Blockchain certificates: a prototype implementation for digitising educational certificates

Richard Otolo,[1] Eunice Maingi,[1] Joseph Sevilla[1]*

[1]@iLabAfrica Research Centre, Strathmore University, Nairobi, Kenya

**rotolo@strathmore.edu*

Abstract

The prevalence of fake academic certificates in the country is worrying. This has come to light in the media during the vetting process for different jobs such as when recruiting candidates for key public sector positions. The fraudulent use of fake qualifications is a fact that educational institutions and employers have to face. In order to curb this trend, institutions have taken measures that usually involve a third party to verify the authenticity of educational certificates. The main weaknesses of this approach include the time taken to verify certificates is too long. Additionally, the fact that there is an intermediary introduces aspects such as extra costs and the potential inaccessibility to institutions in all geographies. This research project proposes to set up a system that not only makes it easy to maintain records of academic achievement but also make the fast and effective verification of a certificate's authenticity possible. The system will rely on distributed ledger technology also known as blockchain, to accomplish this goal. A blockchain is a trustworthy, distributed digital ledger that contains records of transactions that are replicated across many systems. The records are cryptographically guaranteed to be immutable. This research intends to leverage these and other features of distributed ledgers in order to develop a system that can address the issue of fake certificates.

Background

Current systems of issuing and verifying educational certificates are faced by challenges revolving around the issue of trust. Educational institutions face brand and reputation damage when fake certificates are issued in their name. Employers lack a fast and cost-effective way of providing the validity and authenticity of educational certificates that potential candidates present them with. This gap has led to widespread cases of misconduct by unethical parties. In an attempt to address these challenges, this study describes the designing, developing and testing a blockchain based verification system for educational certificates.

Universities and other institutions of learning issue certificates as an indication that the recipient has met a particular set of conditions required to acquire certain skills. The university's role is to issue, store, verify and validate these certificates. Job seekers have to engage in a slow, complicated and often expensive process in order to obtain official transcripts from their former schools. Additionally, potential employers have to contact the job seeker's alma mater in order to ensure that a transcript is authentic and there is no way to really know for sure.

Blockchain technology can be used to develop new systems that address these challenges. A blockchain is a ledger that records groups of transactions which are linked together cryptographically in a liner sequence. The transactions represent anything that can be represented digitally such as educational certificates. These transaction entries are transparent, permanent and searchable by members in a particular community. Thus, blockchain technology makes use of strong cryptography to make it easy to detect if an educational certificate is legitimate. Additionally, fraud can be eliminated through the use of public key infrastructure to authenticate both the issuing institution and the recipient of the certificate.

An example of how this would work is, a job seeker wanting to apply for a job through an employer's human resource system, would eventually paste a link to their certificate stored on a blockchain. The human resource system would then use an independent blockchain verification service to verify that the certificate is legitimate.

A certification system is a process by which a certificate is issued as proof of a certain claim. In education, certifications are issued as evidence of achievement of learning outcomes, instructor competence, evidence of meeting criteria and proof of authorization (Grech, Camilleri, & Inamorato, 2017). According to (Grech et al., 2017) the certification processes consist of three main steps:

1. *Issuing:* In this step, the claim, issuer, evidence, recipient, and signature are recorded onto a certificate. Additionally, this information can be stored in a centralized database of claims.
2. *Verification:* in this step, a third party verifies the authenticity of the certificate. This includes verification of security features built into the certificate such as a seal, verification with the original issuer or verification against a centralized database of claims hosted by a third party.
3. *Sharing:* in this step, the certificate recipient shares the certificate with a third party. Three ways of sharing certificates include directly transferring the certificate or its copy to the third party, storing it with a custodian authorized to share it, publishing it in a public registry where anyone can access it.

Traditional certification systems face a number of challenges. In order to verify a university diploma, both employers and job seekers have to contact the university or a third-party service to request for official transcripts. This method is both cumbersome and susceptible to fraud as well as costly since typically universities charge a handling fee for each transcript. For example, at the Massachusetts Institute of Technology, "the base cost for a transcript is $ 8.00 with a $2. 00 handling charge for each transcript ordered online (Office of the Registrar, n.d.).

Paper certificates are not immune to the risk of forgery; thus, the issuer must maintain a central database of all certificates issued. This necessitates building manual processes around maintaining the database and answering queries on the validity of certificates. Also, it is not possible to revoke a certificate once it has been issued without the permission of the owner (Grech et al., 2017). Digital certificates were created to address some of these challenges, but they also introduced new weaknesses.

They are subject to risks that affect their confidentiality, integrity, availability, and authenticity. Digital signatures, which require a third party, are necessary to guarantee the integrity of a certificate. Use of proprietary systems results in certificates that can only be used in specific software systems. Registries of digital certificates are at risk of cyber-attacks and data leaks, taking into account that a primary way people use to share digital certificates is email which is usually not secure (Grech et al., 2017).

These challenges point to the need for a more robust certification technology. In 2016, the MIT Media Lab released their blockchain based credential system and made it available for researchers to test. This study investigates how the use of the MIT system can be used to address the above challenges.

Method

This section discusses the software methodology that will be used to design and implement a prototype of the blockchain based educational certificate management system.

The Agile software methodology was proposed by the Agile team in 2001. It describes how a team of software developers can deliver working software to users at regular short intervals called sprints (Dingsøyr, Nerur, Balijepally, & Moe, 2012). This project proposes to use four main phases from the framework as described below.

Requirements Gathering

The goal of this phase is to find out what the functional and non-functional requirements of the system are. These requirements were gathered through a thorough review of the existing literature describing blockchain based educational certification systems. These are the capabilities, basic functions, and processes that the prototype is expected to perform. They include the following:

i. Certificate Issuance: The prototype should allow a certificate issuing institution to sign and issue a certificate onto a blockchain.
ii. Certificate Verification: the system should enable educational institutions, employers and other stakeholders to verify the authenticity and integrity of the certificates in question.
iii. Certificate Viewing: the prototype should allow the certificates to be viewable via a web browser.

System Implementation

This study used the Blockcerts open credentialing system (Blockcerts, n.d.-b) to build a prototype of a system that issues and verifies blockchain based educational certificates on a test network. The Blockcerts system was launched by Massachusetts Institute of Technology in collaboration with Learning Machine, is a Python-based framework makes use of open-source tools and libraries and a mobile app in order to build a decentralized certificate verification ecosystem anchored on the Bitcoin blockchain. The framework has all the components needed for creating, issuing, viewing and verifying certificates across any blockchain.

An awarding institution using the Blockcerts system would release a credential by sending a Bitcoin transaction to a recipient student. A hash value of the certificate would be attached to the Bitcoin transaction. The framework then allows an independent verifier, such as an employer to verify the authenticity of a credential by accessing the hash value of the credential on the blockchain and comparing it with a local digital file. In order to avoid high transaction fees associated with sending each credential as a unique transaction, the MIT researchers merge the hash values into a Merkle tree and then publish the Merkle root to the blockchain (MIT Media Lab, 2016).

The section below discusses the implementation of the system from the viewpoint of the prototype workflow which shows how the system works from the user's point of view, the system and database architecture which show the engineering of the system and the coding implementation which describes the structure of the code.

Prototype workflow

To implement the design in the previous section, the workflow depicted in Figure 1 was created. There are three main actors, the issuer, the recipient and the verifier.

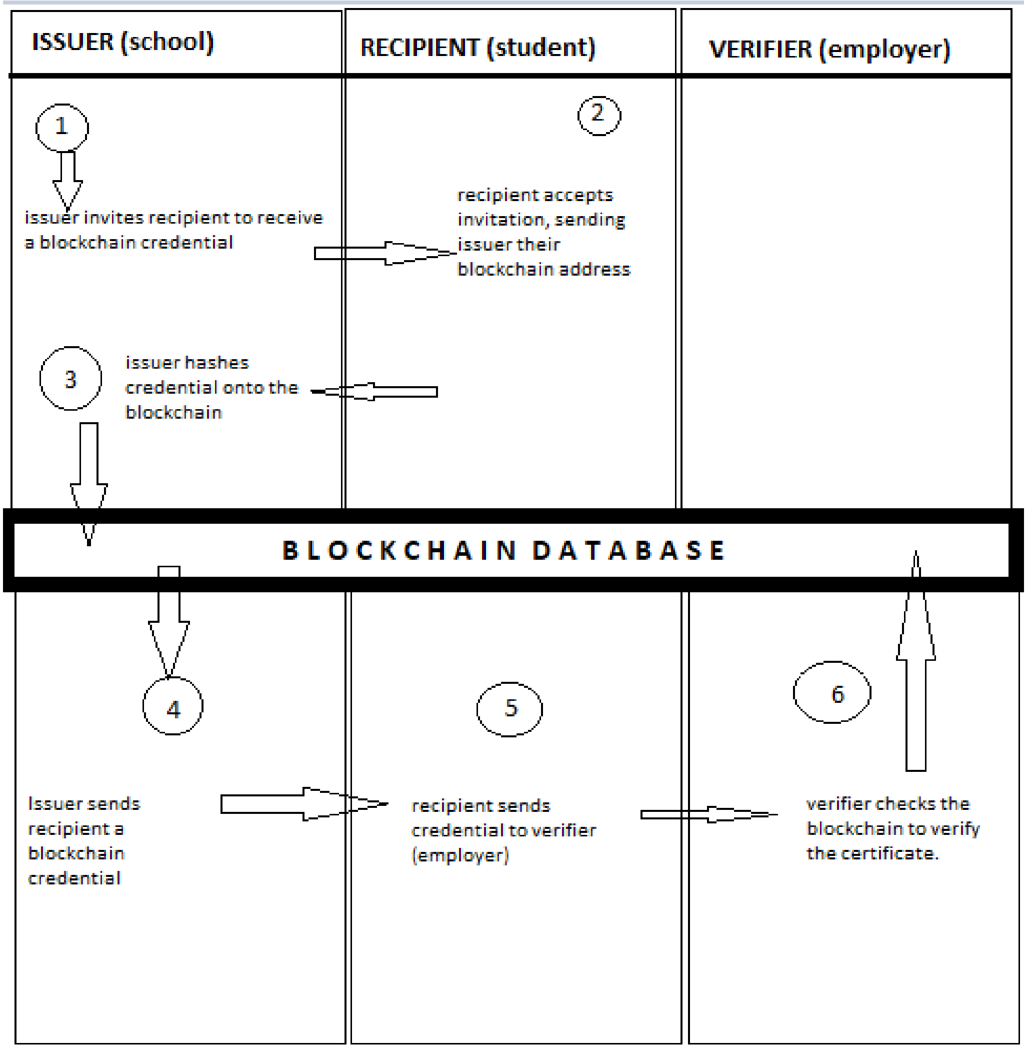

ISSUER (school)
RECIPIENT (student)
VERIFIER (employer)
1
issuer invites recipient to receive a blockchain credential
2
recipient accepts invitation, sending issuer their blockchain address
3
issuer hashes credential onto the blockchain
BLOCKCHAIN DATABASE
4
Issuer sends recipient a blockchain credential
5
recipient sends credential to verifier (employer)
6
verifier checks the blockchain to verify the certificate.

Figure 5 Workflow of the blockchain based credential prototype

The workflow is as follows: The school sends an email to a student inviting them to receive a blockchain based credential. The student responds by sending their public blockchain address. The school merges the credential with a transaction sent on the blockchain. The issuer then sends a JSON-based blockchain credential once the transaction is confirmed by miners on the blockchain. The recipient then provides the JSON-based certificate to an employer when he or she applies for a job. The employer verifies the certificate by checking the blockchain.

System and Database Architecture

The system consists of four main components that handle the Blockchain certificate: issuing applications, verification applications, a local database and the blockchain as shown in Figure 2.

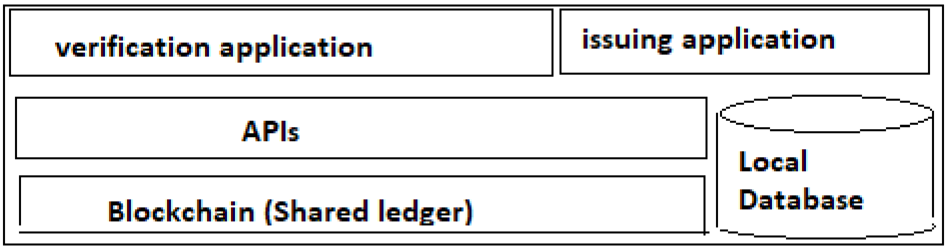


Figure 6 System architecture

The issuing applications are used to design certificate templates, instantiate a batch of certificates and to issue blockchain certificates. In this system, this process is carried out by two modules. The cert-tools module and the cert-issuer module.

The cert-tools module is used to design the digital certificate template and instantiate a batch of certificates. A digital certificate is a JSON file with the necessary fields needed for the cert-issuer module to place it on the blockchain. The cert-issuer module then takes these JSON files and issues them on the Bitcoin blockchain. It does this by creating a transaction on the Bitcoin blockchain from the issuing institution to the recipient. The transaction contains a hash of the certificate.

The verification modules enable anyone to verify the integrity and authenticity of the blockchain certificates. A Blockchain certificate consists of the content as well as other parameters needed for the verification process. Appendix A has an example of a blockchain certificate. The cert-viewer module is used to display and verify the blockchain certificate after they have been issued by the cert-issuer module. This is useful when running in a local test environment.

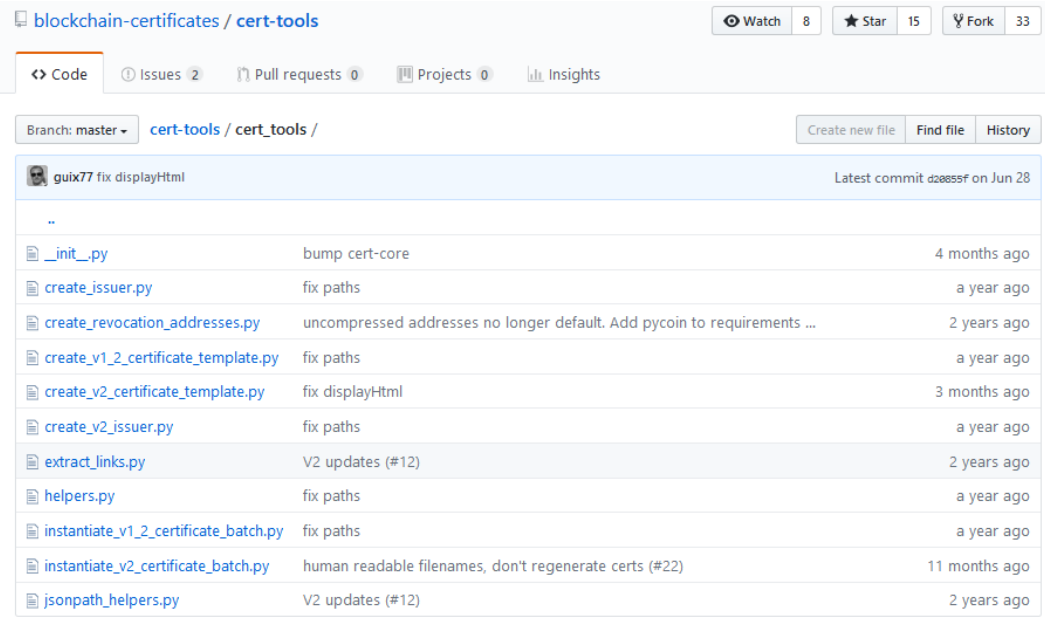
The Bitcoin blockchain itself acts as the distributed database that enforces trust and saves authentication data for thousands of certificates. The local database manages the actual JSON-based certificates. The Bitcoin network is divided into three networks. The main network where coins are traded as real currency, a test network called Testnet where coins with no associated value are traded and a regression network where coins are created instantly and are never to be traded.

The regtest mode only works with a local bitcoin node and acts as a private blockchain with the same basic rules as testnet. It allows complete control over the environment including when to create new blocks. At the time of this report, the Bitcoin Testnet is flooded with all possible inputs locked in unconfirmed transactions. Thus this study will rely on the regtest mode.

Implementation

To issue certificates a regtest-preconfigured bitcoin node is installed in a Docker container. Then an issuing address is created using the command bitcoin-cli getnewaddress. In regtest mode, it is possible to print fake bitcoin. These are then sent to the issuing address. The commands used for this are bitcoin-cli generate 101 and bitcoin-cli sendtoaddress $issuer 5 respectively. Thereafter the certificates are issued to the blockchain using the cert-issuer module.

Figure 3 shows a sample of the code that was used to implement the cert-tools module. It is based on the Python programming language. The python code in the cert-tools module below is responsible for creating the certificate template based on a list of recipients input from a comma-separated values file. The cert-issuer module combines the issuer private key and a bit of cryptocurrency to issue certificates associated with the blockchain.

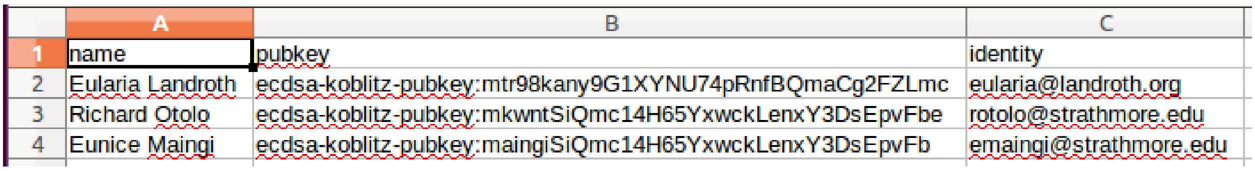


guix77 fix displayHtml — Latest commit d28855f on Jun 28

..		
__init__.py	bump cert-core	4 months ago
create_issuer.py	fix paths	a year ago
create_revocation_addresses.py	uncompressed addresses no longer default. Add pycoin to requirements …	2 years ago
create_v1_2_certificate_template.py	fix paths	a year ago
create_v2_certificate_template.py	fix displayHtml	3 months ago
create_v2_issuer.py	fix paths	a year ago
extract_links.py	V2 updates (#12)	2 years ago
helpers.py	fix paths	a year ago
instantiate_v1_2_certificate_batch.py	fix paths	a year ago
instantiate_v2_certificate_batch.py	human readable filenames, don't regenerate certs (#22)	11 months ago
jsonpath_helpers.py	V2 updates (#12)	2 years ago

Figure 7 Project Code (“Blockcerts,” n.d.-a)

The cert-tools module was used to instantiate a batch of unsigned certificates. This involved putting the recipients in a CSV file containing the student names, bitcoin address and their email as shown in figure 4.



	A	B	C
1	name	pubkey	identity
2	Eularia Landroth	ecdsa-koblitz-pubkey:mtr98kany9G1XYNU74pRnfBQmaCg2FZLmc	eularia@landroth.org
3	Richard Otolo	ecdsa-koblitz-pubkey:mkwntSiQmc14H65YxwckLenxY3DsEpvFbe	rotolo@strathmore.edu
4	Eunice Maingi	ecdsa-koblitz-pubkey:maingiSiQmc14H65YxwckLenxY3DsEpvFb	emaingi@strathmore.edu

Figure 8 Recipient CSV file

It is more efficient to issue certificates as a batch instead of one certificate per Bitcoin transaction. This requires that a Merkle tree of certificate hashes be built and the Merkle root, which is a 256-bit hash is registered as the OP_RETURN field in the Bitcoin transaction and is stored on the Bitcoin blockchain.

The blockchain certificate verification process is carried out by the cert-verifier module. The cert-verifier module makes use of the verifier.py program that takes a blockchain certificate as its input. The verification process consists of several checks as follows:

- Verifying that the actual blockchain transaction used to issue the certificates exists.

- Cross-checking the public keys claimed by the issuer are the same as the ones on the blockchain and that they were valid at the time of issuance.
- Verifying that the certificate has not been revoked,
- Verifying certificate integrity by validating the Merkle proof in the certificate. This is done by comparing the Merkle root in the certificate and on the blockchain,
- Comparing the hash of the local certificate and that on the blockchain.

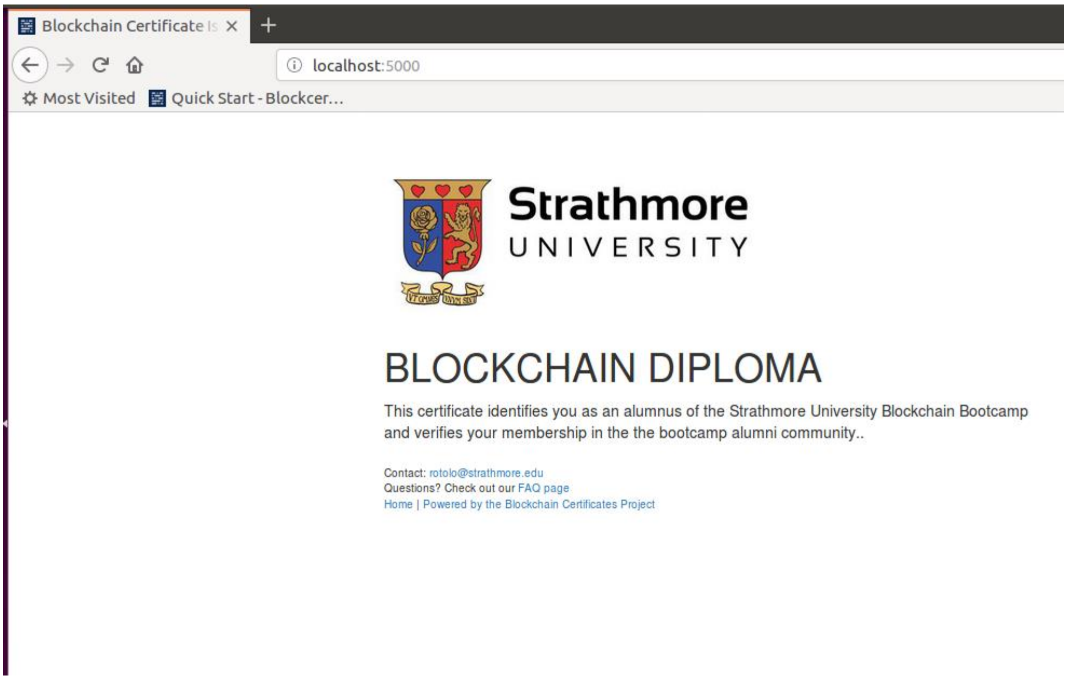
Ethical Considerations

The respondents in this study were required to participate in this study based on their own free will. Respondents personal data was kept private and only used for analysis purposes.

Results

The issuing public address, i.e. the address from which the blockchain certificates will be issued was generated using the bitcoin-cli getnewaddress command and it resulted in the following issuing address for the host institution mtXWDB6k5yC5v7TcwKZHB89SUp85yCKshy

In order to view the blockchain certificates that have been issued, the cert-viewer module is required. It has a cert-data folder into which the certificates are copied into. The cert-viewer module then makes use of a Flask web application to display and verify the blockchain certificates after they have been issued. Figure 5 shows an example of a blockchain certificate that was issued onto a blockchain and viewable on the local host on port 5000.



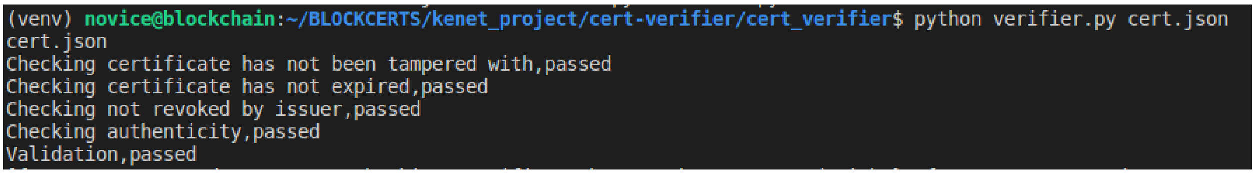
BLOCKCHAIN DIPLOMA

This certificate identifies you as an alumnus of the Strathmore University Blockchain Bootcamp and verifies your membership in the the bootcamp alumni community..

Contact: rotolo@strathmore.edu
Questions? Check out our FAQ page
Home | Powered by the Blockchain Certificates Project

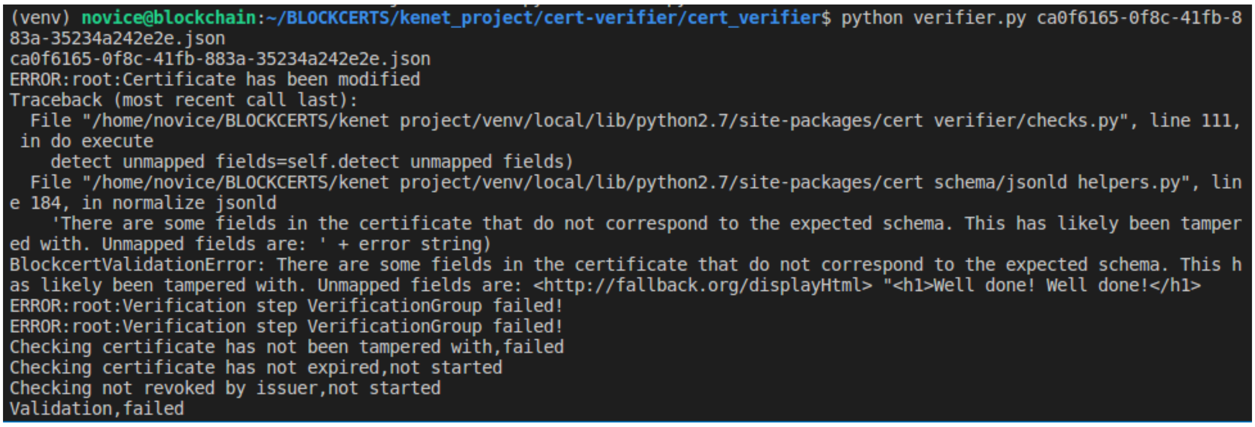
Figure 9 Blockchain certificate issued in Bitcoin regtest mode

The cert-verifier module code was used to check the validity of the blockchain certificates. Figure 6 and Figure 7 show the output for a legitimate and fake certificate after being run through the module.



```
(venv) novice@blockchain:~/BLOCKCERTS/kenet_project/cert-verifier/cert_verifier$ python verifier.py cert.json
cert.json
Checking certificate has not been tampered with,passed
Checking certificate has not expired,passed
Checking not revoked by issuer,passed
Checking authenticity,passed
Validation,passed
```

Figure 10 Legitimate Blockchain Certificate Verified



```
(venv) novice@blockchain:~/BLOCKCERTS/kenet_project/cert-verifier/cert_verifier$ python verifier.py ca0f6165-0f8c-41fb-8
83a-35234a242e2e.json
ca0f6165-0f8c-41fb-883a-35234a242e2e.json
ERROR:root:Certificate has been modified
Traceback (most recent call last):
  File "/home/novice/BLOCKCERTS/kenet project/venv/local/lib/python2.7/site-packages/cert verifier/checks.py", line 111,
 in do execute
    detect unmapped fields=self.detect unmapped fields)
  File "/home/novice/BLOCKCERTS/kenet project/venv/local/lib/python2.7/site-packages/cert schema/jsonld helpers.py", lin
e 184, in normalize jsonld
    'There are some fields in the certificate that do not correspond to the expected schema. This has likely been tamper
ed with. Unmapped fields are: ' + error string)
BlockcertValidationError: There are some fields in the certificate that do not correspond to the expected schema. This h
as likely been tampered with. Unmapped fields are: <http://fallback.org/displayHtml> "<h1>Well done! Well done!</h1>
ERROR:root:Verification step VerificationGroup failed!
ERROR:root:Verification step VerificationGroup failed!
Checking certificate has not been tampered with,failed
Checking certificate has not expired,not started
Checking not revoked by issuer,not started
Validation,failed
```

Figure 11 Illegitimate Blockchain Certificate

Conclusion

In this paper, a prototype implementation of issuing educational certificates to a test blockchain was demonstrated. The prototype was able to meet the requirements as stated in the beginning of the project.

Since the project was based on the regtest network of the Bitcoin blockchain, future work will focus on issuing certificates onto the main Bitcoin blockchain. Additionally, it will be of interest to investigate the implications of issuing to other types of blockchain platforms apart from the Bitcoin blockchain.

Appendix A Blockchain Certificate Sample

Below is a sample of the Blockchain certificate that was issued on the regtest Bitcoin network.

```
  “issuedOn”: “2018-09-10T16:23:48.394568+00:00”,
  “recipientProfile”: {
    “publicKey”: “ecdsa-koblitz-pubkey:mtr98kany9G1XYNU74pRnfBQmaCg2FZLmc”,
    “type”: [
      “RecipientProfile”,
      “Extension”
    ],
    “name”: “Eularia Landroth”
  },
  “type”: “Assertion”,
  “badge”: {
    “description”: “Completion of Certifiaction in Blockchain Technology”,
    “signatureLines”: [
      {
        “jobTitle”: “University Issuer”,
        “image”: “data:image/png;base64,ggg==“,
        “type”: [
          “SignatureLine”,
          “Extension”
        ],
        “name”: “Your signature”
      }
    ],
    “image”: “data:image/png;base64,...Jggg==“,
    “name”: “Certificate of Accomplishment”,
    “criteria”: {
      “narrative”: “pass the exam”
    },
    “type”: “BadgeClass”,
  “id”: “urn:uuid:82a4c9f2-3588-457b-80ea-da695571b8fc”,
  “issuer”: {
    “name”: “Strathmore University”,
    “url”: “https://www.strathmore.edu”,
    “image”: “data:image/png;base64,iVBORw0KGgoAAA ....”,
    “id”: “https://www.blockcerts.org/samples/2.0/issuer-testnet.json”,
    “revocationList”: “https://www.blockcerts.org/samples/2.0/revocation-list-testnet.json”,
    “type”: “Profile”,
    “email”: “info@strathmore.edu”
  }
},
“verification”: {
  “publicKey”: “ecdsa-koblitz-pubkey:msBCHdwaQ7N2ypBYupkp6uNxtr9Pg76imj”,
  “type”: [
    “MerkleProofVerification2017”,
    “Extension”
  ]
},
“@context”: [
  “https://w3id.org/openbadges/v2”,
  “https://w3id.org/blockcerts/v2”
],
“displayHtml”: “<h1>Well done! Well done!</h1>“,
“recipient”: {
  “type”: “email”,
  “hashed”: false,
  “identity”: “eularia@landroth.org”
},
“id”: “urn:uuid:ca0f6165-0f8c-41fb-883a-35234a242e2e”,
“signature”: {
  “type”: [
    “MerkleProof2017”,
      “Extension”
    ],
    “merkleRoot”: “033ddbd052d13d492b5f5691f4a3bbe7e160d6eba960ed9d398f054f65c49bb8”,
    “targetHash”: “671301e0428d98b144f9f099341acf98320f863fa297e09c269b3aaf4bbc4bcc”,
    “proof”: [
      {
        “left”: “f78f56b0abf872838b56f5ca7e9d440eff56322a7793b446d7b2763dbbdc55fb”
      }
    ],
    “anchors”: [
      {
        “sourceId”: “863424b01d98228821cf32d18288ec818f25c3e84e541d7bad5c32e6fbbdbe11”,
        “type”: “BTCOpReturn”,
        “chain”: “bitcoinRegtest”
      }
    ]
  }
}
```

References

1. Blockcerts. (n.d.). Blockchain Credentials. Retrieved June 22, 2018, from http://blockcerts.org/
2. Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, *85*(6), 1213–1221. https://doi.org/10.1016/j.jss.2012.02.033
3. Grech, A., Camilleri, A., & Inamorato, A. (2017, November 2). Blockchain in Education - EU Science Hub - European Commission. Retrieved June 2, 2018, from https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/blockchain-education
4. MIT Media Lab. (2016, June 3). What we learned from designing an academic certificates system on the blockchain. Retrieved May 31, 2018, from https://medium.com/mit-media-lab/what-we-learned-from-designing-an-academic-certificates-system-on-the-blockchain-34ba5874f196
5. Office of the Registrar. (n.d.). Transcripts: MIT Registrar’s Office. Retrieved June 3, 2018, from http://web.mit.edu/registrar/records/transcripts/official.html