BlockCAM: A Blockchain-based Cross-domain Authentication Model

Wentong Wang, Ning Hu
School of Cyberspace Security
National University of Defense Technology
Changsha, China
Email:{wangwentong16, huning}@nudt.edu.cn

Xin Liu*
Department of Computer Engineering & Applied Math
Changsha University
Changsha, China
xin.liu@ccsu.edu.cn

***Abstract*—In a distributed network environment, companies and institutions have their own sharing resource. To prevent unauthorized users to access these shared resources, cross-domain authentication is necessary. For ensuring the safety and efficiency to access resources in different domain, we propose a blockchain-based cross-domain authentication model called BlockCAM and designed the cross-domain authentication protocol. BlockCAM employs consortium blockchain technology to construct a decentralized network with the root Certificate Authorities as the verification nodes. The hash values of the authorized certificates are stored in each block and the verification process only needs to compare whether the hash calculated by the certificate provided by the user is consistent with the hash stored in the blockchain. The authentication process omits the key encryption and decryption overhead. BlockCAM has the characteristics of decentralization, anonymity and temper-resistant. Analyses show that BlockCAM has the advantage over the existing public key infrastructure (PKI) cross-domain authentication schemes at efficiency.**

Keywords— Cross -domain Authentication; Blockchain; Public Key Infrastructure; Digital Certificate

I. INTRODUCTION

With the development of the Internet, the demand for network resources and services is continual increasing. In a distributed network environment where companies and institutions have their own sharing resources. To prevent unauthorized users to access these shared resources, institutions or service providers set up an authentication server to form a relatively independent trust domain for the convenience of managing users. However, a single independent trust domain cannot provide multiple services so that users need to access multiple domains. Therefore, cross-domain authentication problem arises.

There are mainly two cross-domain authentication frameworks: one is authentication frameworks (such as Kerberos[1]) based on the symmetric key management and negotiation, and the other is based on the traditional PKI[2][3][4]. The former has the disadvantage that the symmetric key management and negotiation is very complex. Moreover, it can not effectively deal with the anonymous problem. The latter has the disadvantages of high cost of certificate management, such as certification status checking, certification path construction and certification delivery. It can also cause the network bottleneck of authentication center when under frequently cross-domain accesses. Reference[5][6] proposed an identity-based multi-domain authentication model. It is precondition that the foreign authentication centers are trusted and the key negotiation parameters of each domain must keep same, which causing the limitation that the foreign authentication center can pretend to be a member of local domain. Reference[7] proposed a blockchain-based decentralized PKI authentication system using Certcoin instead of Certificate Authority (CA) to provide key querying and identity binding service. But it has the shortage of privacy leak problem caused by storing the user's identity and public key in the blockchain directly.

In this paper, we propose a blockchain-based cross-domain authentication model called BlcokCA. Built on the existing PKI system, BlockCAM employs blockchain technology to construct a blockchain network in which the root CA serves as verification node. The hash of the authorized certificates and status information are recorded by all CA nodes after passing the consensus algorithm and verification. Since the records in the blockchain are accepted by all root CAs, the cross-domain authentication problem can be quickly resolved by simply judging whether the certificate record in the blockchain is consistent with the certificate provided by its owner.

The **main contributions** of this paper are:

- We propose a blockchain-based cross-authentication model BlockCAM.
- BlockCAM reduces the number of signatures in the authentication process and increases the efficiency of authentication.
- BlockCAM is built on the consortium blockchain and easily scalable.

The rest of this paper is organized as follows: Section Ⅱ introduces the motivation of this paper. Section Ⅲ defines the various components of BlockCAM and details the certificate operations and authentication protocol. In Section Ⅳ we test BlockCAM with other schemes. Section Ⅴ presents some related work of cross-domain authentication. Finally, we conclude the paper in the section Ⅵ.

Supported by NSF 61472438 founding.
*Corresponding author Xin Liu.

978-1-5386-4210-8/18/$31.00 ©2018 IEEE
DOI 10.1109/DSC.2018.00143

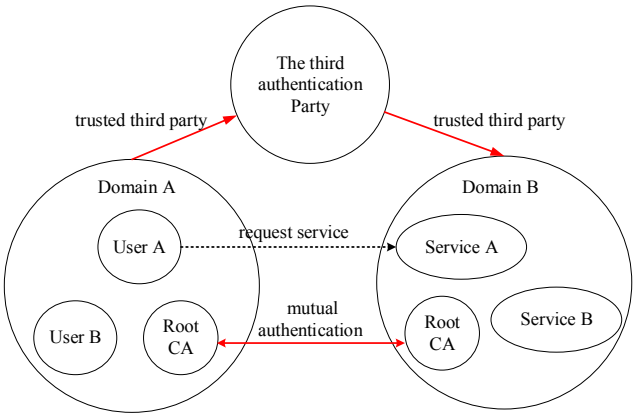

Fig.1. Cross-domain problem model

II. MOTIVATION

A. Problem Description

The problem model is shown as Fig.1. In a distributed network environment where companies and institutions have their own sharing resource, but a independent single trust domain cannot provide multiple services so that users need to access multiple domains. User A in the domain A wants to access the service A in the domain B. To verify user A, one method is that the B-domain authentication server requests user A's root CA certificate to obtain user A's identity certificate. This method has shortages in complex authentication path, frequent signature verification and the difficulty in certificate management, etc. The other method is that Domain A and Domain B are certified by a third-party certification authority. Third-party certification authority can lead to single point of failure and privacy leaks. Therefore, we propose to design a safe, efficient and scalable cross-domain verification model.

B. Problem Analysis

There are no trusted third parties in the above problem model. The traditional cross-domain authentication solution can result in the problem of single point failure, high maintenance costs and complex authentication problems. The new authentication model needs to have the following features:

1) unforgeability : A faked member cannot access the service.

2) Anonymity: Except the authentication server, the user's authentication information is not visible to others.

3) Security: The new model must be enough robust to resist various attacks.

4) Efficency: It has the advantage over traditional authentication schemes for efficiency.

Based on the above analysis, we propose a novel cross-domain verification model BlockCAM.

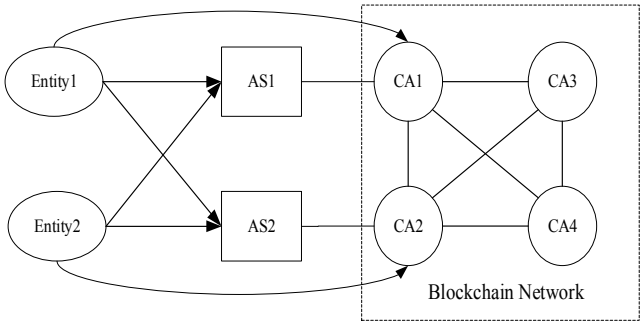

Fig.2. BlockCAM system model

III. BLOCKCAM MODEL

A. System Model

We design a blockchain-based cross-domain authentication model BlockCAM as shown in Fig.2. BlockCAM employs the method of blockchain to achieve the decentralization storage of certificates and solves the cross-domain authentication problem by comparing the blockchain certificate record with the certificate submitted by the user. BlockCAM consists of authentication server nodes, the root CA nodes, blockchain network and entities.

1) AS: These nodes represent the authentication server. AS is responsible for checking the certificates submitted by the user.

2) CA: These nodes represent the root CA. Each root CA is responsible for collecting user's certificate related request

3) Entity: These nodes are represented the users or services in each domain.

4) Blockchain System: BlockCAM is built on top of consortium blockchain. Since consortium blockchain is only open to specific organizations, the authorized root CA joins the consortium chain network to act as a verification node. If a domain no longer has a cross-domain need, or the domain is no longer trusted, the permission to join the consortium blockchain is revoked. Each block consists of a block header and a block body.

a) Block Header: Block header contains the parent blocks hash, timestamp, Merkle root and other information. It's the same as the other blockchain systems.

b) Block Body: Blockchains have limited bandwidth and cannot store much data. Every node on the network has a copy of the data storage on the blockchains. If the whole of certificate is stored on the blockchain, the space and length will grow much more quickly than any other blockchain system so that BlockCAM will be inconvient to use. In our architecture, only certificate hash, certificate ID， the correspond status message are stored in the block body. The certificate hash is used for checking the certificate integrity. The certificate ID is the index of the certificate and the state information ensures the timeliness of the certificate.

B. Certificate operations

Similar to the traditional certificate operation procedures, blockchain-based certificate operations also include application, issuance, update, revocation, and verification processes.

1) Blockchain Certificates(Bcert for short): The certificate owner generates a digital certificate, which can be based on the X.509 standard, only adding an identifier in the extension field so that it can be distinguished from the traditional digital certificates.

2) Symbol Descriptions: To facilitate the description of certificate operations, we define the following symbolic representations:

a) $sig\ (sk, \mu) \rightarrow \sigma$*:* Produce a digital signature σ on the message μ using the secret key sk.

b) $Hash(\mu) \rightarrow \theta$*:* Calculate the hash of the message μ to generate θ.

c) $A \rightarrow B : m$*:* Entity A sends message m to entity B

d) $Func\_Gen(\,) \rightarrow Bcert$: generate the blockchain certificate $Bcert$

e) $ver(pk, \sigma, \mu) \rightarrow b \in \{0,1\}$*:* Verify whether or not σ is a valid signature on μ under the secret key corresponding to the public key pk.

3) Certificate Application

The applicant generates a blockchain certificate locally. And then he or she posts the certificate to the root CA. The application procedure is described in full detail in Fig.3.

Procedure 1 Certificate Application

The user generates a blockchain certificate locally.

- $Func\_Gen(\,) \rightarrow Bcert$

And then he or she posts a message to CA:

$User \rightarrow CA: (ID, \theta, application, info, values = (pk, \sigma))$

to the correspond CA, where

- ID is the certificate identifier,
- θ is the hash of $Bcert$,
- *application* is the type of the message,
- $info$ is the certificate owner's identity verification information,
- pk is the owner's public key,
- σ is the signature of the $Hash(Bcert||info)$ and signed by the owner's private key: $\sigma = sig(sk, Hash(Bcert||info))$.

The signature demonstrates that the certificate owner is able to sign with his or her sk and also ensures that the message is not tampered with.

Fig.3. Certificate application procedure

4) Certificate issuance

The CA in each domain receives the request message and verifies the authenticity and integrity of the certificate. The certificate verification and issuance procedures are described in full detail in Fig.4.

Since all the CAs are in the consortium blockchain, the certificate records that have been stored in the blockchain will be accepted by all CAs. This exactly addresses the issue of mutual trust between different CAs.

5) Certificate update:

When the user needs to update the certificate, he or she generates a new certificate as described in Fig.3 and a new key pairs, but the certificate identifier information is not changed. To judge whether the owner of the original certificate is to update the certificate, we leverage digital signatures here to ensure that a new public key can only be posted by the holder of the secret key corresponding to the old public key. This update transaction will be processed if the signature is verified with the old public key pk^{old}. The update procedure is described in Fig.5.

Procedure 2 Certificate Verification and Issuance

The CA preforms the following verifications:

- Check that the certificate has never previous been registered (by iterating through the blockchain)
- Check that the format of the certificate is correct, and the user's identity information is authentic and reliable,
- Check that $ver(pk, \sigma) = 1$,

If any of these verifications fail, the CA will return "ERROR" and with the error reason message to the user. Otherwise, the CA includes them and preforms the following work:

- Generate a block using the consensus algorithm, where the block stores the hash of Bcert and certificate status information,
- Post the block to the network

The CA checks the correctness of each record in the block. If all the records are correct, the block will be saved to the local blockchain. Otherwise, the block will be discarded.

Fig.4. Certificate verification and issuance procedure

Procedure 3 Certificate Update

Firstly, the certificate applicant generates a new block chain certificate locally.

- $Func\_Gen(\,) \rightarrow Bcert_{new}$,

And then he or she posts a message to CA:

$User \rightarrow CA: \left(ID_{new}, Bcert_{new}, update, values = (pk^{old}, pk^{new}, \sigma_1, \sigma_2)\right)$

to the root CA, where

- ID_{new} is the certificate identifier,
- $Bcert_{new}$ is the new blockchain certificate,
- $upadte$ is type of message,
- pk^{old} is the old public key,
- pk^{new} is the new public key,
- σ_1 is the signature of the Hash$(Bcert_{new}||pk^{new})$ signed by the owner's old private key: $\sigma_1 = sig(sk^{old}, Hash(Bcert_{new}||pk^{new}))$
- σ_2 is the signature of the hash of the $Bcert_{new}$ signed by the owner's new private key:

$$\sigma_2 = sig(sk^{new}, Hash(Bcert_{new}))$$

σ_1 proves that the certificate owner knows the old private key sk^{old} corresponding to the old public key pk^{old} and $Hash(Bcert_{new}||pk^{new})$ ensures the new certificate and pk^{new} information is not tampered with. σ_2 proves that the new certificate is signed by the owner's new public key.

Then, the root performs the verifications and new issuance process as shown in Fig.4. The new block will store the new certificate hash and status information.

Fig.5. Certificate update procedure

6) Certificate revocation

The revocation procedure is described in Fig.6.

The essence of the certificate issuance process is to store the certificate and status information in the blockchain.

Procedure 4 Certificate Revocation

The certificate owner sends certificate revocation message to root CA:

$User \rightarrow CA$:

$\{ID, \theta, revocation, \text{info}, \text{values} = (pk, \sigma)\}$, where

- *ID* is the certificate identifier,
- θ is the hash of $Bcert$,
- *revocation* is the type of the message,
- $info$ is the certificate owner's identity verification information,
- pk is the owner's public key, and
- σ is the signature of the hash of the $Bcert$ and the $info$ signed by owner's private key: $\sigma = sig(sk, Hash(Bcert||info))$.

Then the CA validate the revocation message as the Procedure 2. The other process is similar to the certificate issuance process except that the identity information of the certificate is changed to be revocation.

Fig.6. Certificate revocation procedure

C. CROSS-DOMAIN AUTHENTICATION PROTOCOL

Based on the above model and blockchain certificate, we designed a blockchain certificate-based authentication protocol. The protocol is shown as the Fig.7. The specific process is as follows.

1) $U_A \rightarrow AS_B$: The user in domain A sends request to access domain B.

2) $AS_B \rightarrow U_A: \{N\}$:The authentication server in domain B sends a nounce N to the U_A.

3) $U_A \rightarrow AS_B: \{Cert, sig_{sk}(N), N\}$ *:* U_A sends his or her certificate, the signature of N and nonce N to the AS_B,

4) AS_B uses the $Cert$ and *N* to verify that the $sig_{sk}(N)$ is correct. And then AS_B resolve the certificate and verify the validity of the certificate, including whether it is within its validity period. and the format is correct.

5) AS_B lookup the latest status of the record in the blockchain. If the latest Bcert status information is issuance or update, AS_B calculate the hash of $Cert$.

6) AS_B compares the hash with the hash of $Cert$, if they are same, AS_B authenticates U_A to access domain B.

7) $AS_B \rightarrow U_A: \{CertB, sig(CerB)\}$ *:* AS_B sends his or her certificate $CerB$ to U_A.

8) U_A verify the AS_B certificate. The process (11) to (15) is the reverse authentication process of U_A to AS_B.

If AS_B stores all the blockchain records locally, the process (4) to (8) in Fig.6. can be verified in the AS_B independently. The efficiency of authentication will be increased.

The reverse authentication processes for the user in domain B to domain A is similar to above method. Note that there is more than one authentication server which is just a node in the blockchain network.

D. PROTOCOL ANALYSIS

1) Unforgeability Analysis

BlockCAM stores the hash of the certificate in the blockchain and compares the user's certificate hash with the hash record in the blockchain. The authentication server determines the user accessing the domain according to whether the two hash values are the same. Since the hash function has the following properties:

- Preimage resistance: Given a hash value *h*, it should be hard to find any message m such that $h = Hash(k, m)$.
- Collision resistance: Given two messages m_1 and m_2, it should be hard to find a hash such that $hash(k, m_1) = hash(k, m_2)$, where k is the hash key.
- Second preimage resistance: Given a message m_1, it should be hard to find a different message m_2 such that $hash(k, m_1) = hash(k, m_2)$.

Hash function ensures the integrity of the certificate information. The signature of the hash signed by the owner's private key ensures that the certificate is actually issued by its owner. An attacker cannot forge a certificate unless he owns both the user's private key and certificate because both application and authentication process require the provision of the certificate and private signature.

2) Anonymity Analysis

Only the certificate hash, status and ID are stored in the blockchain. It ensures that user privacy is not leaked. Attackers or inspectors cannot obtain the specific identity information of the certificate owner by retrieving the blockchain records.

3) Security Aanlysis

a) Anti-replay attack: The authentication process uses the query-response handshake mechanism, adding a nonce while passing the message. The nouce is stored in the AS. Before verifying the response message, the AS first validates the nonce. By verifying that the random number is the same as that saved by the original server, it can prevent replay attacks.

b) Anti-DDoS attacks: Built on the blockchain, BlockCAM has decentralized and redundant features. Even if one node fails, other nodes will not be affected.

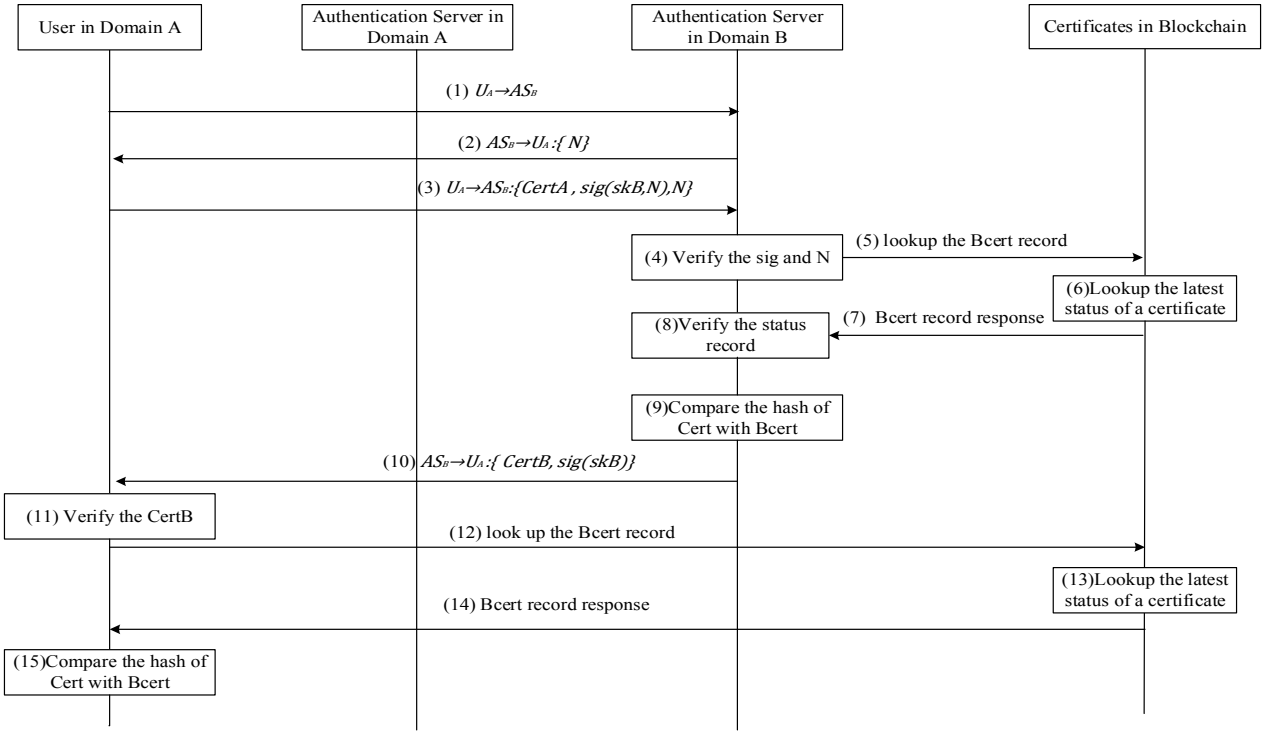

Fig.7. The workflow of cross-domain authentication protocol

4) Bi-directional authentication

In each trust domain, the authentication between the user and the authentication server is achieved by the traditional authentication method. Under the model of consortium blockchain constructed by multi trust domain, the Bcert records of users and authentication servers are stored in the blockchain . The user and the AS can achieve bi-directional authentication through the consortium blockchain.

IV. Experiment

In this section, we report the results of our evaluation of BlockCAM. We compare it with other traditional solutions to verify the improvement and efficiency of this method. The model was developed on the Ethereum consortium blockchain[10]. We configure a multi-node Ethereum consortium network with the published ARM template.

A. Efficiency

We analyze the transmission delay of the BlockCAM scheme and compare its performance with the key pairs management scheme. The comparison is made by evaluating the transmission delay while authenticate the same amount of user information with literature[11] and [12]. The computation overhead is shown in TABLE 1.

To evaluate the efficiency of our system, we used Overlay Weaver[13], an open -source overlay network to construct a simulated network with the number of nodes from 0 to 2500, each time added 500. We simulate the transmission overhead of receiving and dealing with the authentication messages within 30 seconds. The result is as shown in the Fig.8. It shows that BlockCAM processing efficiency is higher than other two schemes at the same condition.

TABLE 1 Comparison Of Computation Overhead

Scheme	encryption/decryption	signature	hash
literature[11]	0	12	4
literature[12]	2	4	10
BlockCAM	0	2	2

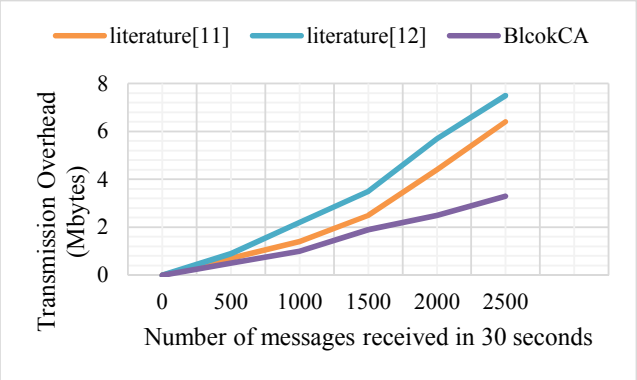

Fig.8. Transmission overhead

B. Analysis

Compared with literature[11], our scheme reduces ten times of digital signatures and verifications and two times of hash operations per time. Compared with literature [12], our scheme reduces 2 times of enciphering and deciphering operations per time. On the same configuration machine, the time spent on RSA with 1024 bits is about half of ECDSA with 192 bits, and the computation time of SHA with 256 bits is about 1/10 of RSA with 1024 bits. So the hash algorithm is much faster than the public key algorithm, and its speed is even more than tens of times. The efficiency and bearing capacity of our scheme for achieving cross-domain authentication is considerable.

V. RELATED WORK

The reference[14] constructs the certification path directly according to the existing PKI structure and topological relation of each domain, but there is a problem of complex authentication path and low authentication efficiency. Reference[15] adopts the authentication scheme of the bridge certificate authority and establishes a bridge CA model that all domains trust. This solution requires each domain to trust this trusted third party, and it is difficult to apply it. At the same time, there is still a problem of how to obtain certificate status information across domains. Reference[16] proposes a PKI authentication system with privacy protection based on the Certcoin[9]. The reference[17] proposes to use the certificate-based PKI authentication system based on the Ethereum blockchain platform to solve the problem of excessive traffic in traditional PKI certificate management and the use of certificate revocation lists and online certificate status protocols.

VI. CONCLUSION

In this paper, we present BlockCAM, a model that can realize cross-domain authentication. We construct a blockchain-based certificate scheme and propose a cross-domain authentication protocol. It can guarantee the security of sharing resources or service in distributed multi-domain network environment. In addition, it has the anonymity that can protect the private information only storing the user's certificate hash in the blockchain. What's more, this model has better security and practicality because it prevent the bottlenecks and the complex certification delivery of traditional PKI model. It is designed based on blockchain and is highly scalable.

REFERENCES

[1] Randy Butler, Von Welch, Douglas Engert. "A national scale authentication infrastructure," IEEE Computer, vol.33(12), pp: 60- 66, 2000.

[2] Jung-San Lee, Chin-Chen Chang, Pen-Yi Chang, et,al. "Anonymous authentication scheme for wireless communications," International Journal of Mobile Communications, vol.16, pp: 590 – 601, 2007.

[3] Liu Wei-hong, Wang Li-bin, Ma Chang-she. "Improved cross-realm C2C-PAKE protocol," Journal of Computer Engineering, vol. 36(19), pp: 162-164, 2011.

[4] Mengbo Hou, Qiuliang Xu, Fengbo Lin. "An Efficient Certificate Revocation and Verification Scheme from Multi-Hashing", Journal of Computers, vol.7(6), pp:1437-1444, 2012.

[5] Peng Hua-xi. "An identity-based authentication model for multi-domain," Journal of Computers, vol. 29(8), pp:1271-1281, 2011.

[6] L Chen, K Harrison, D Soldera. "Smart Applications of multiple trust authorities in pairing based cryptosystems," In Proceedings of Infrastructure Security. Berlin: Springer-Verlag, pp: 260-275, 2002.

[7] Fromknecht C, Velicanu D, Yakoubov S. CertCoin: A NameCoin Based Decentralized Authentication System[J]. Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep, 2014, 6

[8] LUO C Y, HUO SW, XING H Z. "Identity based cross domain authentication scheme in Pervasive Environment," Journal on Communications, vol.32(9), pp:111-115, 2011.

[9] Fromknecht C, Velicanu D, Yakoubov S. CertCoin: A NameCoin Based Decentralized Authentication System. Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep, 2014, 6.

[10] Ethereum consortium blockchain. https://github.com/CatalystCode/ibera-ethereum-consortium-blockchain-network

[11] ZHANG W F, WANG X M, GUO W, et al. Efficient virtual enterprise cross-domain authentication scheme based on Elliptic Curve Cryptosystem. Acta Electronica Sinica, 2014, 42(6): 1095-1102.

[12] LUO C Y, HUO SW, XING H Z. Identity based cross domain authentication scheme in Pervasive Environment. Journal on Communications, 2011, 32(9):111-115.

[13] Kazuyuki Shudo, Yoshio Tanaka, Satoshi Sekiguchi, "Overlay Weaver: An overlay construction toolkit," Computer Communications 31(2): pp.402-412, 2008.

[14] Rouibah K, Ould-Ali S. Dynamic data sharing and security in a collaborative product definition management system[J]. Robotics and Computer-Integrated Manufacturing, 2007, 23(2):217-233.

[15] Millán G L, Pérez M G, Pérez G M, et al. PKI-based trust management in inter-domain scenarios[J]. Computers & Security, 2010, 29(2): 278-290.

[16] Axon L. Privacy-awareness in blockchain-based PKI[J]. Oxford University Research Archive, 2015.

[17] Lewison K, Corella F. Backing Rich Credentials with a Blockchain PKI. 2016.