IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12417

Ethereum-Blockchain-Based technology of decentralized smart contract certificate system

N.Jaya Santhi[1], K.Lakshmi[2], K.Ayusha[3], A.Sravani[4], K.Deevena[5]

M.Tech, Asst.Professor, Computer Science & Engineering, Bapatla Women's Engineering College, Bapatla, India[1]

B.Tech,Computer Science & Engineering, Bapatla Women's Engineering College, Bapatla, India[2-5]

Abstract: Traditional paper certificates and electronic certificates have difficulties in preservation and management, not to mention other problems concerning inconvenient verification, poor reliability, anti-counterfeiting and anti-tampering. This paper proposes a scheme designed to build a decentralized certificate system that is based on blockchain technology and smart contract, in which a set of blockchain certificate system aiming at providing blockchain certificate services for college students' innovation and entrepreneur ship competition is developed. In this system, certain functions of the certificate about management, issuing, verification and revocation are realized via smart contract. Signer information, certificate template and certificate information are stored in a smart contract that adopts structured data, thereby realizing more convenient callings in querying and validating certificate.

Keywords: Blockchain, Decentralization, Data Integrity, cryptography, Transparency, Smart Contracts

I. INTRODUCTION

Certificates in earlier eras were paper made, they needed to be designed and produced by issuing agencies within the issuing process, at a time certificate holders' personal information was needed to finish issuing. After the holders obtained their certificates, which had to be stored on their own, these certificates needed to be shown when necessary. Paper certificates are so inconvenient that due limitations owed emergence to difficulties in preservation, verification, and in anti-counterfeiting. As the IT industry develops, certificates change from printed to electronic. Electronic certificates enable electronic equipment storing, thereby providing corresponding applications and interfaces to facilitate previously required issuance, query and verification, which greatly reduces inconvenience in certificate use, as well as the cost of issuance, management and preservation. However, since electronic certificates are issued by certificate agencies independently, the following factors must be taken account into. First, certificate verification is realized by the agencies that store certificate information within their data base providing applications and interfaces, under which may be attacked and/or tampered by hackers, thereby voiding certificates. Second, the data stored in the agencies' database would have voided and lose its accessibility if the agencies disappear for certain reasons. Third, more difficulties will occur when applications or interfaces are hard to be verified that whether the verification tool is provided by the agencies. Fourth, it is inconvenient and inefficient to verify certificates, because few verification tools provided are shared by various agencies, among which end users have to shift from one tool to another. On January 3, 2009, the bitcoin network was born, simultaneously generating a new technology called blockchain. The very fundamentals of blockchain are new models of combi nation of applications, in which distributed data storage, point to-point transmission, consensus mechanism, and cryptographic algorithm are embodied. Meanwhile, blockchain properties of decentralization, anti-tampering and traceability are char acterized by the above technologies. Based on decentralized blockchain, Ethereum was proposed to realize pro grammable smart contracts at the end of 2013, which makes possible building a easy-verifying, anti-tampering, anti-counter feiting and decentralized certificate system. This article proposes a decentralized system based on block chain technology, which makes possible certificate sharing while enabling thorough trust among certificate receiver, corresponding issuing authority and end users who will verify the certificate validation by adopting smart contract without any other systems.

II. BACKGROUND AND RELATED WORK

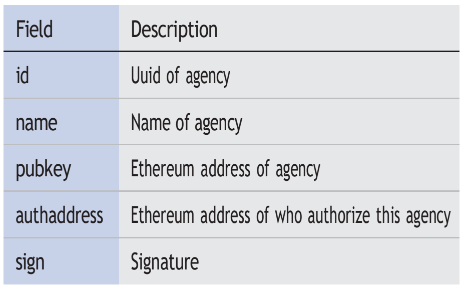
Based on the bitcoin network, the MIT Media Lab Learning Initiative, together with Learning Machine, proposed the Block Cert research project. The project performed hash operation on the certificate data that accords with the Open Badges format, which is provided by the issuing agencies. Then, by initiating the transaction, the Hash value computed based on certificate data is stored in a bitcoin transaction data together with a simple bitcoin script which is used for checking. Verification re-performed Hash operation on the certificate information provided, and the calculated value, through bitcoin script, was compared with the value that is stored in bitcoin transaction, thereby verifying the authenticity of the certificate. In 2018, the project LLP (Blockchain for Education: Lifelong Learning Passport) was proposed by Gräther, Wolfgang and some others, in which IPFS was introduced as the persistent storage of the certificate receiver's information, thereby adopting a centralized document management system to store all the certificate information, as well as using smart contract to record and verify the authenticity of the certificate. Moreover, Clemens and Brunner proposed a project named SPROOF (a platform for issuing and verifying documents in a public blockchain), using HD Wallet to enable better optimized key security, store certificate information in DHT (Distributed Hash Table), and user privacy protection.

III.IMPLEMENTATION

There are three roles in certificate systems:

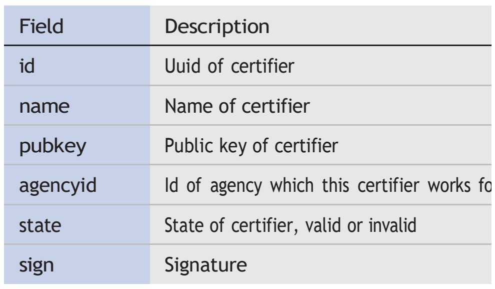
1. Roles of designing, issuing and managing certificates.
2. Roles of certificate acquisition and acceptance.
3. Roles in need of verifying certificates,

which can be further categorized into four types, thus requiring different functions of the system. The first type are the roles of designing, issuing and managing certificates who is responsible for the designing of certificates according to the real situation in granting certificates. The second type is the administrative role. Unauthorized certificate issuing will cause chaos in the real world, in which distinguishing issuers becomes impossible. On the contrary, the administrative role contributes to coordinating works that by whom or by which organizations these certificates are designed and issued, thereby tracking system status and eliminating chaos. The third type is the role of certificate acquisition and acceptance.



Field	Description
id	Uuid of agency
name	Name of agency
pubkey	Ethereum address of agency
authaddress	Ethereum address of who authorize this agency
sign	Signature

Table 1.Data fields for certificate authority



Field	Description
id	Uuid of certifier
name	Name of certifier
pubkey	Public key of certifier
agencyid	Id of agency which this certifier works fo
state	State of certifier, valid or invalid
sign	Signature

Table 2.Data fields for certificate issuer

Certificate issuance needs to be delivered to a designated person, organization or institution. A certificate is meaningful only when it is been acquired and accepted. The fourth role type is certificate verifier. The certificate is regarded as an inter-medium proving to people that the receiver has accomplished something or has met certain qualifications. After being issued, certificates need to be seen, recognized, and/or to be proven, in which the role of certificate verifier is performed and realized. To fulfill different various requirements, certificate systems demand feasible and authentic administration, providing complete information while preventing certificates from being counterfeited and tampered, thus realizing convenience within its issuing, verifying and managing process. Moreover, the system's adequate independence and reliability owe its priority above other critical factors.

BlockCert solves certificate issuing and verifying problems via blockchain, whereas LLP solves certificate issuing, verifying and data storing problems by introducing a centralized document management system and DHT storage matching smart contract. By implementing cryptographic algorithms, the bitcoin network and DHT storages, SPROOF solves certificate issuing, verifying and data storing problems. While the above methods partly meet the certificate system's requirement, despite solving certificate counterfeiting and tampering problems to some extent, there exists no effective method that solves all potential difficulties and problems in certificate systems.

OVERVIEW IN SYSTEM DESIGNING

This system adopts smart contract to realize authority management, certificate issuing, certificate verification and certificate revocation. Authority management is structured as a three-level adjustable mode, in which the contract deployer decides whether to adopt three-level or non-three-level authority mode when deploying. This mode divides the participants of certificate authority into system managers, certificate issuing agencies and certificate issuers, in which the managers deal with the authority of certificate issuing agencies, while the agencies handle the authority of the corresponding issuers. Yet, the non- three-level mode classifies the participants of certificate authority into certificate issuing agencies and certificate issuers, in this mode, certificate issuing agencies can directly issue certificates based on their own demands without requiring the system man- agers' administration, while certificate issuers' authority need to be managed under scrutiny by certificate issuing agencies.

Certificate data is stored in independent smart contracts, through which the relevant data is accessed when it is issued, verified, and revoked. Any access to the data will be checked by authorities; only users with corresponding permissions can access the data by calling the certificate management contract, and the smart contracts storing the data can only be operated by the certificate management contract.

The whole system does not rely on any third-party system other than blockchain smart contract. The system provides certificate export for external data interface. The format of export data is compatible with version 2.0 of the Open Badges proto- col mentioned in the BlockCert project with due modifications, improving system convenience, designing optional graphical tools, as well as facilitating user friendliness. By adopting the smart contract language called Solidity, the system is developed on the version of Ethereum permissioned blockchain, which shows no difference against direct implementation on the Ethereum.

SMART CONTRACT BASED CERTIFICATE DATA MODEL

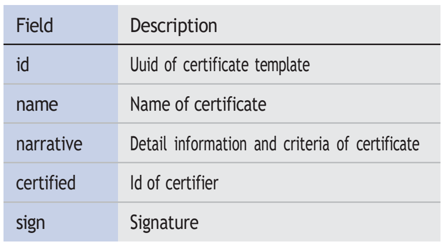
According to the characteristics of blockchain storage, we pro- pose a protocol that conforms to the blockchain smart con- tract storage, a temporary calling named Open Certificates, of which is used to standardize the certificate information, thereby facilitating data compatibility between different systems. In the protocol, only the key information of the certificate is kept, and the certificate data is categorized into issuing agencies data, issuer data, certificate template data, receiver data and certificate data. The public key is to map the 32-byte private key to 65 bytes by adopting the elliptic curve ecdsa-secp256k1 [14, 15] algorithm, while ultimately determining the relationship between public key and private key through the cryptography algorithm. Hence, all the data in this system are scrutinized by the public key contained in the data for signature verifications, thereby examining whether the certificate data is modified [16].

The certification authority data shown in Table 1 is applied by the certification authority in the three-level authority mode and is accordingly added to the blockchain by the system manager. The data contains information concerning the issuing authority, the key pair applied by the authority, and the key pair used by the system manager who added the authority's information; the data signature is also stored in the corresponding data simultaneously. In non-three-level mode, the certificate issuing authority can submit the data to the blockchain on its own.

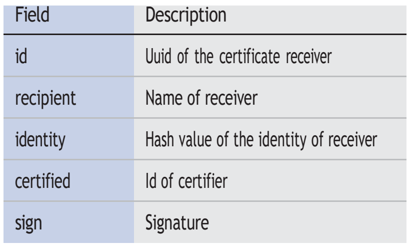
Certificate issuer data is supervised by the certificate issuing authorities. According to its own operation demands, the certificate issuer adds and manages the certificate issuer, in which the data includes the issuer information, issuer public key, and the issuing authority that adds the data and the signature of issuer status and data. Table 2 shows the data of a certificate issuer.

In practice, countless certificate copies of the same kind are given to different certificate receivers. For example, within this year, 120 college students from the School of Computer Science and Technology in a university will receive their degree certificates issued by the university upon graduation, among which the information merely differ in various individual names. At the same time,certain certificate receivers accept various certificates. For example, as a record breaker of Guinness record, Ashrita Furman has broken more than 600 official Guin- ness World Records, keeping 226 Guinness World Records until now, which means he owns more than 600 certificates that share the same receiver information whereas the certificates' information differs. Therefore, we divide and further categorize the certificate information into three types, certificate template information, certificate receiver information and certificate information. In a certificate, the certificate template information records all the general information other than receiver information. The certificate receiver information focuses on the personal data of the certificate receiver. The certificate information records the issuing date and the certificate issuer, associating the adopted template with recipient information. When multiple copies of the certificate need to be issued in this way, only the relational data between the template and the recipient is added, which greatly improves the efficiency in data storage and in processing. In Table 3, the template data that matches the paper certificate information is designed and filled by the issuer, which contains the template name, certificate details, the issuer who generated the template, and the signature of the template data.

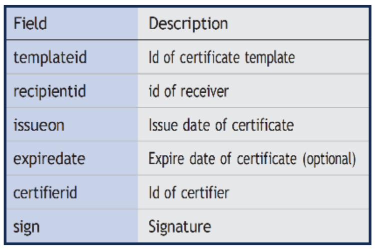
In practice, countless certificate copies of the same kind are given to different certificate receivers. For example, within this year, 120 college students from the School of Computer Science and Technology in a university will receive their degree certificates issued by the university upon graduation, among which the information merely differs in various individual names. At the same time, certain certificate receivers accept various certificates. For example, as a record breaker of Guinness record, Ashrita Furman has broken more than 600 official Guinness World Records, keeping 226 Guinness World Records until now, which means he owns more than 600 certificates that share the same receiver information whereas the certificates' information differs. Therefore, we divide and further categorize the certificate information into three types, certificate template information, certificate receiver information and certificate information.



Field	Description
id	Uuid of certificate template
name	Name of certificate
narrative	Detail information and criteria of certificate
certified	Id of certifier
sign	Signature



Field	Description
id	Uuid of the certificate receiver
recipient	Name of receiver
identity	Hash value of the identity of receiver
certified	Id of certifier
sign	Signature



Field	Description
templateid	Id of certificate template
recipientid	id of receiver
issueon	Issue date of certificate
expiredate	Expire date of certificate (optional)
certifierid	Id of certifier
sign	Signature

Table 3.data fields for certificate template. Table 4.Data fields for certificate receiver. Table 5.data fields for certificate

In a certificate, the certificate template information records all the general information other than receiver information. The certificate receiver information focuses on the personal data of the certificate receiver. The certificate information records the issuing date and the certificate issuer, associating the adopted template with recipient information. When multiple copies of a certificate need to be issued in this way, only the relational data between the template and the recipient is added, which greatly improves the efficiency in data storage and in processing. In Table 3, the template data that matches the paper certificate information is designed and filled by the issuer, which contains the template name, certificate details, the issuer who generated the template, and the signature of the template data.
The receiver information displayed on the paper certificate is public. In this system, the corresponding information is designed as independent data, in which only the receiver names and user identities are preserved. It only needs to retain the recipient's name and user's identities. The user identity performs the Hash value by calculating the receivers' identity information, there- by ensuring the irreversibility of Hash query. Only when the certificate receiver provides the identity information can hash comparison be performed, thus enabling the protection for user privacy. In Table 4, the certificate receiver data is filled in by the issuing authority and can only be added instead of being modified. The receiver data contains the name, the Hash value calculated from the identity information, the issuer information and the data signature.
In Table 5, certificate data is generated by the issuer with regard to certain factors concerning template, receiver information and issuing time, which is embodied with template ID, receiver ID, issuing date, optional expiration date, issuer information and data signature.
The public and private key pair of the user is generated throughout the cryptographic algorithm, which enjoys a unique correspondence relation and is adopted in the smart contract to confirm the user's identity and authority. By checking the recorded certificate content and the public key used for signature, we can scrutinize whether the certificate data is fabricated or tampered.
In this system, all the data is stored in smart contracts and can only be accessed through the data management contract named CertManage Contract. The caller's permissions are checked against their address to guarantee that only the issuing authority adds a new issuer within its agency, thereby further ensuring that only authorized issuers add templates, issue and revoke certificates within their agency.
The issuing authority data is stored in the CertAgency Contract, the issuer data is stored in the Certifier Contract, the template data is stored in CertTemplate Contract, the receiver data is stored in the Receiver Contract, and the certificate information is stored in the Certificates Contract.

AUTHORITY MANAGEMENT FOR THREE-LEVEL SYSTEM

In the three-level system authority management, the system manager, the certificate issuing agency and the issuer have different authorities and access methods. The manager can be selected by forming an alliance committee or by the smart

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12417

contract deployer. In the process of adding a certificate issuing agency, the certificate issuing agency first provides the system manager with the agency information and the public key it uses, then the system manager reviews and verifies the authenticity and qualification of the certificate issuing agency applying for registration. After the approval, the certificate issuing agency can implement routine operations within the system.

The information of the certificate issuing agency and the public key it uses are stored in the CertAgency Contract. Adding data can only be operated by the system manager through the CertManage Contract. The process of adding its authority is shown in Fig. 1:

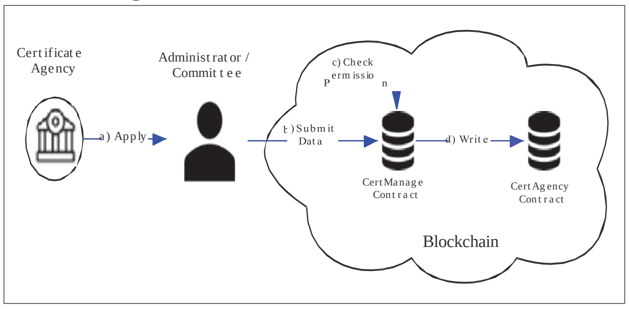


Figure 1.The workflow of adding agency through smart contract

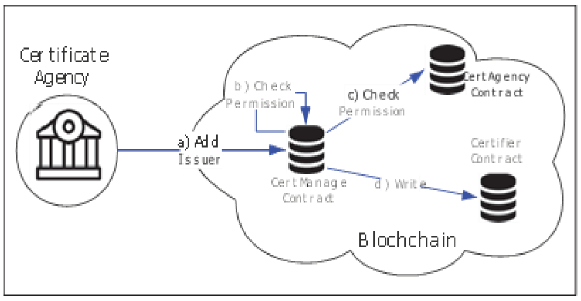


Figure 2. The workflow of adding issuer through smart contract.

- The certificate issuing agency submits its application to the system administrator or certificate alliance committee.
- The system administrator or the certificate alliance committee calls CertManage Contract to submit both the certificate agency's public key and its information.
- CertManage Contract checks whether the caller's permission conforms to the management permission.
- After the check is passed, the issuing agency data is written into CertAgency Contract.

Under the non-three-level authority mode, the certificate issuing agency can register itself as the issuer in the contract by calling CertManage contract without the system manager's permission.

The procedure of adding a certificate issuer to the smart contract is shown in Fig. 2:

- Submit certificate issuer data.
- Check the corresponding calling permission.
- Check the caller's authority via CertAgency Contract.

Write the issuer data into Certifier Contract after the permission is checked Subsequent management operations of the certificate issuer on the certificate will undergo throughout the permission check by CertManage Contract. After being authorized successfully, the certificate issuer can add or revoke certificates, and the relevant operations will be checked by CertManage Contract, in which any certificates cannot be operated without permission.

THE REALIZATION OF CERTIFICATE MANAGEMENT VIA SMART CONTRACT

The certificate issuer calls CertManage Contract, adding a template in CertTemplate Contract. When issuing a certificate, the issuer chooses the suitable template, fills in the receiver information, and signs the information. After signing, CertManage Contract will write the relevant data into the Certifier Contract. If the receiver does not exist, its information will be added into the Receiver Contract.

The specific procedure of certificate issuance is:

- Certificate issuer gathers receiver information.
- Submit information of the certificate being issued by calling CertManage Contract.
- Check permission through CertManage Contract.
- Call the Certifier Contract to check the issuing authority.
- Call the CertAgency Contract to check the current issuing authority.
- Call the CertTemplate Contract.
- Call the Receiver Contract to check receiver.
- Write the certificate information into Certifier Contract after passing all checks.
- Give feedback of the certificate issuing result to the issuer.
- The issuer informs the receiver of the issued certificate information.

The certificate receiver obtains from the issuer its own certificate which includes number, hash value, name, issuing date and receiver information.

The certificate receiver provides the certificate information to the person or institution who needs to verify the certificate if necessary. By calling the smart contract or adopting the tools provided, the certificate verifier performs its verification.Figure 3 shows the smart contract algorithm in verifying certificate information, which can be divided into seven steps:

- Call the smart contract, retrieve data of the entered Certificate to check whether the certificate is in the blockchain; if so, proceed to next verification.
- If the certificate can be found in the blockchain, the certificate information is checked by the smart contract. If the calculated signature is consistent with the existing signature, proceed to next verification.
- Check whether the certificate issuer has been removed; if not, proceed to next verification.
- Check if the current certificate expires using the expiry date in the certificate; if not, proceed to next verification.
- Accomplish certificate verification and return the validation result to the smart contract caller.

```
PROCEDURE VERIFY(certificatehash)
certificate <- Get from Certificate Contract according to certificatehash IF certificate NOT EXIST THEN
RETURN FALSE
template <- Get from CertTemplate Contract according to certificate.certid IF template NOT EXIST THEN
RETURN FALSE
recipient <- Get from Receiver Contract according to certificate.recipientid IF recipient NOT EXIST THEN
RETURN FALSE
issuer <- Get from Certifier Contract according to certificate.issuerid IF issuer NOT EXIST THEN
RETURN FALSE
agency <- Get from CertAgency Contract according to issuer.agencyid IF agency NOT EXIST THEN
RETURN FALSE
IF TRUE != check_signature(recipient.recipient, recipient.identity, recipient.sign, issuer.pubkey) THEN
RETURN FALSE
IF template.issuerid != issuer.uuid OR TRUE!= check_signature(template.name, template.narrative, template.sign,
issuer.pubkey) THEN
RETURN FALSE
IF TRUE != check_signature(certificate.certid, certificate.recipientid, certificate.issueon, certificate.expiredate,
certificate.sign, issuer.pubkey) THEN
RETURN FALSE
IF certificate.id IN issuer.revokelist or certificate.id IN agency.revokelist THEN RETURN FALSE
authorityaddress <- Authority address is set by the contract deployer authoritypubkey <- Authority public key is set by
the contract deployer
IF agency.authaddress != authorityaddress OR TRUE != check_signature(agency.name, agency.address, agency.sign,
authoritypubkey) THEN
RETURN FALSE
IF issuer.state == FALSE or TRUE != check_signature(issuer.name, issuer.address, issuer.agencyid, issuer.sign,
agency.pubkey) THEN
RETURN FALSE
nowdate <- Get the timestamp from transaction IF issuer.expiredate < nowdate THEN
RETURN FALSE
return TRUE
```

Figure 3. Algorithm to verify certificate in smart contract.

Once the certificate needs to be revoked due to certain problems, the certificate can only be revoked by the issuer or agency that issues the certificate. Records of certificate revocation will be kept in the issuer's or issuing authority's certificate revocation list. Expired status validation has been performed during the certificate validation process, which is not included in the scope of revocation.

The certificate revocation process is shown in Fig. 5:

- Call CertManage Contract, submit certificate hash value.
- Check caller permission.

- Check whether the caller has issuer permission.
- Check whether the caller has issuing agency permission.
- Retrieve data of the certificate to be revoked.
- Write the certificate revocation record into the Certifier Con- tract after passing the check.
- Write the certificate revocation record into the CertAgency Contract.

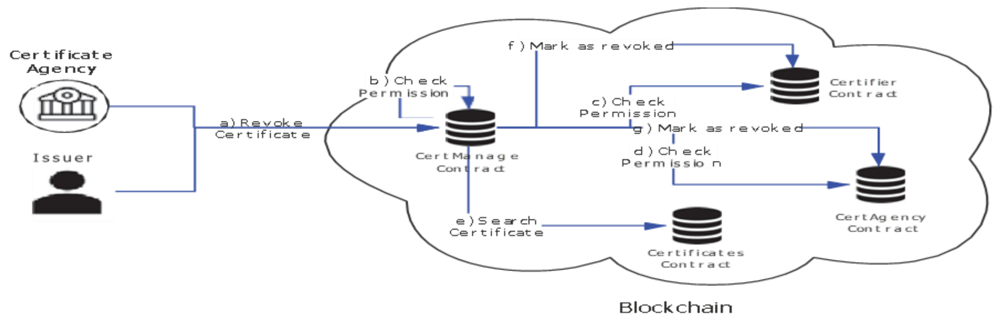


Figure 5. The workflow of revoking certificate through smart contract.

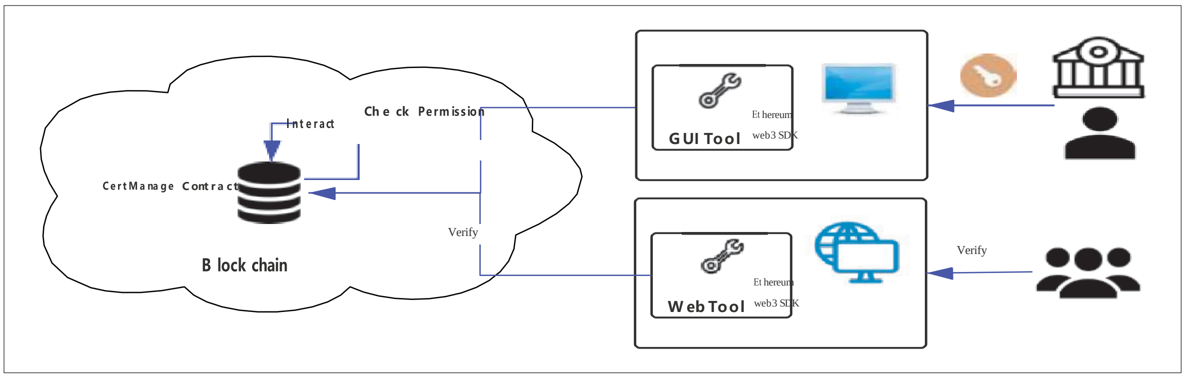


Figure 6. Demonstration of GUI Tools

IV. CASES IN PRACTICE

Based on the above implementations, we develop a block- chain certificate system on the Ethereum permissioned block-chain, thereby providing the blockchain certificate service for college students' innovation and entrepreneurship competition. The system's function includes the authority management of the certificate issuing agency and issuer, as well as the function of certificate issuing, verification and revocation. Although users can directly access the smart contract to per- form all system operations, certain graphical interface tools are accordingly developed to provide better user-friendliness, as is shown in Fig. 6. The Ethereum web3.js is built within the tools, thereby realizing convenient certificate quick query, verification and social sharing. Meanwhile, the developed small tools facilitate the process for calling smart contract that performs permission and certificate batch management, as shown in Fig. 7.

In this practice, the system provides interface-friendly design, facilitating compatibility with other tools or systems. The data obtained by calling the smart contract is expressed in JSON format, which merely demands format changing to be easily compatible with the other protocols.

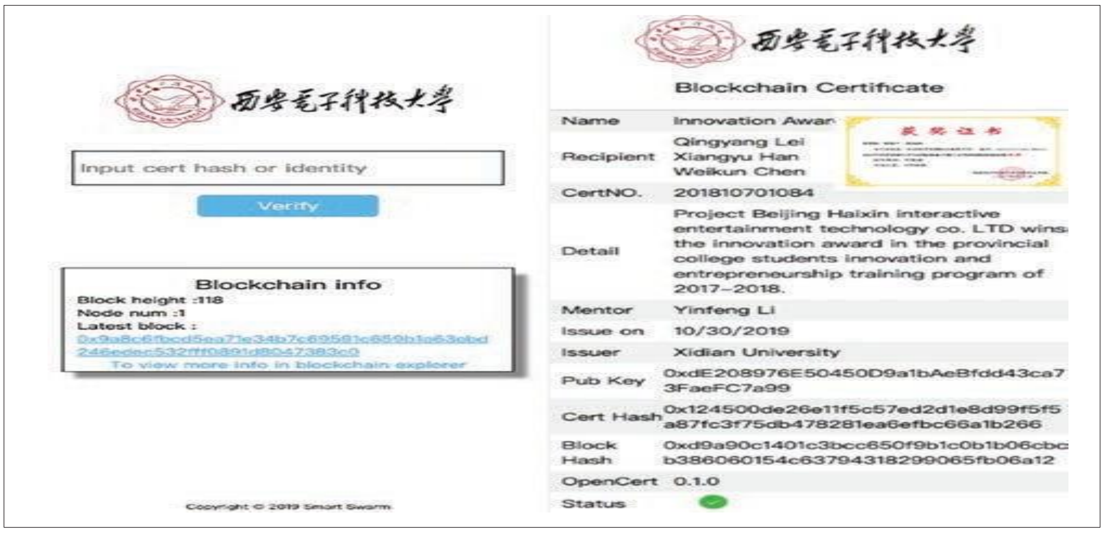
V. ANALYSIS

We introduce two modes in this system designing, i.e., three-level authority management and non-three-level authority management, which can be respectively applied in the situation where the authority management of the issuing agency is required, and where the corresponding management is not required. In practical application, smart contract is fully used to store and to manage certificates without relying on any third-party system, which is completely decentralized and enjoys good reliability. The information of the certificate receiver is not fully preserved, thereby eliminating information leakage of users' privacy on the premise of guaranteeing the integrity of certificate information. The protocol in this system is flexible, independent of certificate control, of which moderate modification helps achieve generalized document storage. Compared with BlockCert, LLP and SPROOF, the system enjoys the following advantages, as shown in Table 6:

- Complete information. The system provides complete certificate information, preventing meaningless Hash values, whereas not requiring the certificate issuer to provide a system for user verification.
- Authority separation. The system introduces the three-level authority mechanism of the manager, the issuing agency and the issuer. By revealing the public key information and other information of the manager, the issuing agency and the issuer, the verifier can distinguish different issuers and issuing agencies.
- High reliability. The system only adopts smart contract to solve the certificate system problems, neither does it rely on any third-party tools and systems, nor will it cause any problems concerning verification cheating. As long as the block-chain network is running, the certificate can be verified.

	BlockCert	LLP	SPROOF	This System
Smart contract	No	Partial	No	Yes
Permission management	No	Yes	No	Yes
Independency	No	No	No	Yes
Storage	Third party	Third party	Third party	Self-contained
Privacy protection	Yes	Yes	Yes	Yes
Full certificate information	No	No	No	Yes
Decentralization	No	No	Yes	Yes
Transparency	Yes	Yes	Yes	Yes
Completeness	No	No	Yes	Yes

Table 6. Distinctive features of Blockchain certificate system.



Blockchain info
Block height :118
Node num :1
Latest block :
0x9a8c6fbcd5ea71e34b7c69581c659b1a63abd246edec532fff0891d8047383c0
To view more info in blockchain explorer

Copyright © 2019 Smart Swarm

Blockchain Certificate

Name	Innovation Awar
Recipient	Qingyang Lei Xiangyu Han Weikun Chen
CertNO.	201810701084
Detail	Project Beijing Haixin interactive entertainment technology co. LTD wins the innovation award in the provincial college students innovation and entrepreneurship training program of 2017–2018.
Mentor	Yinfeng Li
Issue on	10/30/2019
Issuer	Xidian University
Pub Key	0xdE208976E50450D9a1bAeBfdd43ca73FaeFC7a99
Cert Hash	0x124500de26e11f5c57ed2d1e8d99f5f5a87fc3f75db478281ea6efbc66a1b266
Block Hash	0xd9a90c1401c3bcc650f9b1c0b1b06cbcb386060154c63794318299065fb06a12
OpenCert	0.1.0
Status	✓

Figure 7. Display of certificate information.

VI. CONCLUSION AND THE FUTURE

Compared with existing traditional certificate systems and other blockchain solutions, the blockchain certificate system pro-posed in this article is a set of certificate system based on block- chain technology, which completely adopts the smart contract and does not rely on any other third-party system, and can meet the requirements for practical applications, replace the existing traditional electronic certificate system, solve the problems that exist in the past blockchain certificate system, thereby further realizing practical application of the blockchain technology within the field of education, which enjoys great guiding significance in practice.

The current system is only focused on certificates while the basic idea of adopting smart contract for role definition, authority management, data anti-tampering and privacy protection can be used in other fields. In a traditional IoT system, typical devices are facing problems of security vulnerabilities while the thought of authority management based on blockchain can

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⋇ Peer-reviewed & Refereed journal ⋇ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12417

be used to solve these problems. Also, the idea of using smart contract and blockchain in this article can be used to secure data transition, secure data storage, control data access and protect privacy.
The potential issue we have is in three-level authority management, the system will be affected if the private key is lost or compromised. There should be a voting schema in the authority management system to make decisions according to all attendees' opinions in this system to avoid the affect from that private key.

Our next step is to extend the smart contract to adopt new authority management schema, to develop general processes and to define the protocol for digital data storage in smart contract for general purpose.

REFERENCES

[1]. S. A Brands, Rethinking Public Key Infrastructures and Digital Certificates: Build- ing in Privacy, MIT Press, 2000.
[2]. M. Chase and S. S. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Computer and Communications Security, 2009,pp. 121–30.
[3]. D. Boneh et al., "A Method for Fast Revocation of Public Key Certificates and Security Capabilities," USENIX Security Symp., 2001, pp. 22–22.
[4]. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008, https:// bitcoin.org/en/bitcoin-paper, accessed Nov. 20, 2019.
[5]. V. Buterin, "Ethereum: A Next Generation Smart Contract & Decentralized Application Platform," https://github.com/ethereum/wiki/wiki/White-Paper, accessed Nov. 20, 2019.
[6]. L. Luu et al., "Making Smart Contracts Smarter," Computer and Communications Security, 2016, pp. 254–69.
[7]. K. Ohara, "Smart Contracts — Dumb Idea," IEEE Internet Computing, vol. 22, no. 2, 2017, pp. 97–101.
[8]. J. Nazaré and K. Hamilton, "Digital Certificates Project," http://certificates. media.mit.edu/, accessed Nov. 20, 2019.
[9]. W. Gräther et al., "Blockchain for Education: Lifelong Learning Passport," https://dl.eusset.eu/handle/20.500.12015/3163, accessed Nov. 20, 2019.
[10]. C. Brunner, K. Fabian, and E. Dominik, "SPROOF: A Platform for Issuing and Verifying Documents in a Public Blockchain," Int'l. Conf. Information Systems Security, 2019, pp. 15–25.
[11]. P. Maymounkov and D. Mazieres, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," Int'l. Wksp. Peer to Peer Systems, 2002,pp. 53–65.
[12]. W. Cai et al., "Decentralized Applications: The Blockchain-Empowered Soft- ware System," IEEE Access, 2018, pp. 53019–033.
[13]. C. Konstantinos and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," IEEE Access, 2016, pp. 2292–303.
[14]. D. H. Johnson, A. Menezes, and S. A. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," Int'l. J. Information Security, vol.1, no. 1, 2001,pp. 36–63.
[15]. SECG, "SEC 2: Recommended Elliptic Curve Domain Parameters," http:// secg.org/sec2-v2.pdf, accessed Nov 20, 2019.
[16]. A. Ezell and J. Bear, Degree Mills: The Billion-Dollar Industry That Has Sold over a Million Fake Diplomas, Prometheus Books, 2005.
[17]. G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing Privacy: Using Blockchain to Protect Personal Data," IEEE Symp. Security and Privacy, 2015,
pp. 180–84.
[18]. K. Ahmed, "Hawk: The Blockchain Model of Cryptography and Privacy-Pre- serving Smart Contracts," IEEE Symp. Security and Privacy, 2016, pp. 839–58.
[19]. R. L. Rivest, A. Shamir, and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Commun. ACM, vol. 21, no. 2, 1978, pp. 120–26.
[20].R. C. Merkle, "A Certified Digital Signature," Int'l. Cryptology Conf., 1989,pp. 218–38.
[21]. M. Naor and K. Nissim, "Certificate Revocation and Certificate Update," IEEE JSAC, vol. 18, no. 4, 2000, pp. 561–70.
[22]. Open Badges, "Open Badges v2.0 IMS Final Release," https://www.imsglob-al.org/sites/default/files/Badges/OBv2p0Final/index.html, accessed Nov. 20, 2019.
[23]. IPFS, "Peer-to-Peer Hypermedia Protocol," https://ipfs.io, accessed Nov. 20, 2019.
[24]. F. Meneghello et al., "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," IEEE Internet of Things J., vol. 6, no. 5, 2019, pp. 8182–210.
[25]. O. Bello and and S. Zeadally, "Intelligent Device-to-Device Communication in the Internet of Things," IEEE Systems J., vol. 10, no. 3, 2016, pp. 1172–82.
[26]. Y. Zhang et al., "Smart Contract-Based Access Control for the Internet of Things," IEEE Internet of Things J., vol. 6, no. 2, 2019, pp. 1594–605.
[27]. O. Novo, "Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT," IEEE Internet of Things J., vol. 5, no. 2, 2018, pp. 1184–95.
[28]. A. Zanella et al., "Internet of Things for Smart Cities," IEEE Internet of Things J., vol. 1, no. 1, 2014, pp. 22–32.

DOI: 10.17148/IARJSET.2025.12417

BIOGRAPHY

N. Jaya Santhi ,M.Tech, Asst. Professor,
Dept of Computer Science & Engineering,
BWEC, Andhra Pradesh,India

K.Lakshmi [B.Tech],Student,
Dept of computer science & engineering,
BWEC, Andhra Pradesh,India

K.Ayusha [B.Tech],Student,
Dept of computer science & engineering,
BWEC, Andhra Pradesh,India

A.Sravani [B.Tech],Student,
Dept of computer science & engineering,
BWEC, Andhra Pradesh,India

K.Deevena [B.Tech],Student,
Dept of computer science & engineering,
BWEC, Andhra Pradesh,India