UNIVERSITY OF BIRMINGHAM

Blockchain based Academic Certificate Authentication System Overview

1. Purpose

Counterfeit academic certificates have been a longstanding issue in the academic community. Not until the Massachusetts Institute of Technology Media Lab released their project of Block-certs, a technique which is mainly implemented by conflating the hash value of local files to the blockchain but remains numerous issues, did an effective technological approach protecting authentic credential certification and reputation appear.

Based on Blockcerts, a series of cryptographic solutions are proposed to resolve the issues above, including, utilizing a multi-signature scheme to ameliorate the authentication of certificates; exerting a safe revocation mechanism to improve the reliability of certificates revocation; establishing a secure federated identification to confirm the identity of the issuing institution.

2. Overview

The project consists in designing and implementing the system which covered the above solutions. The project also involves a comprehensive evaluation of the system security, and the assessment outcomes provide compelling evidence to prove that implementation is practical, reliable, secured, which might give some hints of important architectural considerations about the security attributes of other blockchain-based systems.

In this section, we discuss the implementation from the point of view of system architecture, database architecture. The system architecture and database architecture show how the system is designed from the engineering point of view.

2.1 System Architecture Overview

The system briefly consists four components in our implementation: verification application including federated identity, issuing application involving multi-signature and BTC-address based revocation, Blockchain and local Database adopted by MongoDB.

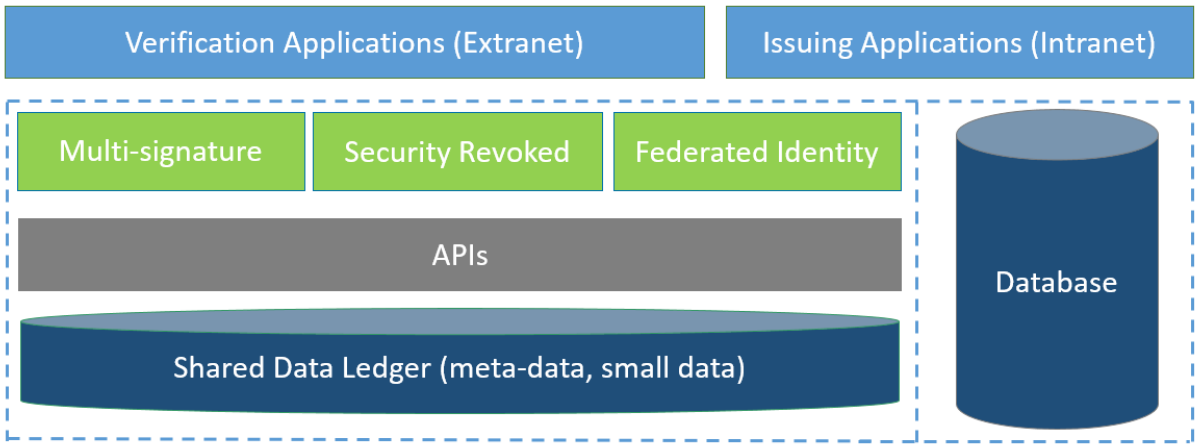

Fig1: System Architecture Context Diagram

Rujia Li, Yifan Wu, IT Innovation Interns
yxw689@bham.ac.uk, rxl635@bham.ac.uk

The issuing applications are responsible for the main business logic which include the certificates applying, examining, signing and issuing. The issuing applications are designed to merge the hash of the certificate in a Merkle tree and send the Merkle root to Blockchain amidst signing by the majority of community members. Also, the issuing applications involved the revocation of certificate. The issuing applications are responsible for the main business logic which includes the applying for, examining, signing and issuing of the certificates. The issuing applications are designed to merge the hash of the certificate with a Merkle tree and send the Merkle root to the Blockchain. Also, the issuing applications deal with the revocations of certificates.

The verification application focuses on checking the authenticity and integrity of the certificates that have been issued. It includes two main components: a web-based page and an Android-based application. They use the same mechanism, and fetch the transaction message through the blockchain API and compare the transaction message with the verification data from the receipt. The mechanism can be briefly described in the following way: check the authentication code is valid; check the hash with the local certificate; confirm the hash is in the Merkle tree; ensure the Merkle root is in the blockchain; verify the certificate has not been revoked; validate the expired date of the certificate. Also, it has to be mentioned that for the convenience of sharing the certificates, the Android-based application allows for verification of the documents by scanning the QR code directly.

The blockchain acts as the infrastructure of trust and a distributed database for saving the authentication data. Typically, the authentication data consist of the Merkle root generated using hashed data from thousands of certificates. The MongoDB is employed as our database since the MongoDB successfully manages JSON-based certificates and provides high availability and scalability.

2.2 Database Architecture Overview

The database has been designed to contain two categories of data: the public authentication data and the private certificate data. The public authentication data is available to the public and released to the blockchain; the private certificate data are stored in the MongoDB where it is securely protected and isolated in the intranet.

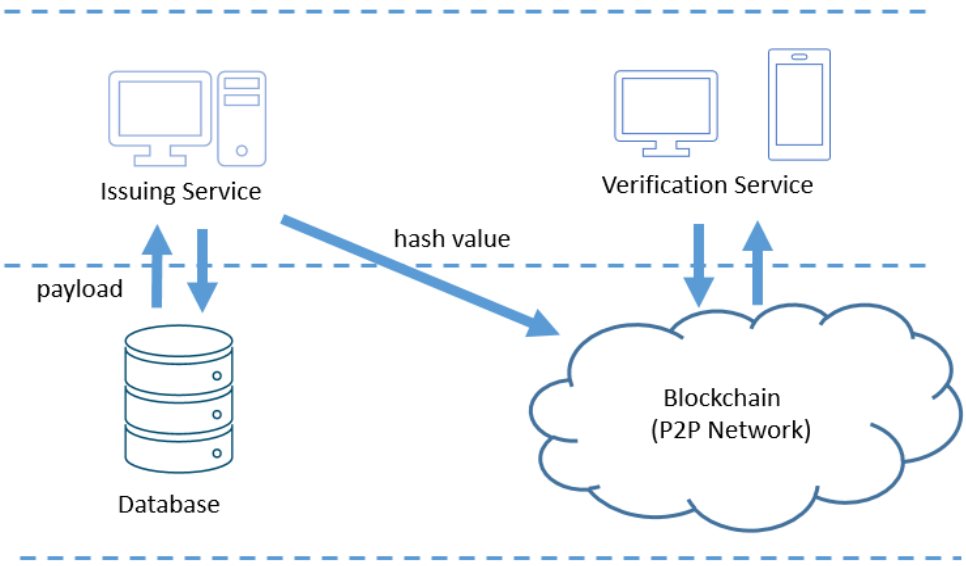

Figure 2. Database architecture.

Figure 2 maps out the top level data flow diagram. It shows that the data flow is unidirectional from the internal areas to the internet. The issuing system reads the certificate from the MongoDB and broadcasts its "point" data to the blockchain. The verification service only needs access to the blockchain to check the authenticity of the certificate.

At the same time, the enterprise MongoDB architecture has been adopted as our local database as shown in Figure 3. In this architecture, the "mongo server" serves as the router to access the primary service, the "configure server" keeps the system working meta data and the "mongo server" saves the core data.

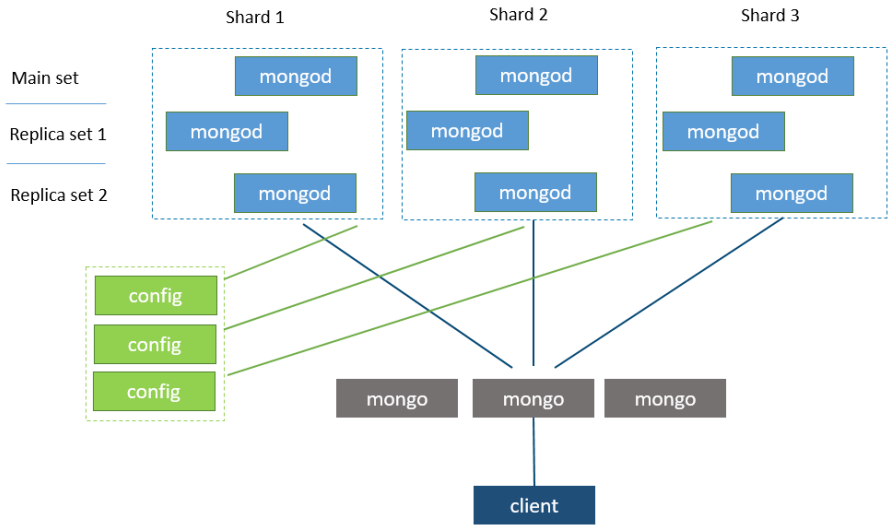

Figure 3. MongoDB Production Cluster Architecture.

3. Functions

3.1 Verification Applications

The verification applications are responsible for checking the authenticity and integrity of the certificates that issued before. It mainly includes two components: the web-based application and the client-based application. The verification applications fetch the transaction and get the verification information through the blockchain API, then the system authenticates the validation of the verification information by comparing with the checking information of the receipt.

The main component functions can be described as follows:

- Upload the PDF files / Scan the QR code
- Calculate the hash value for the PDF file
- The client makes a request with the blockchain
    - The interaction with blockchain API
- The logic of the verification
    - Authentication management: the issuing address relationship with the school identity.
    - The verification of hash value on the certificate (to avoid tampering)
    - The verification to confirm if the hash value is in the merkle tree

    - The verification to confirm if the hash value of the merkle tree root is on the blockchain
    - The verification of the validity of the certificate (to avoid the revoked certificate)
    - The verification of the valid date of the certificate (to avoid the expired certificate)

3.2 Issuing Applications

The issuing applications are responsible for the main business logic, which include the certificates, applying, reviewing, turning over and issuing. It merged the certificates hash in an merkle tree and send the merkle root to blockchain by APIs.

The main component functions can be described as follows:

- Login function
    - The security of login
    - Reset the forgotten password(option)
- Privilege control
    - User role with different privilege
    - Different pages when changing to different user role
- The approval process (student->>checker->>supervisor->>administration staff->>head of school)
- Multi-signature function
- Auditing the certificate
    - View the published certificate
    - View the signed certificate
    - View the certificate ready to sign
- Revoking the certificate
    - For one certificate
    - For batch certificates
- Switch different environment (runtime environment/testing environment)
- Administration page to manage the data, the privilege and more.
- Cold storage for the keys (will release in next version)

4. Implementation

4.1 Prototype workflow

To implement the above design and analysis, we created the prototype model workflow for four main roles, including student, checker, issuer, system and employer. The prototype workflow is shown in the figure below.

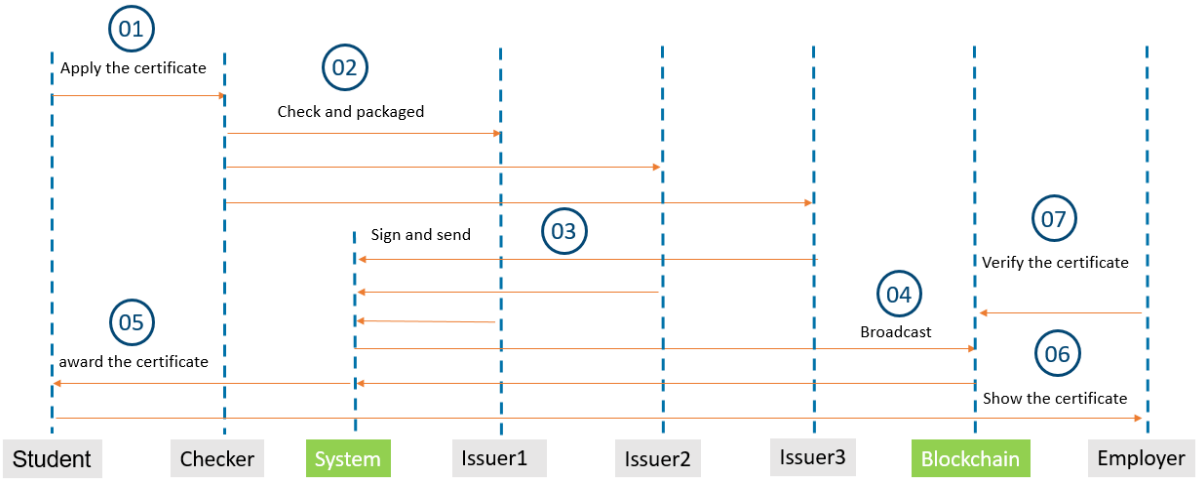

Figure 5. Workflow of prototype.

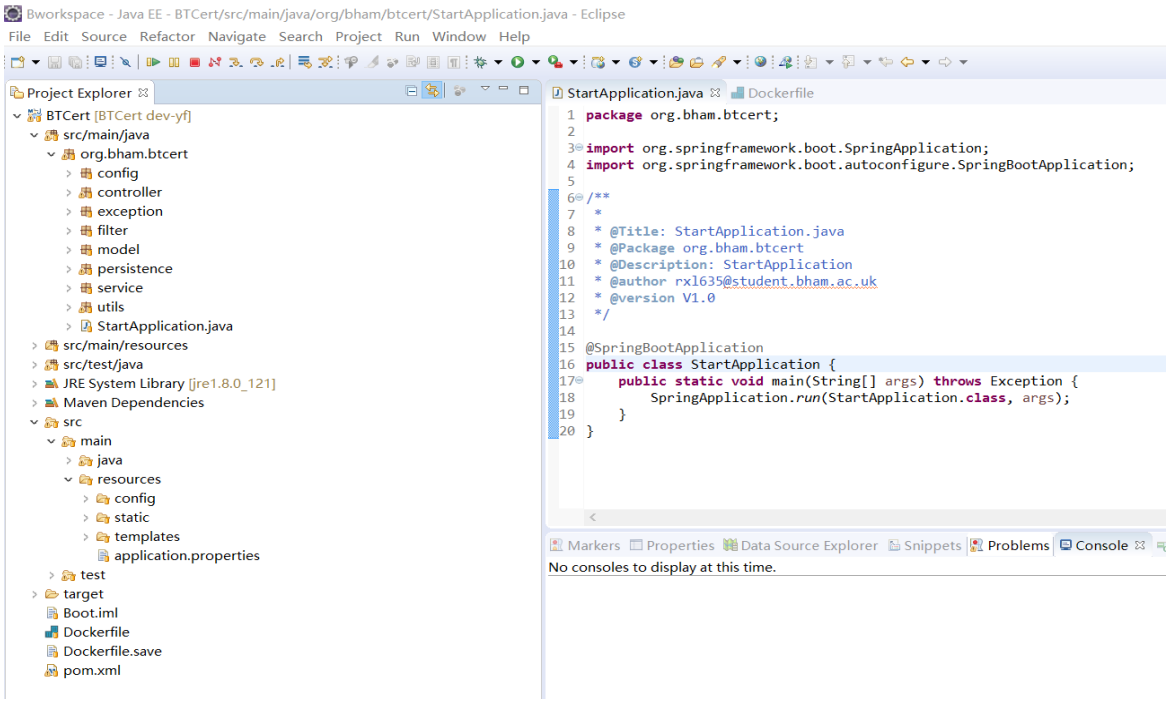

Figure 6. Project code structure.

Specifically, the prototype workflow is as follows: Firstly, the student applies to the school for a credential, and the certifiers check the students' information and merge the credential with a Bitcoin transaction once it is approved. Then the majority of the academic committee members sign it with their private keys. After that, the system broadcasts the transaction which contains the Merkle root for all the certificates. Following the above step, the student receives a JSON-based certificate once the transaction is confirmed by the miners. In the next stage, the student provides the JSON-based certificate to the employer, when he or she applies for a job. Lastly, the company verifies the certificate via access to the Blockchain and checks the authentication code.

4.1 Project implementation

Figure 6 shown us the code structure in our project, which contains the Java code, configure code and the JavaScript code. Among them, Java code is in charge of back-end business on the server. This business includes applying, checking and interacting with the local database. The configure code such as docker file is to config the global setting. The JavaScript code is used to implement multi-signature algorithm and make web pages interactive, also JavaScript code is designed to call the APIs for the blockchain. Lastly, it has to mentioned that all the elements in Figure 6 is our own original code except for the "Maven Dependencies" and the "JRE System library". The Java code consists of the config, controller, exception, model ,service, utils and filter modules. We describe the java code functionality as follows in Table 1.

Table 1. Java package functionality description.

Module	Description
org.bham.btcert.config	Defines the configuration management for access control and security provider.
org.bham.btcert.controller	Defines APIs for the WebView, contains Spring's model-view-controller (MVC) and REST Web Services implementation for web applications
org.bham.btcert.exception	Defines the implementation of exception
org.bham.btcert.filter	Provides the implementation of http filter serve as preventing xss attacks
org.bham.btcert.model	Defines Object Model
org.bham.btcert.persistence	Provides the implementation of BaseMongoTemplate used for connecting the MongoDB
org.bham.btcert.service	Provides the implementation of the service used by org.bham.btcert.controller
org.bham.btcert.utils	Defines utility classes incudling Merkle tree, CHexConver,CryptoUtil

The configuration data to define the running environment, the server, and other devices setting that compose the system and its boundary. Mainly, our system concludes three configuration files which are outlined in Table 2.

Table 2. Configuration data description

Configure File	Description
pom.xml	POM stands for "Project Object Model". It is an XML representation of a Maven project
Dockerfile	A text file that contains all the commands, in order, needed to build a given image in our project
application.yml	Spring Boot configure file to specify the properties

5. Evaluation

In this section, we focus on the protocol security which proposed in the section of "Cryptographic protocol design". To set up a secured configuration and of multi-signature protocol, we did two

experiments to verify the two parameters' effects on signing progress. Also, we evaluated the revocation mechanism from the aspect of reliability, security, cost .etc.

5.1 Multi-signature secure evaluation

The multi-signature is the mixture of all the necessary public keys and required conditions, specifically, an M | N address, where N is the total number of public keys and M is the minimum number of private keys required for validation. Note that we used payment successfully to represent the issuer certificates in all three experiments.

In experiment I, the parameter M is a fixed number and N is set to be a varied number. we created five groups of combined addresses that joined different numbers (3,5,7,9,11) of original ECDSA public keys. These combined addresses are represented as a0, a1, a2, a3, a4 and all required two private keys to validate when redeeming the coins. Then we transferred some coins to these addresses and used these addresses to assemble raw transaction strings by attaching the Merkle root. After that, we used two private keys to sign every raw transaction string while broadcasting them to make a payment to an appointed address (using appointed address for recycling these coins). We found that all the transactions satisfied the corresponding conditions that have successfully been accepted by the blockchain. (Results are demonstrated in Table 5)

Table 5 Result of experiment I

Addresses	Required Number M	Total number N	Executed results
a0	2	3	Accepted
a1	2	5	Accepted
a2	2	7	Accepted
a3	2	9	Accepted
a4	2	11	Accepted

Note: "Accepted" means the transaction is accepted by the blockchain. "Rejected" means the transaction is rejected by the blockchain.

In the experiment II, the parameter N is a fixed number 7 and M is set to be a varied number that ranged from number 1 to 7, we applied the same steps as experiment I and providing two private keys to sign the raw transaction strings. The result is shown in Table 15.Table 6. Result of experiment II

Addresses	Required Number M	Total number N	Executed results
a0	1	7	Accepted
a1	2	7	Accepted
a2	3	7	Rejected
a3	5	7	Rejected
a4	6	7	Rejected

Note: "Accepted" means the transaction is accepted by the blockchain. "Rejected" means the transaction is rejected by the blockchain.

The results of these two experiments suggest that whatever the number of N, as long as it has two signatures, it will be accepted by the blockchain. It did not end up being as positive as we had hoped, which was that the majority of the key owners would be needed to sign the raw transaction strings before making a payment. In other words, we wanted the blockchain to accept the raw transaction string once it met the required number of N.

In a word, our results indicate that the designers should consider the democratic conditions as well as the secure conditions when setting up the configuration. Setting the signature threshold with M > N/2 + 1 is compulsory to achieve security against an adversary that might corrupt any minority of the M privacy peers.

5.2 Revocation mechanism evaluation

We conduct the evaluations form the aspect of reliability, security, applicability and cost. For easy to compare the mechanism, we use the abbreviation to represent the approach, precisely, the abbreviation of “BV1”, “BV2”, “OP” represent the method utilized in the version of 1.0 and 2.0 of Blockcerts and our project respectively.

When it comes to the reliability, BV1 and OP almost have the same performance but higher than BV2, since BV1 and OP are all based on the BTC transaction state, and the BV2 adopted the certificate revocation list based on the fixed URL, which checking the BTC transaction state is stable than the querying the certificate via URL.

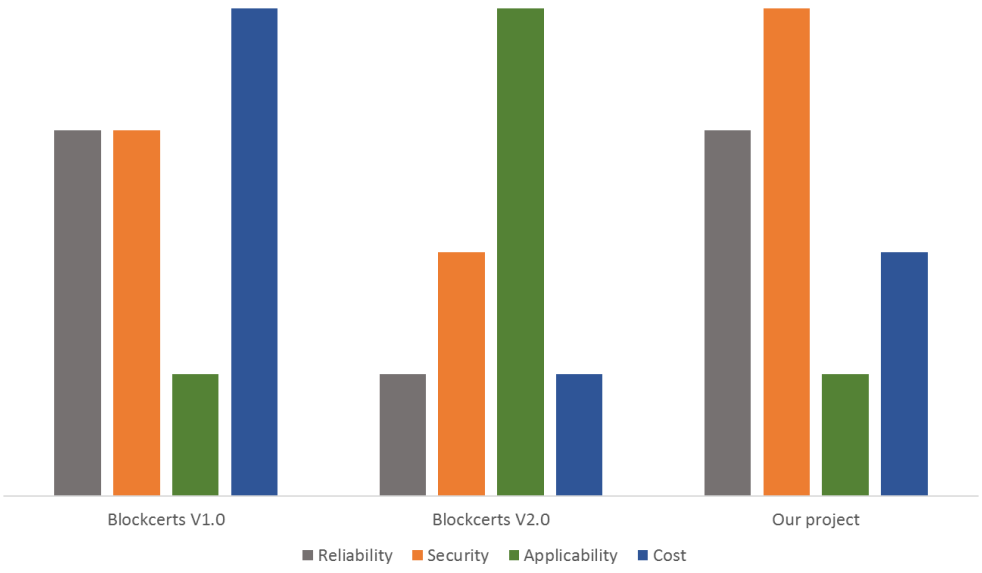

Figure 7. Revocation protocol comparison from four perspectives.

Regarding the security property, OP ranked the first position for the reason that checking transaction state in the blockchain is stable all the revoked address belongs one wallet . Followed by BV1, As the certificate state query service cannot commit working persistently or out of being hijacked. The fact that BV2 has the most security problems is due to storing the all private keys for the awarding body.

Admittedly, BV2 have been widely accepted by the public. Hence it got the highest applicability among all the approaches, while BV1 and OP were reckoned to be less applicable owing to no institution adopting the blockchain-based approach.

In term of the expenditure, BV1 is most costly arisen by each revocation need at least two payments, while BV2 is proved to be the most economical result from it do not need to spend the bitcoin, OP is estimated to be medium because it only needs to experience one transaction for each revocation.

In conclusion. OP succeed in remedying the defect in BV1 and BV2 to a certain extent and become more secured than the other two.

5.2 Summary

We conducted extensive evaluations from the operations security, data security, network security and the protocol security. We outlined all the assessment in Table 4. The evaluation result indicates us that our system is secure enough to meet the production requirements. Table 7. Summaries of security evaluation outcomes.

Safety assessment index	Index category	Index details	Performance
Operations security	Operations security	User unauthorized access.	++++
		Certificate issued and revoked inappropriately	+++
		Leaking the login password	++++
		Leaking the private key and password	++
Data security	Certifcate data	Data confidentiality	++++
		Data integrity & tamper-proofing	+++++
		Data access security	++++
		Data availability	++++
	Blockchain data	Data confidentiality	+++++
		Data integrity & tamper-proofing	+++++
		Data access security	+++++
		Data availability	+++++
Network security	Network security		++++
protocol security	Multi-signature		++++
	Revocation mechanism		++++
Summary			++++

6. SWOT Analysis

SWOT analysis is an acronym for strengths, weaknesses, opportunities, and threats and it is a structured planning method that evaluates those four elements of an organization, project or business venture. Based on the features of BTCert, a SWOT analysis was carried out which is shown as follow:

STRENGTHS	**OPPORTUNITIES**
➢ Proposed an innovative BTC address based solution to revoke a certificate which is more reliable. ➢ Utilizing multi-signature rather than the single private key makes the academic certificates issuing progress more secure. ➢ The authentication data of the credential which published to blockchain is immutable, trustful and verifiable. ➢ The new approach of authenticating the certificate(scan the QR code) simplified the workflow to efficient and economical. ➢ The core data of the credential is secure and private even the blockchain technology crashes in the future.	➢ The Blockchain technology has evolved and has given various opportunities for other industries such as banking and finance, manufacturing, healthcare. Blockchain has the potential to transform education industries(academic certificate as an example) and make processes more efficient, transparent democratic and secure. ➢ The Blockchain is intended to help us create the digital relationship that will reshape the world of business and transform the old order of human affairs for the better. ➢ At the university, the blockchain technology can not only be used at the ceritificate authenticating, but also be used at the statement of official documents or files and other areas.
WEAKNESSES	**THREATS**
➢ In the phase of the multi-signature, the member of the academic committee member needs to remember the private key. However, the private key is a format of some irregular hexadecimal characters which is hard to remember. ➢ In the phase of broadcasting the certificate authentication data to the blockchain, the university should pay a few mining fees for the minner to confirm it on the blockchain. ➢ The broadcasting API using in our project has a potential that it is not available in the future.	➢ Nowadays, the applications related to the blockchain technology are still in the experimental phase. ➢ The blockchain technology is not widely accepted by the public now since most of the people trust 3rd organization. ➢ Our project is based on the Bitcoin blockchain, the maintenance of which relies on thousands of participants in the cryptocurrency ecosystem. Admittedly, it is imprudent to assume that the Bitcoin would work well continuously in the future because multiple types of stakeholders influence the blockchain ecosystem or business model.

7. Conclusions and Recommendations

In June 2016, the MIT media lab released their blockchain-based credential system which is more secure, more reliable and harder to forge, in contrast to existing technologies that based on the third party arbitration. However, there are some serious authentication defects and vulnerable revocation mechanism which limits the prevalence and application of the project. In our project, to solve these problems and make its concept more practical, we proposed and designed a set of innovative cryptographic protocols which includes multi-signature, BTC- address-state-based revocation mechanism and trusted federated identity.

Among these protocols, the multi-signature scheme most notably increases the difficulty of forging owing to the fact that each issuing progress is obliged to be signed by the majority of the academic committee members. Besides, it enhances the safety of the private keys storing for the reasons that the private keys are possessed by separated devices and people. Besides, BTC-address-based revocation mechanism improved the stability of the certificate revocation because BTC address is accessible and stable at any time. Moreover, this approach reduced the failure probability of revocation, because the cancellation process adheres the same the multi-signature algorithm, alike, involving several people. Trusted federated identity innovatively proved the authenticity of the certificate through the trusted path and federated identity. What's more, the protocol of our project can be used in other related realms such as digital right protecting and contract proof. Case in point, our protocol enables the two companies to attach their contract onto the block chain with multi-signature, which is different from the traditional third party-based work mode and dispel the worries of forging credentials.

Moreover, we implemented a blockchain-based certificate system, which embraced all the above protocols, by utilizing Java and JavaScript. This system has remedied the defect in Blockcerts to a certain extent, which makes the theory of blockchain-based certificate more practicable. Eventually, we conducted a series of security assessment from the perspective of operational safety, data security, network security and protocol security. The assessment outcomes provide compelling evidence that system is secured enough to meet the enterprise application standards.

Lastly, there are some limitations remained to be discussed, albeit, these considerations fall outside the scope of this paper: Our project is based on the Bitcoin blockchain, the maintenance of which relies on thousands of participants in the cryptocurrency ecosystem. Admittedly, it is imprudent to assume that the Bitcoin would work well continuously in the future because myriad types of stakeholders influence blockchain ecosystem or business model. In the years to come, we will adopt multiple blockchain sources such as Hyperledger and Ethereum to eliminate the factors of instability.

8. Appendix I. Evaluation Matrix Scores

Area	Scoring System	Score	Reason
Maturity	1 = Idea 5 = Mainstream Product	2	Innovative prototype, currently only has a few examples using block chain for authenticate certificates
Technology (Adoption timescales)	1 = > 3 years 5 = < 3 months	4	Further development and testing of the prototype will be necessary;
Business Process (Adoption timescales)	1 = > 3 years 5 = < 3 months	4	Need to consider the user acceptance of the Blockchain technology, and some resources will be needed for supporting the system;
Adoption Overview	1 = v long time 5 = very short	4	Consider the technology and business process, overall adoption will be in a short tine scale if we want to use the system.
Existing Technology (Impact)	1 = v large impact 5 = very little	4	There will be some impact to the current student certificate authentication system.
Resources Required	1 = v large impact 5 = very little	3	Resources will required for software development, testing, training, and support.
Scope	1=very difficult 5=very easy	3	The system can also be used in securing contracts, personal information, although it has the limitation in editing content.
Usability	1=very difficult 5=very easy	4	It is easy to use for the user, employing company and the university.
Security	1 = very poor 5 = excellent	5	The system has high level of security measures
Innovation Value	1 = low innov. 5 = high innov.	5	If the system is adopted,the university could be the pioneer in using blockchain to authenticate students' certification
Cost Effectiveness	1=very expensive 5=very cost effective	5	Provide the development and testing is in-house, and later it will need some support resources, it is very cost ffective.
Adoption Readiness Score	*<20 - not ready* *20-29 - emerging* *30-39 - Adoptable* *>39 Fully Ready*	*35*	*Using blockchain to authenticate students' certificate is innovative and effective. The process simplified the workflow to be more efficient and economical. The prototype is ready for testing and adoption.*
***Note**: Rows that have no highlight colour indicate the score value is not added to the adoption readiness total. Instead, the overview score for that area is used as part of the total score.*			

Appendix II. Project screenshots

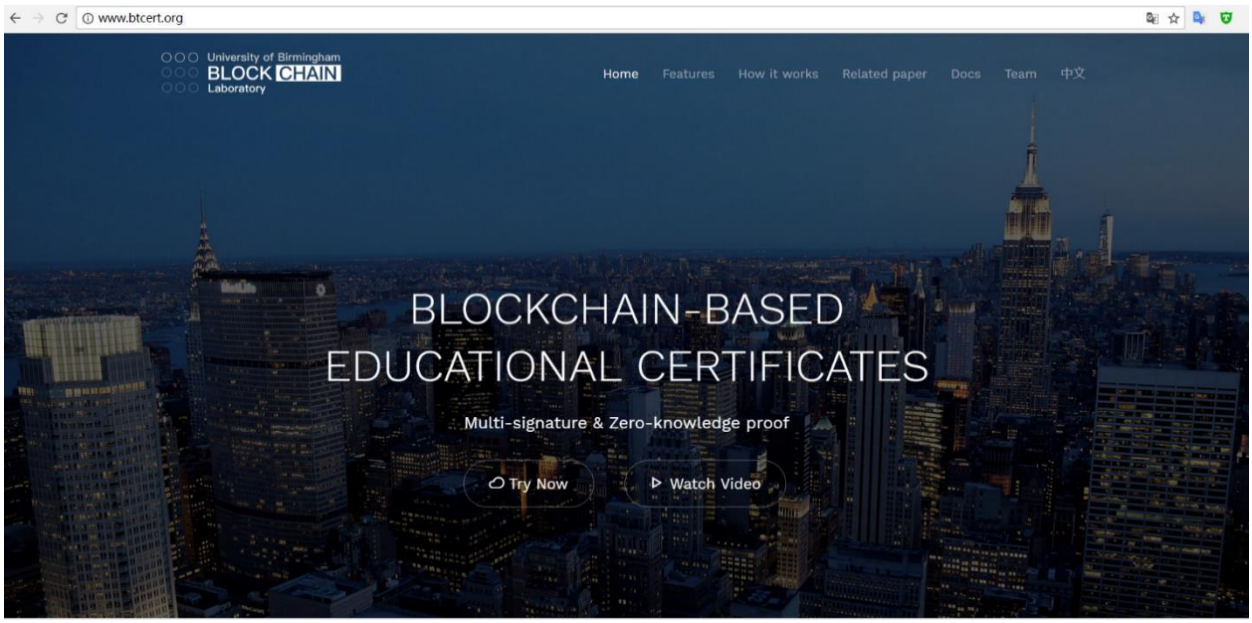

Figure 8. Public home page.

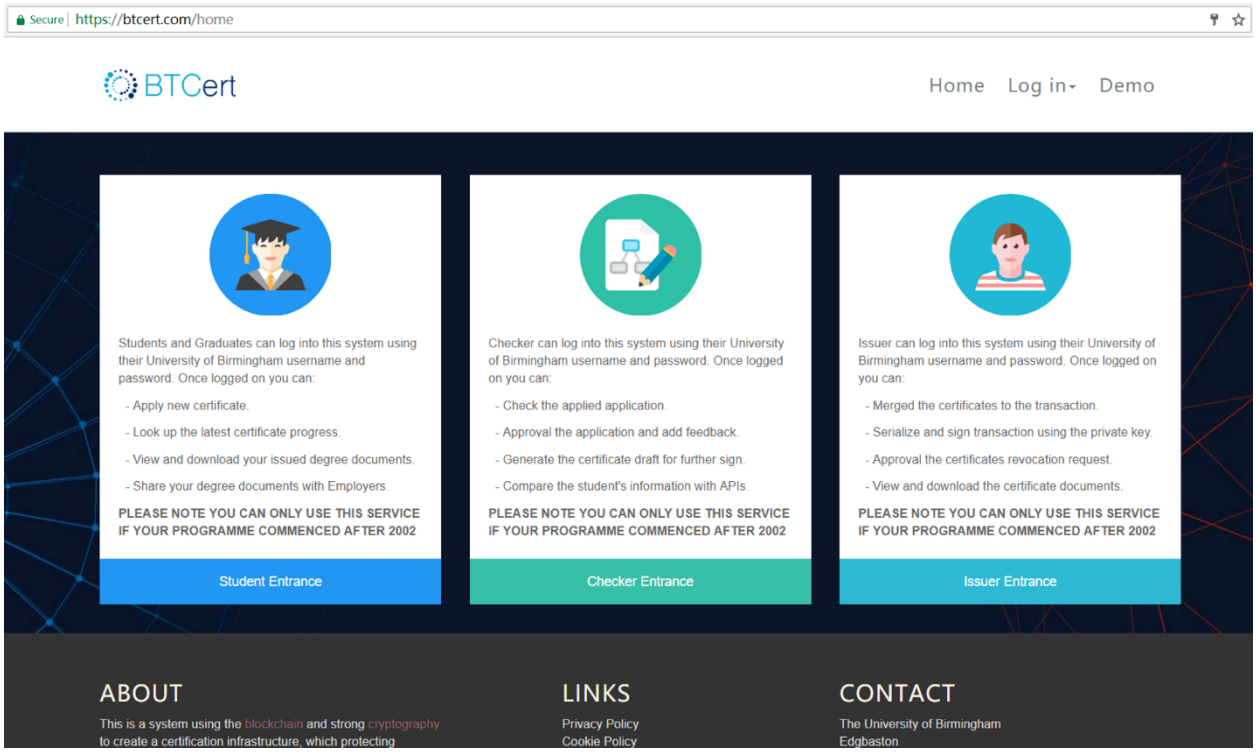

Figure 9. Intranet home page

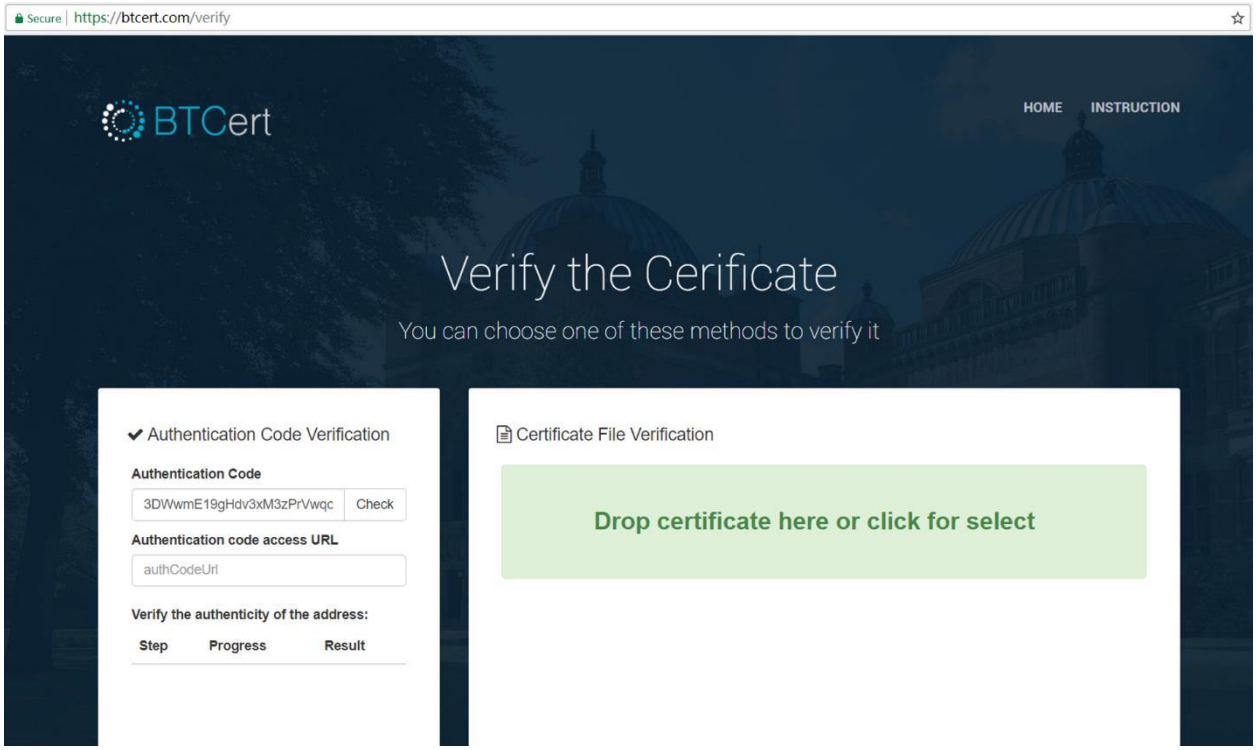

Figure 10. Public verifying page.

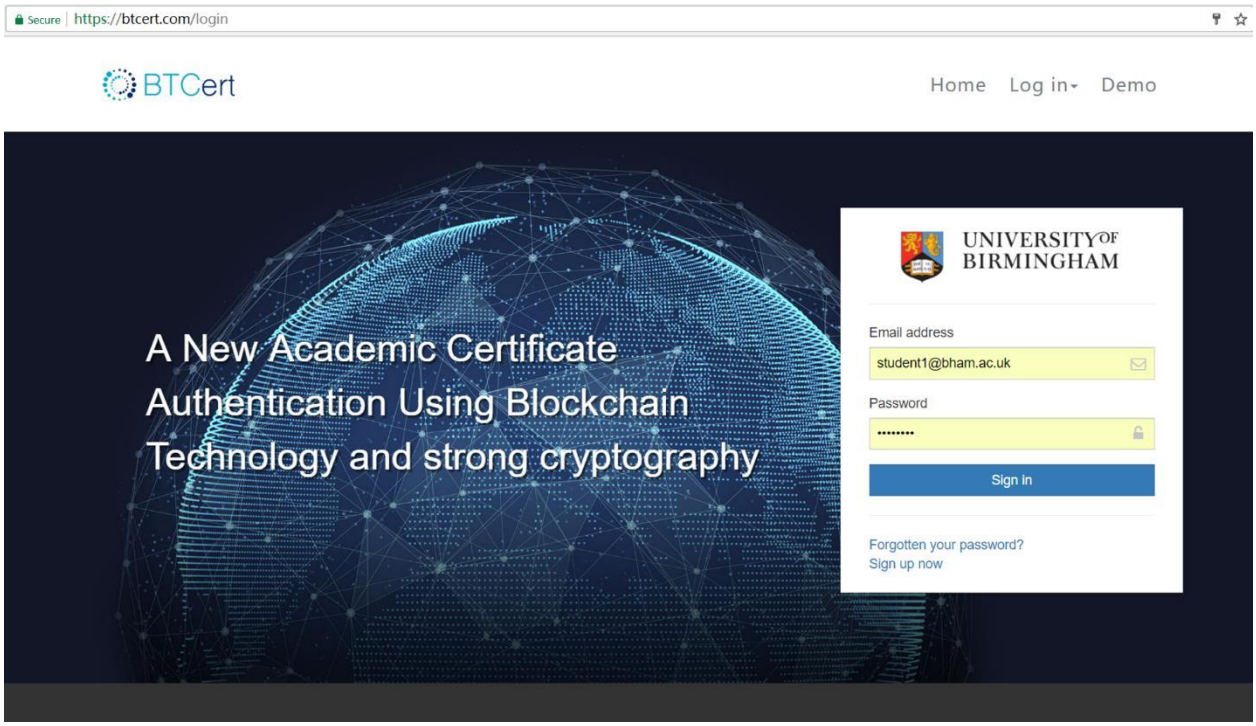

Figure 11. Intranet login page

Appendix III. Certificate example


```
{
  - badge: {
        created: "2017-01-01",
        description: "good",
        expires: "2100-01-01",
      + fileClaim: { … },
        id: https://example.org/robotics-badge.json,
      + identityClaim: { … },
        image: "good",
      - issuer: {
            email: "admin@bham.ac.uk",
            id: "862c72c7472d485b859d9c4f44bc833b",
            image: http://www.bham.ac.uk/test.png,
            name: "University of Birmingham",
            type: "Profile",
            url: http://www.bham.ac.uk
        },
        name: "Bachelor of Arts",
      + revocationClaim: { … },
        type: "Certifacte"
    },
  - context: [
        https://w3id.org/openbadges/v2,
        https://w3id.org/blockcerts/v2,
        https://blocktechcert.github.io/www/json/context.json
    ],
    id: "f8fee31efcb244719a2584548a2d17f5",
    issuedOn: "2017-08-21 14:54:26",
  - recipient: {
        hashed: "false",
        identity: "test1@bham.ac.uk",
        type: "email"
    },
  - signature: {
      + anchors: [ … ],
        context: https://w3id.org/chainpoint/v2,
        merkleRoot: "b46da6effee9ba735f4aafe2b35a847c2aeb902bcb6c9a15c745da560e5dfe18",
      + proof: [ … ],
        targetHash: "86473d316c60ddf26b4d7ec6825915bea97d9c2eb9a6790a99957dd781afe0be",
      + typelist: [ … ]
    },
    type: "badgeClass",
  - verification: {
      + type: [ … ]
    }
}
```


Figure 23. An certificate example

Appendix IV. Project instructions

Project resource:

Project home page: http://www.btcert.org/
Intranet home page: https://btcert.com/verify
Public verifying page: https://btcert.com
Intranet login page: https://btcert.com/login

Project code repository:

Main repository: https://git.cs.bham.ac.uk/BTCert/BTCert.git
Auxiliary repository: https://git-teaching.cs.bham.ac.uk/mod-msc-proj-2016/rxl635.git

Project prerequisites:

Operating system: Centos 7 x86_64
JDK: 1.8
Docker: Docker CE 17.6
Maven: Maven 3.3

Project deploy instruction:

1. Install the running environment.
2. Import the code from repository
3. Switch to directory
4. Start the project without Docker
   $ > Cd /BTCert
   $ > mvn package && java -jar target/Boot-0.0.1-SNAPSHOT.jar
   $ > nohup java -jar target/Boot-0.0.1-SNAPSHOT.jar &
5. Start the project with Docker

$ > mvn package docker:build

$ > docker images

$ > docker run image_name:tag_name