extremely difficult [17] making fraudulent blocks very time and compute resource expensive. Therefore, the immutability of a block is guaranteed by its quickly computed hash. If any bit—or group of bits—changes, the SHA-256 hash will change in significant ways, making detecting of mutations fast and reliable. This same hash algorithm is used for both validation of integrity and to generate the Proof-Of-Work.

The Proof of Work (POW) for creating a new block is implemented in a loop that starts with an initial nonce value of zero. Within the loop, the checksum is computed from the relevant data and housekeeping fields within the block. The resulting checksum is then typographically compared to a pattern and, if it does not match, the nonce is incremented and the loop repeated. This process repeats until the most recently computed checksum matches the pattern. Our pattern was adopted from the Bitcoin model where the computed checksum starts with a defined number of zero characters. In other implementations, the specific pattern or matching algorithm can vary within the constraints of the network depending on computational difficulty—the more difficult, the more secure. Regardless of which hashing algorithm is used, the checksum becomes the thumbprint of the block and guarantee its immutability.

The Use Case diagram (Figure 1. *Use Case Diagram*) demonstrates the relation between all of our conceived actors and their relation to the needed service(s). The main idea behind the use case is that we need a blockchain to be able to publish immutable student degree information which can be viewed by any authorized party at any time. The basic concept behind the use case diagram is to illustrate that students will need to make requests to have their information published on the blockchain. If the information being requested is valid, the appropriate transaction will be published onto the blockchain which all authorized actors will be able to view.

The activity diagram (Figure 2. *Activity Diagram*) above shows the complete process of actions and information transactions across the blockchain. Initially, a student will need to make a request. This request is published to the blockchain where an academic institution process verifies requirements and handles the request accordingly. This creates a new transaction which is published onto the blockchain network. Upon sending the updated information to the blockchain, all of the active nodes will verify the transaction and then publish it in a new, validated block where it will be permanent and visible by anyone. In order to view a specific student's information, the party will obtain the block containing the information of the identified student, extract its transaction, and decrypt.

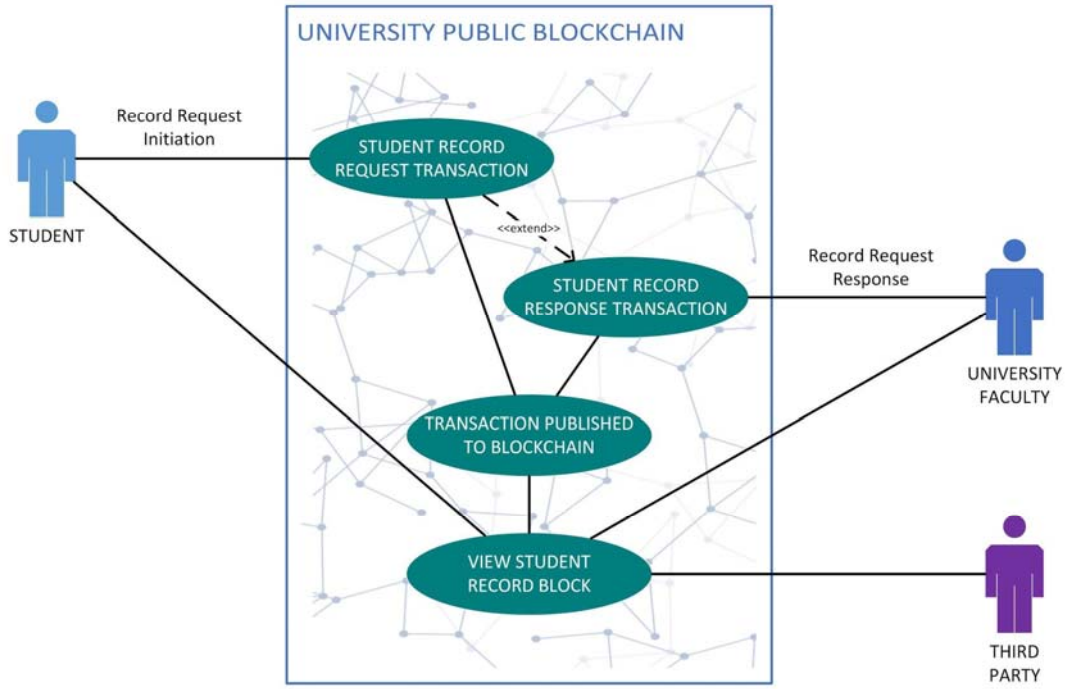

Figure 1. Use Case Diagram

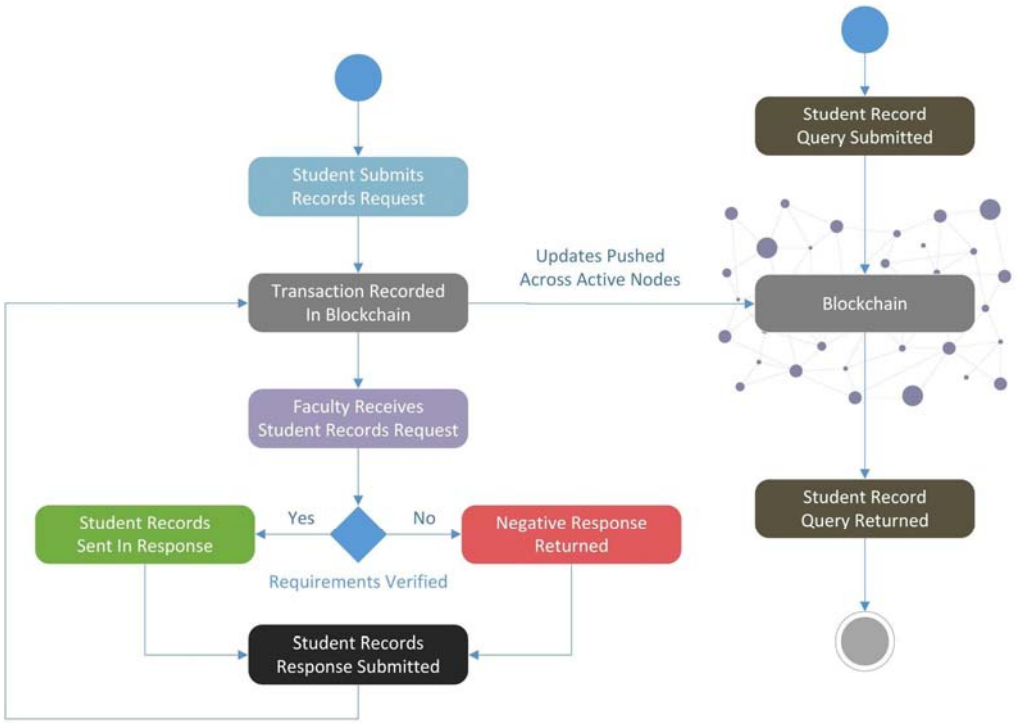

Figure 2. Activity Diagram

The diagram below (Figure 3. *Typical Block*) illustrates a layout of how the transcript data is stored in a block, and its relation to the blockchain. This approach is similar to a traditional relational database only in the sense that each block maintains a unique primary key; in this case it is the block's sequence number [18]. As a foreign key, it maintains a copy of the previous block's checksum value computed by the hashing algorithm [19] in the block header. This allows a linked chain by key, but without necessitating any particular storage mechanism. In cases where saving storage space is necessary, older blocks can be compressed using a Merkle tree approach [19].

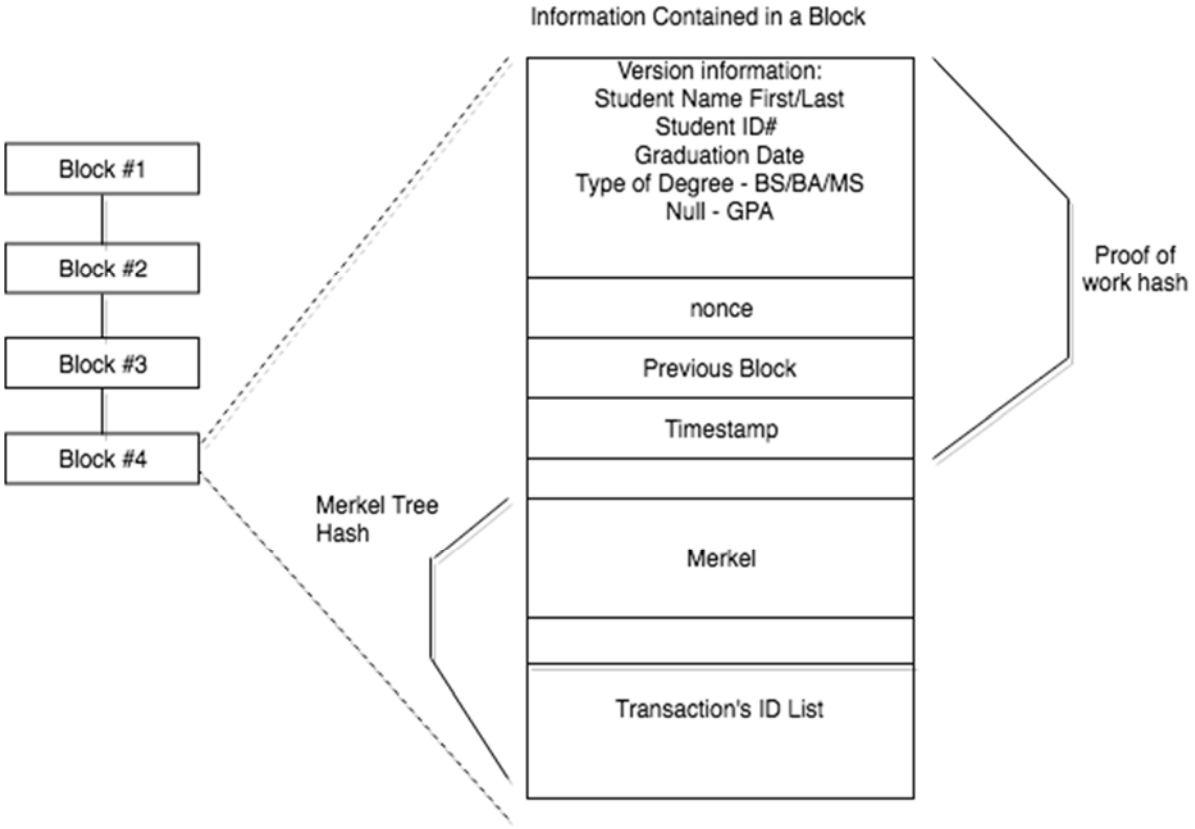

Figure 3. Typical Block

The prototype demonstration included a small number of nodes that were interconnected—each node which came online contacted an existing node (except the first to come online). The contact handshake sent a list of known nodes to the new node. From this list, other links were forged.

Afterwards, a transaction containing a transcript request was created and sent to a node. The transaction was tagged with a 32-character hexadecimal value, essentially a UUID (universally unique identifier). The transaction with identifier was published to all other known nodes. Each node that receives a new transaction adds it to its list of as-of-yet-unprocessed transactions. It then communicates the transaction to all other nodes it knows about. If one of those nodes has not yet seen the transaction, which is determined by its identification string, it is added; if it has already seen the transaction, it is ignored.

The next step of processing was to have nodes perform the mining operation based on some criteria: we chose to use a manual step to better control the demonstration. In a real system, the trigger to initiate block creation could be: a threshold of awaiting transactions, the expiration of an interval, or some specific algorithm that decides that the time is ripe to attempt the mining operation.

This gave each node a roughly equal chance of mining the new block. Upon a valid computation of the checksum, the first node broadcast the new block to its list of nodes. These nodes accepted the new block only if it maintained the invariants of a proper checksum to the sequentially previous block and that it contained the checksum value that matched the POW pattern and was validated by recomputing the checksum over the proper fields of the block. Valid blocks were then accepted into the blockchain and broadcast to a subset of the network. As with adding transactions, the nodes will either add the new block or respond that it already has that block. In the case of any conflict arising from duplicated blocks generated by different nodes using the same transaction and then broadcast around the same time, consensus was used to determine the winner. This scheme ensures that no single node, or group of nodes less than the consensus limit can affect the validity of the data, thus ensuring accuracy and integrity [17].

Conclusion

Our team created a blockchain network which we used to store and distribute pseudo-transcripts and diplomas in a POC manner. This project helped in clarifying issues and benefits to academic institutions and third-parties when processing and viewing transcripts of students in a blockchain. The technology guarantees to a high degree the validity and non-modifiable nature of critical records. It also benefits students requesting their transcripts should a future employer or graduate school request validation of their credentials by supplying the information in hours, when the interest and recollection of the third-party agents is still fresh. We demonstrated that utilizing blockchain technology will reduce the overall time needed to process and access transcripts from three weeks to the same day of the request. Additionally, the immutability provided by the blockchain technology will make it nearly impossible to falsify or modify these documents, giving great certainty to the interested third-parties.

Blockchain technology provides network users with decentralized data storage, creating a significant increase in data protection and speed of recovery. Data stored in a redundant manner makes our system less prone to malice, natural disasters, and cyber-attacks. The encryption utilized in our blockchain allows a user to store sensitive or private information such as a transcript on the blockchain, while giving it public access. This is accomplished without having to concern about third-parties improperly gaining viewing access to the stored information without having the proper private key. Lastly, the cost of transcripts could be reduced by automating the requesting process with blockchain technology, freeing up resources in a registrar's office or human resources department.

Our work with blockchain technology demonstrated its ease of use as a technology, that it does not depend on any particular technology or technology stack and is amenable to amateurs and

professionals alike. The key idea is that of properly signing the payload data with a one-way hash that makes undetectable modification extremely expensive, yet verification trivial. This can be done in myriad ways as long as the agreed upon hashing algorithm is used. Our argument is that any university, or other institution, that wishes to publish data publicly, non-refutably, unalterably, yet securely, can, with minimal effort, implement blockchain technology.

This work also demonstrates the effectiveness of requiring small groups to complete relatively small projects on current "hot topics" in computer science and engineering in courses throughout a degree program as a way to stimulate student interest which some will then choose to explore in more depth in a Capstone Project.

Bibliography

1 Lewis, A. (Feb. 29, 2015) Bits on blocks, a gentle introduction to immutability of blockchains. Retrieved on July 20, 2018 from https://bitsonblocks.net/2016/02/29/a-gentle-introduction-to-immutability-of-blockchains/

2 Posey, B. (Aug, 2014) *Redundant Cloud Storage Ensures High Availability.* Retrieved on July 20, 2018 from: https://searchstorage.techtarget.com/feature/Redundant-cloud-storage-ensures-high-availability

3 Concordia, U. (2018) *Transcripts.* Retrieved on July 20, 2018 from: https://www.cui.edu/studentlife/registrar/index/id/2722

4 Orcutt, M. (Nov 16, 2017). *Blockchains use massive amounts of energy-but there's a plan to fix that.* Retrieved on July 20, 2018 from: https://www.technologyreview.com/s/609480/bitcoin-uses-massive-amounts-of-energybut-theres-a-plan-to-fix-it/

5 Uhlig, R., Sinha, B., Jawad, S., Dey, P., Amin, M. (Aug 2017). *Enhancing Student Collaboration for Improved Learning,* published in the Journal of Modern Education Review (JMER). Volume 7, Number 8, August 2017

6 Uhlig, R., Jawad, S., Dey, P., Amin, M., Sinha, B., (Jun 2018). *Enriching Responsiveness to Enhance Student Learning in Online Courses,* 2018 Hawaii Universities International Conferences on STEM/STEAM and Education, Honolulu, HI, June 2018

7 Pluralsight (Oct 11, 2017). *Blockchain architecture.* Retrieved on July 20, 2018 from https://www.pluralsight.com/guides/blockchain-architecture

8 Blockgeeks (2019) *What is ethereum? The most comprehensive guide ever!* Retrieved on January 6th, 2019 from: https://blockgeeks.com/guides/ethereum/

9 Durant, E. (2017). *Digital diploma debuts at MIT.* Retrieved on July 20, 2018 from: http://news.mit.edu/2017/mit-debuts-secure-digital-diploma-using-bitcoin-blockchain-technology-1017

10 Ritesh (Dec 24, 2017) *Blockchain 101 -- only if you 'know nothing'!* Retrieved on Jan 6th 2019 from: https://hackernoon.com/blockchain-101-only-if-you-know-nothing-b883902c59f7

11 Tar, A (Jan 17, 2018). *Proof-of-work, Explained.* Retrieved on Jan 6, 2019 from: https://cointelegraph.com/explained/proof-of-work-explained

12 Blockchain C. (2017). *The problem of data scaling in the blockchain has been solved.* Retrieved on January 6th 2017 from: https://medium.com/@credits/the-problem-of-data-scaling-in-blockchain-has-been-solved-169f0d39ea07

13 Nakamoto, S. (nd). *Bitcoin: a peer-to-peer electronic cash system.* Retrieved on July 20, 2018 from: https://bitcoin.org/bitcoin.pdf

14 Bitcoin Wiki (2018). Transaction fees. Retrieved on July 20, 2018 from:https://en.bitcoin.it/wiki/Transaction_fees

15 Kansal, S. (2018). Developing a blockchain application from scratch in python. Retrieved on July 20, 2018 from: https://www.ibm.com/developerworks/cloud/library/cl-develop-blockchain-app-in-python/index.html

16 NIST (n.d.) Descriptions of SHA-256, SHA-385, and SHA-512. Retrieved on July 20, 2018 from: https://web.archive.org/web/20130526224224/http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384-512.pdf

17 Pfleeger, C. P., Pfleeger, S. L, and Margulies, J. (2015). *Security in computing*, page 799, Prentice Hall, Upper Saddle River, NJ.

18 Tutorialspoint(2018). *DBMS Data Schemas.* Retrieved on July 20, 2018 from: https://www.tutorialspoint.com/dbms/dbms_data_schemas.htm

19 Cosset, D. (Dec. 27, 2017). *Blockchain: what is in a block?* Retrieved on July 20, 2018 from: https://dev.to/damcosset/blockchain-what-is-in-a-block-48jo