Electronic Certificate Sharing Scheme with Searchable Attribute-Based Encryption on Blockchain

Xingjian Li, Minsheng Tan*
School of Computer, University of South China, Hengyang 421001, China

*417939964@qq.com

Abstract. With the promotion of "Internet +" and smart cities, as an important data voucher, electronic certificate brings great convenience to people's life. However, the phenomenon of "data island" based on the electronic certificate database of each city hinders the sharing of an electronic certificate. In addition, problems such as centralization of data storage, poor security, and tamper-proof modification are common in electronic certificate libraries. To solve these problems, we present a blockchain-based electronic certificate sharing scheme, which uses the InterPlanetary File System (IPFS) and Ethereum smart contract to achieve secure storage and data sharing of electronic certificate by taking advantage of the non-tampering and decentralization of blockchain. At the same time, we achieve fine-grained access control by using attribute-based encryption of ciphertext policy and assigning attribute keys to users without affecting retrieval efficiency. Finally, through the simulation and performance analysis of the Ethereum test network, the analysis results show that our scheme is effective and feasible.

Keywords: Electronic Certificate, Blockchain, Smart Contract, IPFS, Attribute-based encryption

1. Introduction
With the comprehensive advancement of the "Internet +" construction work, more and more provinces and cities have also begun to implement electronic certification systems. As electronic data with legal and administrative effects, electronic certificate liberates the burden of keeping and carrying the certificate owner, and can also provide accurate and comprehensive information for the user. Nowadays, it has increasingly become the main electronic voucher for market entities and citizens' activities and is also an important basic data to support government service operation. It not only makes people's life more convenient but also improves the government's working efficiency. However, the existing electronic certificate database system brings convenience to people's lives, it also has the following main problems:
- At present, most of the current electronic certificate database is centered on various regions or organizations, and independent storage of data, which causes the phenomenon of "data island", and reveals the lack of safe and reliable sharing channel between the databases;
- As electronic information data, an electronic certificate is stored in the form of bit bytes. In the process of data transmission and storage, the authenticity and integrity of data are difficult to be guaranteed and the information is easy to be tampered with by criminals.
The current electronic certificate database mainly uses centralized storage. Once a single point of failure occurs, the entire electronic certificate system will be unavailable.

The current electronic certificate data protection scheme is mainly realized through digital signatures, data watermarking, and other technologies. Because of the lack of reasonable Access control rules, any users can obtain the original data by using the public key of the issuing party, which threatens users’ privacy. What’s more, using other encryption methods to alleviate this problem needs very complex secret retrieval that is difficult to realize.

In recent years, due to the decentralized distributed storage and anti-tampering characteristics of the blockchain[1]-[3], the application of the blockchain has gradually extended from the initial financial field to other application fields, and data protection and sharing in various fields are emerging. For example, Elisa et al. [4] proposed a blockchain-based government affairs system that can increase the trust of the public sector while ensuring information security and privacy. Zhang A et al. [5] proposed a blockchain-based electronic health record sharing system by building a private chain and a consortium chain, which effectively realized access control, privacy protection, and secure search; Tan HB et al. [6] proposed a blockchain-based archive data protection and sharing method through the combination of smart contracts, InterPlanetary File System (IPFS), and public chain alliance chains, and developed an archive data protection and sharing system.

In 2007, Bethencourt et al. [7] proposed a ciphertext policy attribute-based encryption mechanism. Currently, attribute-based encryption (ABE) is the best way to implement access control[8-10]. The access policy was formulated by the data owner to achieve fine-grained access control. Yan xixi et al. [11] proposed an attribute based encryption scheme based on blockchain and supporting verification, which solved the problem of incorrect search results returned under the cloud server. Wang et al. [12] proposed an attribute based encryption scheme based on the Ethereum smart contract. In this scheme, the data owner has the ability to distribute keys for data users, which eliminates the phenomenon of key abuse and ensures the security of private data.

The main contributions of this paper are as follows:

(i). Using IPFS as the storage platform and blockchain records the whole process of data storage and search. We use the characteristics of the blockchain to ensure that the information is authentic and immutable, and the data is credible and traceable.

(ii). We set an encryption key index of the electronic certificate which is stored on the Ethereum blockchain, and use the smart contract of Ethereum to realize the key search.

(iii). The issuing party obtains the key through the personal attributes of the owner to encrypt the data and sets the access policy for the encryption key. The owner can generate the key according to its own attributes to decrypt each data, and the third-party licensed user passes The own attribute key satisfies the access policy to obtain the decryption key, realizing fine-grained access control.

(iv). We designed an attribute-based encryption scheme to achieve access control to shared electronic certificates. Certification authority obtains the key through the personal attributes of the Certificate Owner to encrypt the data and sets the access policy for the encryption key. The owner can generate the key according to its own attributes to decrypt each data. The certificate user obtains the decryption key by using their own specific attribute keys to realize fine-grained access control.

2. System model

The scheme can be divided into two stages: data storage and data search. The figure shows the process of data storage, and the figure shows the process of data search.
There are six entities in our model: Key Manager (KM), Certification Authority, Certificate Owner, Certificate User, Ethereum platform, and IPFS. First, KM generates system parameters and deploys smart contracts on the Ethereum platform, assigns attribute keys to the certificate users, and authorizes the certification authority. Only the certification authority can upload the smart contract index. The certification authority uses IPFS for ciphertext storage and uploads Ethereum transactions to ensure data credibility. After the index is uploaded by the IPFS address and Ethereum transaction ID, the certificate owner and the certificate user can use keywords to search, and the certificate owner has the key that can be directly decrypted. According to its own attributes, the certificate user can only obtain the key if it meets the key access strategy, so as to decrypt the shared electronic certificate data.

3. Concrete construction

- \( \lambda \): Input security parameter \( \lambda \); let \( G_1 \) and \( G_2 \) be two multiplicative cyclic groups with generators \( p \), and \( g_1, g_2 \) are two generators of \( G_1 \). Let \( \mathbb{Z} \) be an admissible bilinear map. The system randomly selects \( Z_\mathbb{Z} \), computes \( g_2^\alpha g_1^\beta \), Select two hash functions \( H_1: \{0,1\}^* \rightarrow Z_p^* \), \( H_2: \{0,1\}^* \rightarrow Z_p^* \), Km defines a pseudo-random function \( (\text{Enc}, \text{Dec}) \) is a symmetric encryption algorithm. The system parameters \( \lambda \), the master secret key keeps secret.

- \( \text{sk} \): KM chooses \( Z_p^* \), and computes \( g_1^\alpha, g_2^\beta \). For each attribute \( a \in S \), randomly chooses \( \alpha \) computes \( X_\alpha = g_1^\alpha, Y_\alpha = g_2^\beta \). Lastly, the attribute authority sends secret key \( \text{sk} \) to the user.

- \( \text{sk} \): The certificate issuer selects the keywords \( \text{sk} \) from the original data, takes sha256 as the key K by using the relevant data \( k_i \) of the certificate owner (such as the admission card number), and then encrypts the original data D of the certificate by using the key K through the encryption algorithm to get \( \text{ciphertext} \), then calculate \( h_{hash} = \text{SHA(D)} \).

- The certificate issuer chooses a tree as the access structure, randomly chooses secret value \( \text{sk} \), for each leaf node in the access structure \( \Gamma \) to...
obtain a secret value and calculates $e^a, L_1 = KH_2(e(g_1, g_1^s)^t)$. The encryption algorithm output is expressed as:

$$\text{Where } s \text{ is the secret value and } y \text{ is the attribute value of leaf node } y.$$ Then, the certificate issuer stores $h_{location}$ in the IPFS, and the IPFS returns a hash address $h_{location}$ about $v$. Randomly chooses $S_{id}$, calculate $\text{token} = F(k_w|\tilde{w}_2|\cdots|\tilde{w}_m)$, and then use token as a parameter to call the smart contract.

- **IndexGen**: The certificate issuer selects $SK_e$ (e.g., certificate name) as the search key, computes $(\tilde{w})$, then stores $(T_n)$ to smart contract with key value pairs, the key is $w$, and the value is $(ID, h_{location})$.
- **Search**: According to the token of the owner or the user, the smart contract returns the result $= (S_{id}, h_{location})$ to the owner or user according to the key value pair, where $S_{id}$ is the ID set satisfied and the $h_{location}$ set satisfied by $S_{location}$.
- **Decrypt**: The decryption phase is divided into two situations:

  (i). The first is decryption by the owner. After receiving the result returned by the smart contract, the owner downloads it from IPFS to $(CT, C)$ according to the $h_{location}$ in the result. Because the owner holds his own relevant data (such as admission card number) to get the encryption key, he can directly decrypt $\text{token}$.

  (ii). The second is decryption by the certification user, The certification party obtains $CT$ and ciphertext $C$ from IPFS according to $h_{location}$. According to the obtained $CT$ and the user's own attribute key $SK$, according to the reference[7], it can be concluded that if the attribute key satisfies the access policy. Then the following formula can be obtained:

$$\text{Then the encryption key } K \text{ can be recovered:}$$

$$\text{Then use the encryption key } K \text{ to calculate to recover the original data.}$$

In order to verify whether the original data has been tampered with, the owner or the user of the certificate finds the $c$ in the Ethereum transaction according to the ID, According to $c$, calculate $h_\tilde{D} = SHA256(\tilde{D})$ and $h'' = h_{location}$, and then compare whether $h_\tilde{D}$ is equal to $h_0$ and whether $h''$ is equal to $h'$. If $h_\tilde{D} = h_0, h'' = h'$, then the data has not been tampered with.

4. **Construction of smart contract**

In this section, we mainly introduce the related interfaces and algorithm logic of the smart contract used in this article. We use Ethereum as the operating platform of the smart contract. Ethereum smart contracts are written and developed in solidity language. The index smart contract used in this article includes four interfaces (addIssuer, removeIssuer, addIndex, search).

(i). **addIssuer**: This interface only allows contract creator km to call, only KM can update the issuerlist variable.
(ii). removeissuer (oldissueraddress): this interface only allows contract creator km to call, only KM can update the issuers list variable.

(iii). addindex (encrypted keyword, ID, hlocation): the certificate issuing party can upload encrypted information through this interface. According to the structure described above, the certificate issuing party selects keywords from the file, builds the corresponding encrypted keyword index, and stores it in the smart contract. The first parameter of the function represents the encrypted keyword, the second parameter represents the transaction ID, and the third parameter represents the IPFS file address.

(iv). Search (token): this interface can be called by any user. According to the above description, when a user (owner or user) wants to search the data he needs, he first generates a search trap according to the required keywords, uploads the trapdoor to the smart contract for data retrieval, and the smart contract finally returns the matching result.

Among the four interfaces mentioned above, three of them (addissuer, removeissuer, addindex) need to change the value of the internal variable of the smart contract, while the search interface is identified by the keyword view, which means that the process will not change the state of the smart contract, so it will not be recorded on the blockchain.

5. Performance analysis

In this section, we evaluate the actual performance of the proposed scheme. Because this scheme combines blockchain and attribute-based encryption, we prepare two different simulation environments. In ethereum platform, our smart contract uses the solidity language, the account uses the Ethereum wallet Metamask, and the smart contract of our solution is deployed on the official Ethereum test network Kovan to store encrypted indexes and searches. When we conducted the experiment, the gasoline price was set to 9Gwei, where 1Gwei = 10^9 wei = 10^-9 ether. We implemented it on Ubuntu 18.04.5 LTS virtual machine, with 4G memory, Intel® Core™ i5-8400 CPU @ 2.80GHz × 4, calling PBC-0.5.14 library and Charm-Crypto- 0.43 Library. In this scheme, the two hash functions are based on SHA-256, the pseudo-random algorithm F uses HMAC-SHA256, and the symmetric encryption algorithm uses AES-256.

The contract was deployed on the Kovan Testnet of Ethereum with the following address:

- **KM account address**: 0x0d85AC9684629feaf38CB1310c0aDcb033A60531
- **Issuer account address**: 0x29FECF506B0B57b238efea426c79a56CCf1b9fc
- **Contract address**: 0x99d8fe97e7e7a3ef924288f1c217b210d6efa02666f4f419e50a41946f5e6ed

Using these addresses, the transactions can be seen at https://kovan.etherscan.io/

5.1. Ethereum platform

The cost of this scheme in Ethereum is mainly derived from smart contracts. The gas cost results of smart contracts are as follows, in the cost of contract create, adduser, and remove, KM only needs to perform contract creation once to initialize the contract, which consumes 918608 gas. When the issuing party needs to obtain authorization, call the addIssuer interface, or when the authority of the issuing party needs to be revoked, call the removeIssuer interface. The cost of these two calling interfaces is 44828 gas and 14925 gas.

| Function    | Gas Used | Actual Cost(ETH) |
|-------------|----------|------------------|
| contract create | 918608   | 0.008267         |
| addIssuer    | 44828    | 0.000403         |
| removeIssuer | 14925    | 0.000134         |

**Tab.1 Smart contract cost (gasprice=9 Gwei)**

Then, we have 5 different numbers of files 1, 5, 10, 15, 20 to build an encrypted index. Tab.2 shows the gas overhead required for addindex under different file numbers, addindex overhead
increases linearly with the number of tasks, adding five encrypted indexes costs about 432038 gas, although calling the search interface will not incur costs, but we will also calculate the gas cost of different numbers of matching operations to evaluate the speed of the search contract.

| The number of indexes | Gas Used  | Actual Cost(ETH) |
|-----------------------|-----------|------------------|
| 1                     | 86410     | 0.000778         |
| 5                     | 432038    | 0.00389          |
| 10                    | 864052    | 0.007779         |
| 15                    | 1296090   | 0.011669         |
| 20                    | 1728128   | 0.015558         |

**Tab.2** Smart contract cost test under different number of indexes (gasprice=9 Gwei)

5.2. *Attribute-based encryption*

In this part, the performance and efficiency of the proposed scheme are analyzed through numerical simulation experiments and compared with the scheme[13 -15].

First of all, we make a theoretical analysis, The coincidence used in this section is introduced as follows: P represents the exponential operation in the cyclic group, E represents the bilinear pair operation, m is the number of keywords, and v is the number of attributes. Tab.3 shows the performance of this scheme and compares it with related schemes.

|               | YAHK   | ABKS   | SSAS   | Ours   |
|---------------|--------|--------|--------|--------|
| Setup         | 3P+E   | 3vp+E  | E      | 3P     |
| Keygen        | (3v+3)P| (2v+3)P| (4v+6)P| (2v+2)P|
| Index         | (2v+2)P+E| (vm+2)+E| (v+1)P| (2v+1)P+E|
| Search        | 2(v+3)E| (v+2)P | (v+2)P| O(1)   |

**Tab.3** Computational cost

It can be seen from Tab.3 that in Keygen, our complexity (2v+2) Is smaller than the other three options. In the index, our solution is better than the YAHK scheme the ABKS scheme, and lower than the SSAS schem. In search, the time complexity of our solution is O(1), which is better than the other three.

Next, we look at the experimental simulation, Since our scheme and the other three schemes are attribute-based encryption, we let |S| ∈ [0,50] in the KeyGen, Index, and Search algorithms to indicate the degree to which it is affected by the number of attributes. According to Fig.3, in addition to the ABKS scheme to reduce the overhead of setup and index and increase keygen, our scheme is better than the other three schemes in keygen and search. It is also better than the other two schemes in setup and index. The simulation shows that our scheme is safe and effective.
Fig.3  The computational costs analysis of the different algorithms in our scheme. (a)KeyGen generations time. (b)Index generations time.

6. Conclusion
Based on Ethereum technology, attribute-based encryption, and IPFS, we propose a new encryption scheme for secure storage and efficient sharing of electronic certificates. By using the attribute-based encryption technology, it provides efficient access control to the electronic certificate database. In addition to the owner himself, only the users whose attribute meets the conditions can view the owner's electronic certificate data. At the same time, we use blockchain technology to store the index, that ensures the security and traceability of data and uses IPFS and smart contract to achieve data sharing and access. In the future work, we will further study the application of electronic certificate in blockchain, and increase the storage capacity and efficiency by combining public chain and alliance chain. In general, our method can achieve efficient and safe protection and sharing of data in the existing electronic certification system with low cost, and provides a new direction for the construction of the electronic certificate library in the "Internet +".

Acknowledgments
This work was supported by Hunan Provincial Innovation Foundation For Postgraduate under Project CX20200935, Project 61403183 of the National Science Foundation of China, Project 18A230 of the Hunan Provincial Education Office Science Research of China. Project 20183350502 and 20191550502 of the Hunan Provincial Finance Office Science Research of China.

References
[1] C. Esposito, A. De Santis, G. Tortora, H. Chang and K. R. Choo 2018 Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy IEEE Cloud Computing 5(1) pp 31-37
[2] J. Wang, M. Li, Y. He, H. Li, K. Xiao and C. Wang 2018 A Blockchain Based Privacy-Preserving Incentive Mechanism in Crowdsensing Applications IEEE Access 6 pp 17545-17556,
[3] A. Dorri, M. Steger, S. S. Kanhere and R. Jurdak 2017 BlockChain: A Distributed Solution to Automotive Security and Privacy, IEEE Communications Magazine 55(12) pp 119-125
[4] Elisa N, Yang L and Chao F 2018 A framework of blockchain-based secure and privacy-preserving E-government system J. Wireless Networks pp 1-11
[5] Zhang A, Lin X 2018 Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain. *Journal of medical systems* **42**(8) p 140

[6] Tan HB, Zhou T, Zhao H, Zhao Z, Wang WD, Zhang ZX, Sheng NZ, Li XF 2019 Archival data protection and sharing method based on blockchain. *Journal of Software* **30**(9) pp 2620–2635 (in Chinese)

[7] Bethencourt J, Sahai A and Waters B 2007 Cipher text-policy attribute-based encryption 2007 IEEE symposium on security and privacy (SP07) (IEEE) pp 321-334

[8] Mhatre S and Nimkar A V 2019 Secure cloud-based federation for EHR using multi-authority ABE. *Progress in Advanced Computing and Intelligent Engineering* (S Singapore; pringer) pp 3-15

[9] Reedy B E and Ramu G A 2016 Secure Framework for Ensuring EHR's Integrity Using Fine-Grained Auditing and CP-ABE. *2016 IEEE 2nd International Conference on Big Data Security on Cloud* (BigDataSecurity) IEEE International Conference on High Performance and Smart Computing (HPSC) IEEE International Conference on Intelligent Data and Security (IDS) (IEEE) pp 85-89

[10] Ramu G and Reddy B E 2015 Secure architecture to manage EHR’s in cloud using SSE and ABE. *J. Health and Technology* **5**(3-4) pp 195-205

[11] Yan X X and Yuan H X 2020 Verifiable attribute-based searchable encryption scheme based on blockchain. *J. Journal on Communications* **41**(02) pp 187-198

[12] Wang S, Zhang Y and Zhang Y 2018 A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *Ieee Access* **6** pp 38437-38450

[13] Yamada S, Attrapadung N and Hanaoka G 2014 A framework and compact constructions for non-monotonic attribute-based encryption International Workshop on Public Key Cryptography (Berlin, Heidelberg; Springer) pp 275-292

[14] W. Sun, S, Yu W, Lou Y, T Hou and H Li 2014 Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications* (Toronto) pp 226-234

[15] J. Sun, X. Yao, S. Wang and Y. Wu 2020 Blockchain-Based Secure Storage and Access Scheme For Electronic Medical Records in IPFS *IEEE Access* **8** pp 59389-59401