IU-SmartCert: A Blockchain-Based System for Academic Credentials with Selective Disclosure

Thanh-Tung Tran[1,2](✉) and Hai-Duong Le[1,2]

[1] International University, Ho Chi Minh City, Vietnam
{tttung,lhduong}@hcmiu.edu.vn
[2] Vietnam National University, Ho Chi Minh City, Vietnam

Abstract. Blockchain-based systems for academic credentials have shown the potential to overcome existing limitations of paper-based credentials. These systems define procedures to issue and manage credentials using blockchain technology to enhance security and reduce the administrative cost of both universities and employers. However, more effort is needed to provide features that empower the flexibility and privacy of learners. In this paper, we present `IU-SmartCert`, a blockchain-based system and procedures for issuing, verifying and exchanging academic credentials with selective disclosure feature. The security analysis of the proposed system shows that while learners can selectively present their credentials, the validity and integrity of the credentials are verifiable. Also, the proposed system is general and can be extended for other domains in the future.

Keywords: Blockchain · Academic credential · Multiple levels of granularity · Selective disclosure

1 Introduction

Blockchain-based systems for digital academic credentials have become an emerging research topic [7,8,18,19]. Such a system consists of a set of procedures to issue, exchange, revoke and verify credentials in which the blockchain technology is used as a repository of digital fingerprints of the original credential. These procedures protect the integrity of issued credential, and allow relying parties to validate a credential without involvement of the issuer.

The usability and the value of these systems comes from features that they provide to issuers, learners and relying parties. The previous of blockchain-based systems for education focused on bringing traditional procedures to blockchain. These systems helps to protect credentials from counterfeit and to provide a better mechanism to verify credentials. However, in the current trend of learner centered education [7,12], more effort is needed to provide features that empower the flexibility and the privacy of learners.

T. K. Dang et al. (Eds.): FDSE 2021, CCIS 1500, pp. 293–309, 2021.
https://doi.org/10.1007/978-981-16-8062-5_20

In this paper, we revisit the procedures of these blockchain-based systems, and develop a prototype system with selective disclosure feature for digital credentials. This feature allows learners to choose some components of their credential to present to a relying party, and also allows them to choose not to disclose some others. Thereby, it give learners more control over their credentials and their privacy.

In particular, we make the following contributions:

- Proposing new procedures to define and issue credentials with many levels of granularity
- Proposing a procedure to exchange credentials with a selective disclosure option

In the next section, we present related work. We then provide an overview of our proposed system `IU-SmartCert` and shows in detail procedures to manage credentials with selective disclosure feature. After that, the security analysis, and an proof of concept implementation of the proposed system are presented. And the paper closes with a discussion and conclusion section.

2 Related Work

Using blockchain technology for education has been studied to leverage the distributed ledger to issue and secure academic credentials. Blockcert [11] was the first important open-source system using Bitcoin blockchain [15] for academic credentials. The project was the collaboration between the MIT Media Lab Learning Initiative and Learning Machine in 2016. Blockcert defines procedures for creating, exchanging, and verifying academic credentials. The credential includes information of the issuer, information of the learner, issue date and the achievement of the learner. The credential is signed and published to blockchain, hence a relying party can validate it. Since 2017, many pilots project using Blockcerts for education have been developed in Malta [5], Italy [2]. This suggests that the use of blockchain for education is mature enough for official adoption.

Several authors [7,8,16,17,19] furthered the idea by using smart contracts on Ethereum blockchain network [20]. The Blockchain for education platform in [8] improved the security and the privacy of the issuing procedures, and specially the identity of issuers. In the same year, EduCTX [19] proposed to use blockchain to enable credit transferring among higher education institutions. The system defined procedures to manage academic credentials at a finer gain such as the score and the credit of a course. In the same theme, the Knowledge Institute, Open University in UK has developed smart contracts to document educational microcredentials [7].

These studies show that blockchain technology can help to overcome limitation of traditional paper-based credentials. It helps issuers to ensure the integrity of credentials, and also helps relying parties to verify credentials without contacting issuers, by that reduces administrative bureaucracy.

Recently, in a report of the European Union [7] and the latest report of the American Council on Education of US [12], key themes of the next generation of applications using blockchain for education are not only on issuing and verifying credentials but also on the ability to empower lifelong learning and data privacy.

Following these key themes, in this paper, we propose a blockchain-based system for issuing credentials at different granularity levels, and for exchanging credentials with a selective disclosure option. Although, these new features would be essential for future adoption of blockchain technology in education, as far as we know, the closest project that mentioned them is [3,14] and there are no other studies on them.

3 Proposed Solution

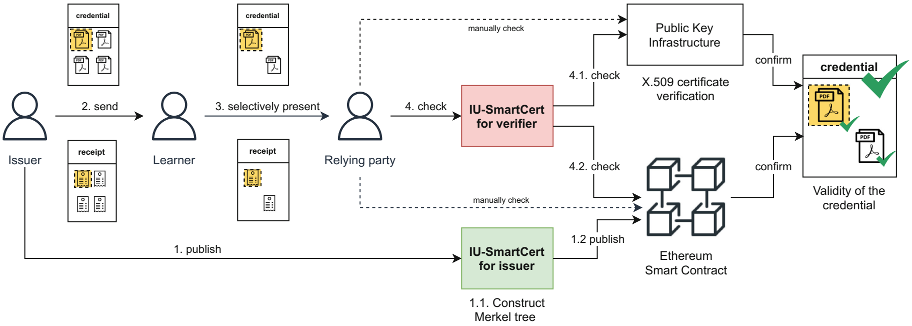


Fig. 1. Overview of procedures in IU-SmartCert.

This section outlines the proposed system IU-SmartCert, a blockchain-based educational credential management system with a selective disclosure option.

Our system has 3 main groups of users: issuers, learners, and relying parties. An issuer can be broadly understood to include individuals or organizations qualified to issue a credential. A learner is a person who received the issued credential. A relying party in this paper is used to refer to a person or a company who uses and wants to check the validity of a credential.

Figure 1 shows key functions and typical interactions among users of our system. First, given a set of credentials to issue, the issuer organizes each credential into our proposed format that composes of two parts: a mandatory component and a list of optional components, then inputs the fingerprint of all credentials to IU-SmartCert. From the input fingerprints, the system constructs a Merkle tree and publishes the root node of the tree along with the issuer identity to the public Ethereum blockchain as a smart contract. After publishing, the issuer sends a digital credential and its receipt to each learner. Later when the learner needs to show her/his credential to a relying party, like an employer, s/he can select the most relevant optional components of the credential to present along with the mandatory component to the employer. Finally, a relying party verifies the authenticity and the integrity of a given credential by checking the received data against the public key infrastructure and the Ethereum public blockchain. The verification process can be done independently without contacting the issuer, and even without using the `IU-SmartCert` system.

Formally, our blockchain-based credential management system provides functions to

R1. Define and issue credentials with a mandatory component and optional components for selective sharing,
R2. Verify and validate a credential,
R3. Select optional components of the credential to disclose,
R4. Revoke a credential issued by the system

In addition, to increase user's flexibility and initiative, the system should have the following quality attributes:

R5. Ability to independently check the integrity and validity of a credential
R6. Security for credentials.

In the following sections, we describe in detail important procedures in the `IU-SmartCert` system.

3.1 Defining a Credential Schema

In `IU-SmartCert`, issuers of credentials like universities and institutions can define their own schema and vocabularies of a credential for each program. A credential in `IU-SmartCert` composes of two parts:

- One mandatory component
- and a list of optional components

A mandatory component is one that learner must disclose to every replying party. This component stores all information needed to validate and evaluate a credential. For instance, in a credential of an undergraduate student, the mandatory component is the diploma which contains the name of the university, the name of the learner, the issued date, and the enrollment year.

The list of optional components is a tool for issuers to define a flexible credential data model. An issuer can issue credentials with different levels of granularity, therefore allow learners to selectively disclose their credential on demand. For instance, a university could issue a transcript of a student by issuing the score of each course separately as a list of optional components. This credential schema allows the student to choose courses to disclose to a relying party. Conversely, if a university does not allow students to cherry-pick courses to disclose, the university issue the entire transcript as a single component of the credential. Moreover, in a stricter rule that requires students to disclose the transcript along with their diploma, the university can combine the diploma and the transcript into a single mandatory component in the input to the `IU-SmartCert`. Therefore, the structure helps to fulfill the requirement **R1** of the system.

The procedure to define a schema and vocabularies in `IU-SmartCert` is performed via the data input of an issuer. For every credential, the issuer must provide a list of files corresponding to each component in the credential with the following naming rule

```
CredentialID.ComponentName.(R).ext
```

where

- `CredentialID` is the identity of a credential and all components of a credential will share the same value.
- `ComponentName` is the name of a component.
- `.(R)` is the marker for the mandatory component. An optional component's filename does not include this marker.
- `.ext` is the file extension.

For instance, consider a bachelor's credential composes of a diploma, a transcript and a scientific profile. The issuer could define that the diploma is a mandatory component, while the transcript and the scientific profile are optional components. In such case, the input to the `IU-SmartCert` of a credential with identifier `ITIU01` is 3 pdf files named as following

```
IUIT01.diploma.(R).pdf
 IUIT01.transcript.pdf
    IUIT01.profile.pdf
```

3.2 Issuing Credentials

When learners complete a program, the institution generates digital credentials of learners and passes them to the `IU-SmartCert` system. The system then organizes and publishes the digital fingerprint of those credentials to the Ethereum blockchain in such a way that learners can choose components of their credential to disclose and a relying party can validate a given credential (Fig. 2).

Constructing Merkle Tree. To begin this procedure, we use the schema and the digital fingerprint of each credential to construct a Merkle tree [13]. We first combine the identity of the credential with the content of the component and its type, mandatory or optional, defined by the schema, then uses `SHA-256` to calculate the hash value of the combination and create a leaf node in the Merkle tree. We apply this procedure to all components of all input credentials and obtain corresponding leaf nodes. From those leaf nodes, we build the rest of the tree by concatenating two leaf nodes, calculating a new hash value resulting in a parent node, and continuing up toward the root of the Merkle tree.

We can construct a Merkle tree from any arbitrary number of credentials and always results in a single root node, so an institution could issue credentials in batch to save time and effort.

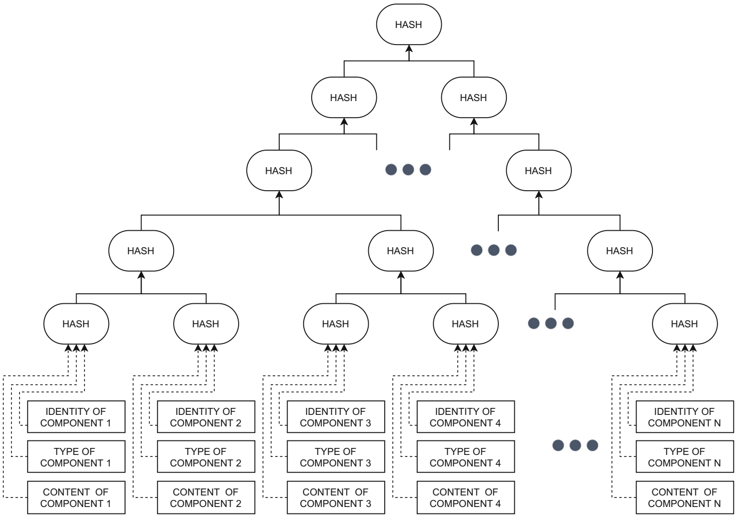


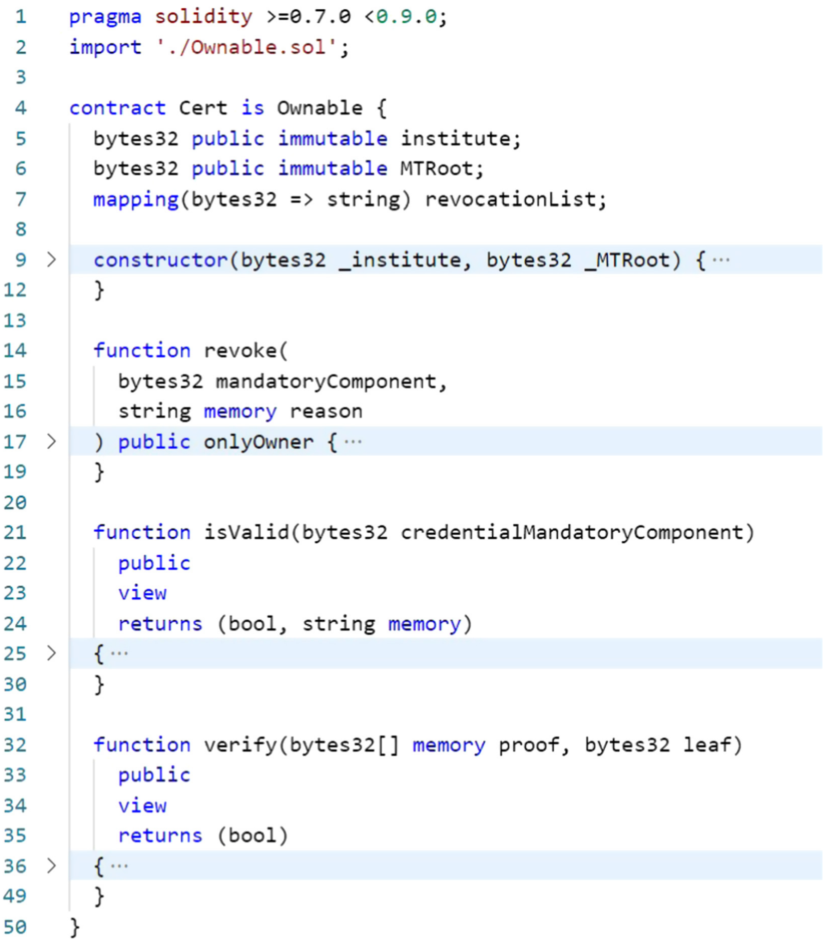
Fig. 2. A merkle tree of a batch of credentials

Publishing Data as a Smart Contract. Once the Merkle tree is built, the `IU-SmartCert` system publishes the hash value of the root node and supporting data to the Ethereum blockchain as a smart contract so that a relying party can validate an issued credential independently without contacting the issuer. The smart contract (as shown in Fig. 3) consists of

- `institute` A read-only variable for the hash of the issuer name. The issuer name is the organization field in the issuer's X.509 certificate, and thus binds to the identity of the issuer.
- `MTRoot` A read-only variable for the hash of the root of the Merkle tree.
- `revocationList` A list of revoked credentials along with a reason of the revocation. The function is accessible only for the issuer which is the owner of the contract. See Sect. 3.5 for the revoking procedure.
- `verify(bytes32[], bytes32)` A function to check whether a component belongs to the Merkle tree represented by the root node stored in `MTRoot` of the contract.
- `isValid(bytes32)` A function to check whether a credential is revoked. If the credential is revoked, the function returns `false` with a reason, otherwise, returns `true`.
- `revoke(bytes32, string)` A function to revoke a credential. See Sect. 3.5 for the revoking procedure.

It is worth noting that the smart contract stores only the root node of the Merkle tree constructed from the batch of credentials, and no other data about the credentials are published.

When the deployment of the smart contract is confirmed, the system keeps the metadata to generate receipts for learners.



```
pragma solidity >=0.7.0 <0.9.0;
import './Ownable.sol';

contract Cert is Ownable {
  bytes32 public immutable institute;
  bytes32 public immutable MTRoot;
  mapping(bytes32 => string) revocationList;

  constructor(bytes32 _institute, bytes32 _MTRoot) {···
  }

  function revoke(
    bytes32 mandatoryComponent,
    string memory reason
  ) public onlyOwner {···
  }

  function isValid(bytes32 credentialMandatoryComponent)
    public
    view
    returns (bool, string memory)
  {···
  }

  function verify(bytes32[] memory proof, bytes32 leaf)
    public
    view
    returns (bool)
  {···
  }
}
```

Fig. 3. Smart contract to manage a batch of credentials

3.3 Generating Receipts

After publishing credentials to Ethereum network, the system generates a receipt for each credential and sends it along with the digital credential to the corresponding learner. A relying party will use it to verify and validate the credential.

A receipt contains metadata of the smart contract, a proof of existence in the Merkle tree of each component of the credential, and the X.509 certificate of the issuer.

The metadata in a receipt is to help a relying party to verify the identity of the issuer and the validity of the smart contract. The metadata consists of the address of the smart contract that manages the credential, the hash value of the transaction that deployed the smart contract and the identity of the issuer. To provide a verifiable identity, we use a hybrid approach [6] where we establish a binding between accounts in Ethereum and an X.509 public key certificate of the issuer. This X.509 certificate is issued by a trusted certificate authority (CA) of the traditional public key infrastructure (PKI), which comprises of hardware and software for creating, managing, distributing, using and revoking public keys. At registration time, CA verifies the identity of the issuer and writes its information into the X.509 certificate; then, CA digitally signs this certificate with its private key [1]. Therefore, originality of digital credentials in this system is guaranteed. Before using our system `IU-SmartCert`, the issuer needs to endorse the Ethereum account used to publish credentials by signing the respective address with the private key of the X.509 certificate. Then, the issuer input to the system the address of its Ethereum account, the signature, and its X.509 certificate chain. Later, replying parties like employers can retrieve from the receipt the information to verify them versus their trusted certificate authority, thereby authenticating the identity of the issuer and the validity of the smart contract.

In a receipt, the proof of existence of a component is extracted from the Merkle tree build in the issuing phase. Since each component corresponds to a leaf node of the tree, its proof of existence is a list of nodes in the Merkle tree required to recalculate the path from the corresponding leaf node to the root node [13]. In an example in Fig. 4, the proof of existence of component 3 is the hash value `Hash 4` and the hash value `Hash 12`. From these proofs, one can calculate the path from component 3 to the root node.

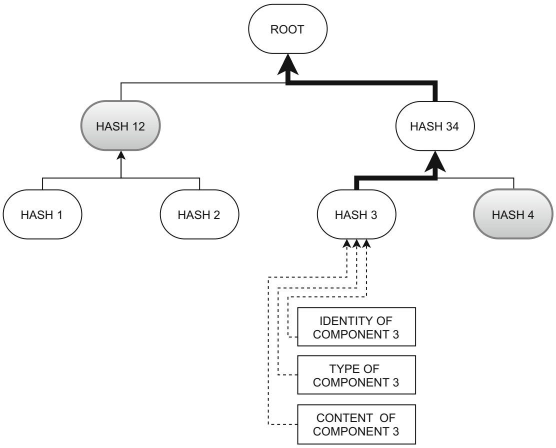


Fig. 4. Proof of existence of a component. `Hash 4` and `Hash 12` is the proof of existence of component 3 in the Merkle tree.

Finally, the system generates one receipt for each credential as a file in JSON data format (as shown in Fig. 5) with the following fields:

- `issuedOn`: the time when the credential was published to the blockchain

- `transactionHash`: the hash value of the transaction that deployed the smart contract that manages the credential
- `contractAddress`: the address of the smart contract that manages the credential
- `credentialID`: the identity of the credential
- `components`: the data to prove the authenticity of the credential. For each component of the credential, the data includes
  - `name`: the name of the component
  - `mandatory`: a boolean value indicating whether the component is mandatory or not
  - `proof`: the proof of existence of the component in the Merkle tree
  - `hash`: a hash value of the concatenation of the credential's identity, the type, and the content of the component.
- `issuer`: the verifiable identity of the issuer
  - `ethereumAccount`: the Ethereum account of the issuer
  - `ethereumAccountSignature`: the signature of the issuer's Ethereum account endorsed by the private key of the X.509 certificate
  - `IssuerCertificateChain`: the chain of the X.509 certificates of the issuer in PEM format

3.4 Exchanging Credentials

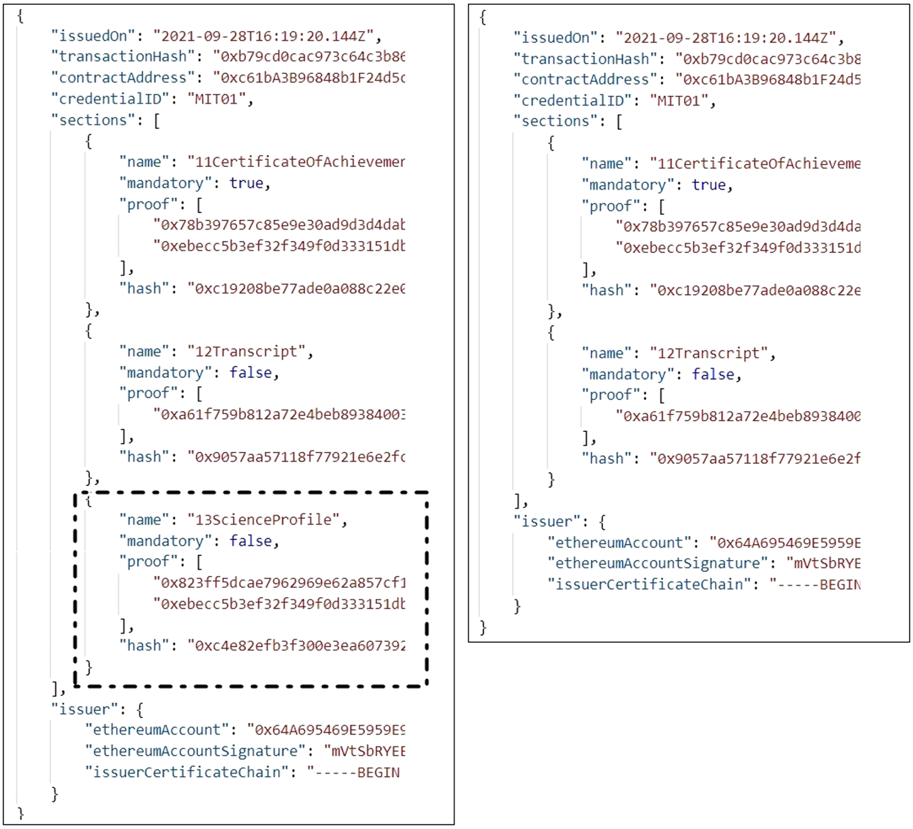
In our proposed system, learners can exchange their credentials with a selective disclosure option. In other words, learners can choose components to share, and also can choose components not to share while following the schema defined by issuers (Requirement **R3**).

The exchanging credentials begin when an employer requests the learner to present his credential. First, the learner takes his credential and picks the most relevant components to share. This selection is possible because credentials in our system are organized into two parts a mandatory component and a list of optional components. And since the type of each component is defined by the issuer, the learner can freely pick and skip some optional components without invalidating the credential.

After the selection, the learner needs to generate a new receipt which is the proof of existence for those selected components. He can upload his original credential's receipt to the `IU-SmartCert` system, select the chosen components, and download a new receipt. On the other hand, the learner can make the new receipt on his own without using the system. He can open his receipt in a text editor, remove the sections corresponding to the components that are not chosen, and save the file as a new receipt.

For example, in Fig. 5 there are two valid receipts of one credential. The receipt on the left is used when the learner would like to show all of the three components of his credential. The receipt in the right is used when the learner chooses not to share the *ScienceProfile* component.

Finally, the learner sends the new receipt and related files to the employer.



```
{
    "issuedOn": "2021-09-28T16:19:20.144Z",
    "transactionHash": "0xb79cd0cac973c64c3b8
    "contractAddress": "0xc61bA3B96848b1F24d5
    "credentialID": "MIT01",
    "sections": [
        {
            "name": "11CertificateOfAchievemen
            "mandatory": true,
            "proof": [
                "0x78b397657c85e9e30ad9d3d4da
                "0xebecc5b3ef32f349f0d333151d
            ],
            "hash": "0xc19208be77ade0a088c22e
        },
        {
            "name": "12Transcript",
            "mandatory": false,
            "proof": [
                "0xa61f759b812a72e4beb8938400
            ],
            "hash": "0x9057aa57118f77921e6e2f
        },
        {
            "name": "13ScienceProfile",
            "mandatory": false,
            "proof": [
                "0x823ff5dcae7962969e62a857cf1
                "0xebecc5b3ef32f349f0d333151d
            ],
            "hash": "0xc4e82efb3f300e3ea60739
        }
    ],
    "issuer": {
        "ethereumAccount": "0x64A695469E5959E
        "ethereumAccountSignature": "mVtSbRYE
        "issuerCertificateChain": "-----BEGIN
    }
}
```

```
{
    "issuedOn": "2021-09-28T16:19:20.144Z",
    "transactionHash": "0xb79cd0cac973c64c3b
    "contractAddress": "0xc61bA3B96848b1F24d5
    "credentialID": "MIT01",
    "sections": [
        {
            "name": "11CertificateOfAchieveme
            "mandatory": true,
            "proof": [
                "0x78b397657c85e9e30ad9d3d4da
                "0xebecc5b3ef32f349f0d333151d
            ],
            "hash": "0xc19208be77ade0a088c22e
        },
        {
            "name": "12Transcript",
            "mandatory": false,
            "proof": [
                "0xa61f759b812a72e4beb893840
            ],
            "hash": "0x9057aa57118f77921e6e2f
        }
    ],
    "issuer": {
        "ethereumAccount": "0x64A695469E5959E
        "ethereumAccountSignature": "mVtSbRYE
        "issuerCertificateChain": "-----BEGI
    }
}
```

Fig. 5. Two sample receipts of a credential. The receipt on the right is created by removing the optional component *ScienceProfile* from the receipt on the left. Both of receipts are valid.

3.5 Revoking Credentials

Like other credential management systems, the `IU-SmartCert` allows issuers to revoke issued credentials (Requirement **R4**).

While in `IU-SmartCert` learners can select components to present to relying parties, they always have to present the mandatory component of the credential otherwise the credential is invalid. With that structure, issuers to revoke an issued credential by marking its mandatory component as revoked.

In detail, we store in the smart contract that manages the issued credentials a list of revoked components along with the reason for their revocation. And only the owner of the smart contract, i.e. the issuer of the credential, can add a record to that revocation list. Later, any relying party can check the status of a credential by checking its mandatory component against the revocation list.

In Fig. 3 the data and the functions for the revocation are the `revocationList`, the `revoke`, and the `isValid` functions.

3.6 Verifying a Credential

An employer or any relying party can verify a credential issued by `IU-SmartCert` from the files, the receipt and the Ethereum network without contacting the issuing institution.

The credential verifying process includes the following steps:

1. Check the receipt information and the issuer's information
   - The issuer's X.509 certificate
   - The signature of the Ethereum account used to issue the credential.
   - Validity of the owner of the smart contract on Ethereum.
   - The name of the issuer in the smart contract and the name on the X.509 certificate.
2. Check the integrity of all components of the credential. This step checks the number, the type and the hash value of the files against the value stored on the receipt.
3. Check the validity of the components in the certificate. This step checks the hash value and proofs to confirm whether the credential belongs to the merkle tree whose root node is published on the Ethereum blockchain.
4. Check whether the certificate is revoked. This step checks if the mandatory component of the credential in the list of revoked credentials stored in the smart contract on the Ethereum blockchain.

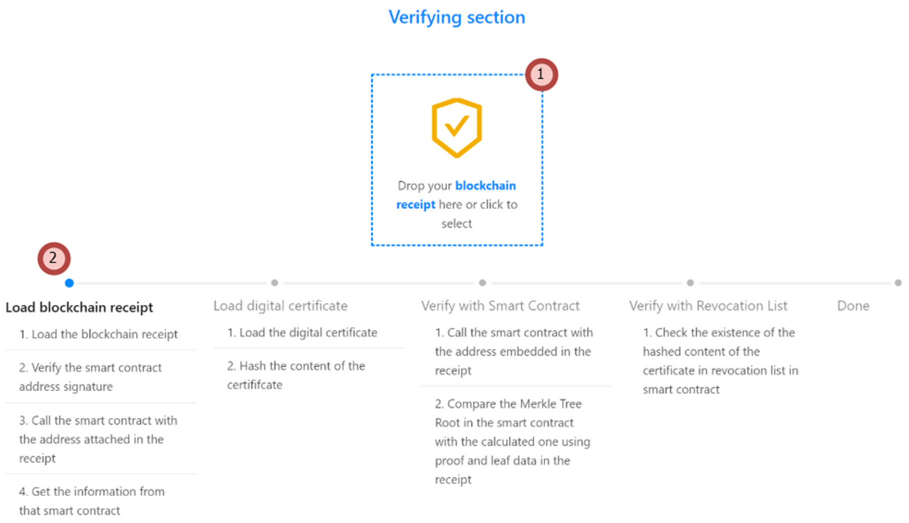
The above procedure is implemented in `IU-SmartCert` with a progress bar for each step as shown in Fig. 6.

On the other hand, relying parties can perform the verifying procedure on their own. Once received a credential and its receipt, a relying party can extract all the data to verify a credential without the involvement of the issuer. Then with a connection to the Ethereum blockchain and any publicly available tools to verify a digital signature and a X.509 certificate, the relying party can finish the verification procedure. That helps us fulfill the requirements **R2** and **R5**.

4 Security Analysis

In this section, we analyse the security aspects of our digital credential management system based on the STRIDE threat model [9]. This model specifies six security risks in an information system, with STRIDE as the initials for those types of risks, including: Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service, and Elevation of privilege. Other security features like authenticity, originality are also analysed.

In the digital credential management system, there are the following entities: credential issuer, credential holder/credential owner, and credential verifier. We assume that there are attackers who are attempting to breach the security of this system in order to gain information or to forge credentials.



Verifying section

Drop your **blockchain receipt** here or click to select

Load blockchain receipt
1. Load the blockchain receipt
2. Verify the smart contract address signature
3. Call the smart contract with the address attached in the receipt
4. Get the information from that smart contract

Load digital certificate
1. Load the digital certificate
2. Hash the content of the certififcate

Verify with Smart Contract
1. Call the smart contract with the address embedded in the receipt
2. Compare the Merkle Tree Root in the smart contract with the calculated one using proof and leaf data in the receipt

Verify with Revocation List
1. Check the existence of the hashed content of the certificate in revocation list in smart contract

Done

Fig. 6. User interface of `IU-SmartCert` for the verification procedure.

4.1 Authenticity and Protection Against Spoofing Identity

Considering the situation where an attacker $\mathcal{A}$ wants to impersonate a credential issuer $\mathcal{CI}_i$ to issue fake credentials, the attacker $\mathcal{A}$ needs to possess the credential issuer's private key to create a valid transaction in order to store fake credentials into the blockchain. Assume that a private key protection mechanism is already in place, $\mathcal{A}$ cannot impersonate the credential issuer $\mathcal{CI}_i$ in this way. Another way is to forge the account of the credential issuer. However, in the `IU-SmartCert` system, credential issuer accounts are verifiable via X.509 certificates. Thus, forging credential issuer accounts is equivalent to creating fake X.509 certificates; this is infeasible.

Since it is impossible to forge valid credentials, the proposed `IU-SmartCert` system ensures the authenticity of its generated credentials. Any credential in this system is verifiable to prove its authenticity by hashing the credential together with other information given in the user's receipt to compute a Merkle tree's root and comparing it against the value stored in the blockchain. Because the hash function in this scheme is secure, it is infeasible to forge credentials that could yield the same hash value as the corresponding Merkel tree's root stored in the blockchain.

4.2 Integrity of Users's Credentials (Protection Against Tampering)

In the proposed method, users' credentials are stored in the receipts provided by `IU-SmartCert` system, and their verification information is kept in Ethereum blockchain. In order to modify a user's credential, an attacker $\mathcal{A}$ needs to modify both the receipt and the record of that receipt in the blockchain to make the modified credential valid. Since the most important property of the blockchain is immutability, it is infeasible to modify any transaction once it had been committed into the blockchain. Therefore, modifying or deleting credential data of the IU-SmartCert system stored on the blockchain is impossible for any attacker. Once digital credentials are issued, they are not legitimately alterable by any entity including the issuer of the digital credentials. The only way that the issuer updates any credential is by revoking the current credential and issuing a new one.

In addition, it is also impossible for an attacker $\mathcal{A}$ to cherry-pick identity components from different credentials to form a valid credential because identity components of a credential of the user $\mathcal{U}_|$ are combined into one input to generate a hash value $\mathcal{H}_|$; this hash value is used as the value of a leaf node of the Merkle tree. Assume that the used hash function is secure, it is infeasible to find another input combination that generates the same hash value $\mathcal{H}_|$.

4.3 Originality and Repudiation

In the `IU-SmartCert` system, a credential issuer $\mathcal{CI}_i$'s blockchain ID is digitally signed using the private key corresponding to the public key in the issuer's X.509 certificate. The X.509 certificate contains information regarding the issuer, such as the organization field; those information had been verified by the certificate authority (CA) at registration time. The X.509 certificate of the issuer's organization field is stored in the read-only variable `institute` of the smart contract. Thus, the originality of the credentials can be ensured by verifying the X.509 certificate of the issuer using digital signature verification with issuing CA's public key and comparing the organization name in the X.509 certificate against the one in the smart contract's `institute` variable. Once $\mathcal{CI}_i$ committed a set of credentials into blockchain, it cannot deny issuing those credentials since its identity is tied to a public X.509 certificate and not anonymous. All information posted on the blockchain by the credential issuer $\mathcal{CI}_i$ will always be traced back to the credential issuer, no one could impersonate them as analyzed above.

4.4 Protection Against Information Disclosure

In the `IU-SmartCert` system, the private keys of the digital signature and the Ethereum account must be kept secret. These keys are stored in private places, not on the public blockchain. It is infeasible for any attacker to compute privates from public keys or Ethereum accounts. Therefore, there is no risk of exposing the secret keys on public blockchain.

The issued digital credentials are available in users' receipts which are in plaintext. The verification information is the hash of these credentials and stored in blockchain. Obtaining the blocks in blockchain containing verification information will not reveal users' credentials. Therefore, users' credentials are only disclosed within their discretion.

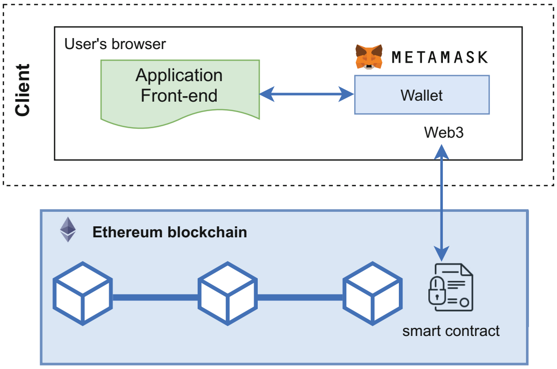


Fig. 7. Architecture of the `IU-SmartCert` system.

4.5 Denial of Service

The `IU-SmartCert` service runs on the Ethereum blockchain platform. The goal of a denial-of-service attack is to prevent digital credentials from verifying or issuing on Ethereum. Since the blockchain is a distributed system, it is not possible to crash or misappropriate the computing power of all the nodes on the blockchain for a denial-of-service attack. Therefore, the `IU-SmartCert` system is immune to a denial-of-service attack.

4.6 Elevation of Privilege

The highest authority in the `IU-SmartCert` system is the credential issuer's authority. This right is guaranteed by a digital signature. Only the owner of the credential issuer's secret key can create a valid digital credential on the blockchain. Therefore, no attacker can gain authority (certificate issuer's authority) on the `IU-SmartCert` system unless they can obtain the private keys of a credential issuer.

With the above STRIDE model analysis and evaluation for the `IU-SmartCert` system, we can conclude that the system meets the desired requirements for security **R6**.

5 Proof of Concept and Experimental Results

In this section, we present a prototype to evaluate the feasibility of our proposal. We first describe a proof of concept of the `IU-SmartCert` and then we measured the cost of sending transactions in the system.

5.1 Implementation

The `IU-SmartCert` system is implemented as two components as illustrated in Fig. 7: a graphical user interface component which is a website and a set of smart contracts on the Ethereum blockchain network that works as storage for proofs of existence of all credentials and the verification algorithm.

The website is implemented with Javascript and ReactJS to provide functions and interfaces for all management procedures of issuers, learners, and replying parties. From the website, the Web3.js library is used to deploy and interact with smart contracts in Ethereum blockchain network directly from the user's browser through a cryptocurrency wallet such as Metamask[1].

The `Cert` smart contract is implemented using Solidity and its instances are deployed to the public Ethereum Ropsten Testnet for the testing purpose.

5.2 Cost of Transactions

The cost of transactions is one of the main concerns of a blockchain-based system. Our `IU-SmartCert` system defines one smart contract named `Cert`, and deploys one instance of the contract for each batch of credentials (described in Sect. 3). Thus, the cost of using blockchain in `IU-SmartCert` is the cost of interactions with the deployed contract which consists of 1 constructor and 3 functions `revoke, isValid` and `verify`.

Table 1. Transaction gas consumption for smart contract functionalities

Function	Task	Transaction gas	Actual cost (*Ether*)	*USD* ($)
Cert.constructor	Deployment	830,264	0.0415132	124.54
Cert.revoke	Transaction	47,996	0.0023998	7.20
Cert.isValid	Call	0	0	0
Cert.verify	Call	0	0	0

The system is analysed by running experiments over the Ethereum Ropsten Testnet. Table 1 depicts the cost of deployment an instance of `Cert` contract and the cost of execution of each function in terms of gas consumption where we set the gas price to 50 $Gwei = 10^{-9}$ *Ether*, and the price of one *Ether* to $3000. The table shows that the deployment of the smart contract is the most expensive transaction, followed by the transaction to the `revoke` function. And the transactions to the remaining functions `isValid` and `verify` cost nothing since these functions only read the data from the smart contract without changing any state in the blockchain.

Although the deployment is the most expensive transaction, it is worth noting that the deployment of the `Cert` smart contract always consumes a fixed amount of gas irrespective of the number of issuing credentials in the batch. It is because the smart contract stores only the root of the Merkle tree representing the whole batch of credentials.

[1] Metamask - https://metamask.io/about.html.

6 Discussion

Our proposed system IU-SmartCert illustrates a new way to use public permissionless blockchain to issue academic credentials with selective disclosure option. The security and the cost analysis show that IU-SmartCert satisfied several security properties and it is practical.

A comparison of our IU-SmartCert with the CreChain [14] is interesting since it is the closest to our work that provides a solution to issue credential with selective disclosure option based on blockchain. The CreChain uses redactable signature technique and therefore requires a procedure to generate an updated signature for each credential redaction of learner. Our IU-SmartCert combines the atomic credential and hashed values approaches to provide selective disclosure option. In contrast to CreChain, in our IU-SmartCert an updated receipt for a selective disclosure is generated by a simple modification of the JSON receipt.

Furthermore, the CreChain builds a tree structure for each credential and cannot issue credentials in batch. It makes the cost of issuance in CreChain proportional to the number of credentials while our IU-SmartCert allows issuing a batch of credentials in one transaction to the blockchain, therefore reduces cost for issuers.

7 Conclusion

We have illustrated a new way to use blockchain to issue academic credentials with a selective disclosure option while maintaining their validity and integrity. Our approach allows learners to proactively prepare and share their credentials for different purposes. Moreover, in the theme of digital transformation in government, our approach could be used for official documents and certificates. For instance, a certificate of ownership with selective disclosure could allow citizens to show and prove the possession of a property without prevailing too much sensitive details. To be efficient in a large-scale use case such as official documents, more research is needed on how to make smaller proofs of existence of the documents with a better data structure like Verkle tree [10], and more advanced data minimization with zero knowledge proofs [4].

Acknowledgement. This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number C2019-28-05.

References

1. Buldas, A., Draheim, D., Nagumo, T., Vedeshin, A.: Blockchain technology: intrinsic technological and socio-economic barriers. In: Dang, T.K., Küng, J., Takizawa, M., Chung, T.M. (eds.) FDSE 2020. LNCS, vol. 12466, pp. 3–27. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63924-2_1
2. CINECA: BESTR: Italian digital credentialing platform. https://bestr.it/. Accessed 23 July 2021

3. Consortium, D.C.: Building the digital credential infrastructure for the future. https://digitalcredentials.mit.edu/. Accessed 23 July 2021
4. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. J. Cryptol. **1**(2), 77–94 (1988)
5. Foundation, C.: Malta, the First Nation State to deploy Blockchain in Education Pilots. https://connectedlearning.edu.mt/malta-first-nation-state-to-deploy-blockchain-in-education/. Accessed 23 July 2021
6. Gallersdörfer, U., Matthes, F.: AuthSC: mind the gap between web and smart contracts. arXiv preprint arXiv:2004.14033 (2020)
7. Grech, A., Camilleri, A.F.: Blockchain in education. Publications Office of the European Union, Luxembourg (2017)
8. Gräther, W., Kolvenbach, S., Ruland, R., Schütte, J., Torres, C., Wendland, F.: Blockchain for education: lifelong learning passport. In: Proceedings of 1st ERCIM Blockchain Workshop 2018. European Society for Socially Embedded Technologies (EUSSET) (2018)
9. Howard, M., Lipner, S.: The security development lifecycle, vol. 8. Microsoft Press Redmond (2006)
10. Kuszmaul, J.: Verkle trees. Verkle Trees, pp. 1–12 (2019)
11. Lab, M.M., Machine, L.: Blockchain Credentials. http://blockcerts.org/. Accessed 23 July 2021
12. Lemoie, K., Soares, L.: Connected impact. Unlocking education and workforce opportunity through blockchain (2020). https://www.acenet.edu/Documents/ACE-Education-Blockchain-Initiative-Connected-Impact-June2020.pdf
13. Merkle, R.C.: A certified digital signature. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_21
14. Mukta, R., Martens, J., Paik, H.Y., Lu, Q., Kanhere, S.S.: Blockchain-based verifiable credential sharing with selective disclosure. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 959–966. IEEE (2020)
15. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. Decentralized Business Review, p. 21260 (2008)
16. Nguyen, B.M., Dao, T.C., Do, B.L.: Towards a blockchain-based certificate authentication system in Vietnam. Peer J. Comput. Sci. **6**, e266 (2020)
17. Nguyen, D.H., Nguyen-Duc, D.N., Huynh-Tuong, N., Pham, H.A.: CVSS: a blockchainized certificate verifying support system. In: Proceedings of the Ninth International Symposium on Information and Communication Technology, pp. 436–442 (2018)
18. Sharples, M., Domingue, J.: The blockchain and kudos: a distributed system for educational record, reputation and reward. In: Verbert, K., Sharples, M., Klobučar, T. (eds.) EC-TEL 2016. LNCS, vol. 9891, pp. 490–496. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45153-4_48
19. Turkanovic, M., Holbl, M., Kosic, K., Hericko, M., Kamisalic, A.: EduCTX: a blockchain-based higher education credit platform. IEEE Access **6**, 5112–5127 (2018)
20. Wood, G., et al.: Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper **151**(2014), 1–32 (2014)