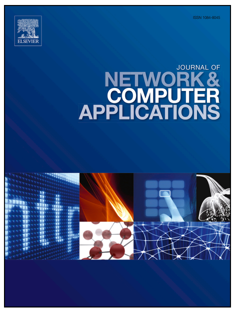
Journal Pre-proof



SBAC: A secure blockchain-based access control framework for information-centric networking

Qiuyun Lyu, Yizhen Qi, Xiaochen Zhang, Huaping Liu, Qiuhua Wang, Ning Zheng

PII: S1084-8045(19)30304-2

DOI: https://doi.org/10.1016/j.jnca.2019.102444

Reference: YJNCA 102444

To appear in: *Journal of Network and Computer Applications*

Received Date: 6 May 2019

Revised Date: 23 July 2019

Accepted Date: 9 September 2019

Please cite this article as: Lyu, Q., Qi, Y., Zhang, X., Liu, H., Wang, Q., Zheng, N., SBAC: A secure blockchain-based access control framework for information-centric networking, *Journal of Network and Computer Applications* (2019), doi: https://doi.org/10.1016/j.jnca.2019.102444.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2019 Published by Elsevier Ltd.

Journal Pre-proof

SBAC: A secure blockchain-based access control framework for information-centric networking

Qiuyun Lyu[a,c], Yizhen Qi[b,*], Xiaochen Zhang[a], Huaping Liu[d], Qiuhua Wang[a], Ning Zheng[a,c,*]

[a]*School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China*
[b]*School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, China*
[c]*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China*
[d]*School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331, USA*

Abstract

The information-centric networking (ICN) has been proposed to meet the increasing demand for efficient content delivery. However, the in-network caching mechanism of ICN makes it hard to provide data security and privacy for content providers (CPs). Although many access control schemes are proposed to improve the security of ICN, there are still some problems unsolved. On the one hand, in spite of the centralized access control schemes are proposed to improve the security of data sharing, the single point of failure issue is inevitable. On the other hand, the decentralized access control schemes allow the content provider to control the key distribution for encrypted content, but it is inefficient when they are applied in hierarchical access, and the audit of content access is ignored. In this paper, we propose a secure blockchain-based access control framework which is called as SBAC, to provide a content provider with the ability of sharing, audit and revocation on his content in a secure way. Specifically, we design a matching-based access control model to achieve hierarchical access, and present a blockchain-based access token mechanism to resist the single point of failure and balance privacy and audit. And Cuckoo filter is introduced to achieve efficient query of access token in verification. Furthermore, our SBAC keeps the characteristic of pervasive caching of ICN. The security analysis and experimental results demonstrate that SBAC is suitable in practice.

Keywords: access control, information-centric networking, blockchain, access audit, security

Journal Pre-proof

1. Introduction

According to the Cisco Visual Networking Index (VNI) forecast, global mobile data traffic will increase seven-fold between 2017 and 2022, nearly four-fifths (79 percent) of the world's mobile data traffic will be video by 2022 (Cisco, 2019). However, due to the lack of natural support for content distribution (Anand et al., 2009; Chen et al., 2015; Fang et al., 2015), the current host-centralized network structure cannot satisfy consumers' increasing need in high efficient access of content today. Thus, a novel information-centric networking (ICN) is proposed (Ahlgren et al., 2012; Jiang et al., 2015; Paul et al., 2011) for the efficient distribution and sharing of information, which provides a promising candidate for the future network (Tourani et al., 2018; Zhang et al., 2015). In the ICN system, in-network caching is introduced to achieve efficient distribution of content. However, this pervasive caching property makes content be copied and spread without any authentication and authorization, hence throwing a big challenge to the access control management.

For better control of the ICN content, the centralized access control schemes (Aiash and Loo, 2015; Fotiou et al., 2012; Ghali et al., 2015; HHamdane and El Fatmi, 2015; Li et al., 2018; Renault et al., 2010; da Silva and Zorzo, 2015) and decentralized access control schemes (Abdallah et al., 2016; Li et al., 2017, 2015; Misra et al., 2019; Xue et al., 2018) are proposed by a few researchers. In these solutions, the centralized access control schemes

[*]Corresponding author
Email addresses: `laqyzj@hdu.edu.cn` (Qiuyun Lyu), `161080037@hdu.edu.cn` (Yizhen Qi), `1097170185@qq.com` (Xiaochen Zhang), `huaping.liu@oregonstate.edu` (Huaping Liu), `wangqiuhua@hdu.edu.cn` (Qiuhua Wang), `nzheng@hdu.edu.cn` (Ning Zheng)

Journal Pre-proof

introduced additional infrastructures to authorize and authenticate the requesting entities. For example, Aiash and Loo (2015) integrated an ID-Based Cryptography (IBC) scheme into the name resolution server (NRS) to deter the masquerading and content poisoning attacks. da Silva and Zorzo (2015) introduced a proxy server and an access structure of attribute-based encryption (ABE) to implement fine grained access control and an immediate revocation. Li et al. (2018) employed a trusted third party and ABE-based naming scheme to hide access policies into a content name for enhancing privacy. However, these schemes will inevitably bring in the single point of failure issues because of the centric operations by a third party. Therefore, several researchers proposed the decentralized access control schemes to overcome the problems from a third party. In a decentralized system, no trusted third party is employed (Azad et al., 2018a,b,c, 2017). For example, Abdallah et al. (2016) proposed a decentralized access control scheme using the Diffie-Hellman (D-H) key exchange paradigm in the publishing/subscribing process of ICN. Li et al. (2015) designed a peer to peer access control scheme and provided two security levels for sharing content. Misra et al. (2019) adopted Shamir's (t+1, n)-threshold secret sharing paradigm together with broadcast encryption to present a secure content delivery framework for ICN without a third party. However, these decentralized access control schemes failed to provide multilevel security for content and the content access audit was impossible since they did not keep an account for each content access.

Inspired by decentralization and tamper-proof features of blockchain (Bonneau et al., 2015; Garay et al., 2015; Nakamoto, 2008) and its innovative application (Chen et al., 2018; Fan et al., 2018; Gai et al., 2019a,b; Maesa et al., 2017; Ouaddah et al., 2016, 2017; Zhu et al., 2019; Zyskind et al., 2015), we intend to utilize blockchain to transfer access tokens and audit access to address the above issues in the decentralized access control schemes, where the access tokens are processed on the blockchain and the content access is off the blockchain. However, when we introduce blockchain into the access control framework for ICN, challenges have been brought in, too. Firstly, how to synchronize the on-blockchain's tokens and the off-blockchain's content will be the major challenge. Secondly, the mandatory traversal operation is needed when we want to query data in the blockchain, which brings the computational cost.

In this paper, we construct a secure blockchain-based access control framework (SBAC) to solve the challenges discussed above for ICN. In SBAC, a matching-based access control model is designed for hierarchial access, and a blockchain-based access token mechanism is presented for transferring access tokens and auditing content access. In summary, we have made the following contributions.

- We propose a secure access control framework based on blockchain technology for ICN, which can guarantee users completely control over published content.
- To deal with the synchronization of "on/off-blockchain" issue, the existing blockchain wallet is extended to not only handle on-blockchain transactions but also manage the off-blockchain access, such as generating/verifing access tokens, content management and ICN cache.
- In order to achieve the high efficiency of access, the matching-based access control model is employed to map one access token to multiple resources, and the Cuckoo filter is introduced to avoid the overhead of mandatory traversal the entire blockchain during token verification.
- We analyze the security of the proposed framework and evaluate the performance of prototype.

The rest of this paper is organized as follows. Section 2 summarizes the recent related works in the related field. In section 3, we state the system model and problem definition, adversarial model, design objectives, and review some blockchain preliminaries. The details of our secure blockchain-based access control framework are presented in section 4. In section 5 and section 6, we analyze security properties and evaluate the performance. Finally, we made a conclusion in section 7.

2. Related work

2.1. Centralized access control schemes for ICN

The centralized access control schemes (Aiash and Loo, 2015; Fotiou et al., 2012; Ghali et al., 2015; HHamdane and El Fatmi, 2015; Li et al., 2018; Renault et al., 2010; da Silva and Zorzo, 2015) are first proposed for secure

Journal Pre-proof

control of content in the ICN and they all introduce a third party to be a centric controller. In 2015, Aiash and Loo (2015) proposed a verifiable access control mechanism for the NetInf (one type of ICN) with the help of a name resolution server (NRS). The mechanism is divided into two processes: registration and authorization. With the help of the authentication and key agreement (AKA) protocol which is based on ID-Based Cryptography (IBC) scheme, the NRS generates a pair of public/private keys and assign them to a publisher or a subscriber in the registration process. During the authorization process, the NRS generates an ObjToken and a SubToken for a publisher and a subscriber, respectively. The publisher uses the ObjToken to publish contents. And the subscriber requests the content with the SubToken where the publisher and NRS verify it. The proposed scheme in (Aiash and Loo, 2015) provides a candidate solution to prevent an unauthenticated node from publishing invalid data. However, the disadvantage of this scheme is that each access request for resources needs the assist from NRS, which introduces a large amount of communication overhead and a single point of failure problem.

And also in 2015, da Silva and Zorzo (2015) presented an access control mechanism for the named data network (NDN, one type of ICN), which relies on a proxy server and employs an attribute-based encryption scheme with a privilege of immediate revocation. Before publishing content, a publisher encrypts the content and generates its access policy. The encrypted content is stored in the content routers while the access policy is only stored in the proxy server in the form of encrypted access structure. In the content requesting process, the requester first retrieves the encrypted content from the content router and gets the access policy from the proxy server, then decrypts the content. The requesters' revocation is achieved by converting the access structure with a new proxy server's key. Comparing with Aiash and Loo's scheme, the structural access policy is introduced to implement fine-grained access control in this scheme and the proxy server frees the publisher from verifying the requester for each content access. Besides, efficient requesters' revocation is added to satisfy the practical need. However, it also holds the single point of failure problem since the proxy server taking part in each content access.

In 2018, Li et al. (2018) designed a privacy enhancing naming scheme using attribute-based encryption (ABE) for the ICN access control, where a trusted third party is employed to create attribute ontology for managing the subject attributes and the object attributes. For publishing the content, a publisher generates the access policy according to the attributes defined by the third party and encrypts the content with a random symmetric key, then hides the random key and the access policy in the content name. Only the authorized users can correctly decrypt the content name with their own attributes to fetch the symmetric key and decrypt the content. The proposed scheme in (Li et al., 2018) achieves privacy with hiding the access policy in the content name, and the content name varies according to access policy or random choosing symmetric key. Furthermore, the third party does not participate in the verification process for each resource's access, which relieves the stress from a single point of failure to some extent. However, the users revocation is ignored in the scheme, and the name immutable principle of ICN is broken since one resource has multiple names according to different access policies or keys.

2.2. *Decentralized access control schemes for ICN*

The ICN access control is implemented without a third entity in the decentralized access control schemes (Abdallah et al., 2016; Li et al., 2017, 2015; Misra et al., 2019; Xue et al., 2018). In 2016, Abdallah et al. (2016) embedded the Diffie-Hellman (D-H) protocol in the publishing/subscribing process to realize decentralize access control. The content, content name and content metadata are sent to the ICN by a publisher, while only the content name is published. A subscriber launches the D-H key exchange with the content name, then responds the parameters for key exchanging and metadata computing if the content publisher agrees. With these parameters, the subscriber computes a public parameter and sends it to the ICN with the metadata. Then the ICN verifies the received metadata and returns the encrypted content with the shared secret key. No single point of failure exists in this scheme since each node of the ICN can implement content caching, publishing, verifying and forwarding. The access control is decentralized due to the publishers and the ICN jointly control the content access. However, the cached content in the ICN is in plaintext which is vulnerable to various attacks, such as content leakage, content poisoning (Roy et al., 2019), etc. Furthermore, the proposed scheme in (Abdallah et al., 2016) is only applicable to the content for reading.

Li et al. (2015) designed a lightweight integrity verification architecture to enable a content provider (CP) to control the content access in the named data network (NDN). In this scheme, a CP can classify the content into two types: public or private. If the content is public, the CP generates a public token and sends it to the NDN routers for verifying. If the content is private, then the CP produces two private tokens: one of them is sent to the authorized NDN router for verifying (note: the cached content is in ciphertext); the other one is sent to the user for content access. The

Journal Pre-proof

proposed scheme in (Li et al., 2015) achieves peer to peer access control, where it allows a user to get content directly from a NDN router or the CP without any third party. Compared with the above mentioned schemes, the scheme provides two security levels for sharing content which is better in practical system. However, the scheme suffers the overhead of token synchronization and token storage since two private tokens generated by the content provider are shared with the NDN router and users for each private content. Besides, the revocation of access is not discussed.

In 2018, Xue et al. (2018) proposed a secure, efficient and accountable access control scheme for ICN using group signatures and hash chains without the third party. In this scheme, a content provider adds an access level to the raw content name, and uses broadcast encryption to propagate encrypted content (C, C_1, C_2) to the network. Edge routers authenticate the users'access requests based on group signatures and hash chains. Once the authentication is passed, the user gets (C, C_1, C_2) and decrypts it. Moreover, the content provider can affirm the service amount received from the network and extract feedback information from the signatures and hash chains. Compared with (Abdallah et al., 2016; Li et al., 2015), this scheme achieves user anonymous, unlinkability and multilevel access. Furthermore, their scheme provides an accounting mechanism for CPs, where CPs can get the amount of served requests, content popularity and users' preferences. However, they did not implement the auditing of users' access activities and the revocation of access.

In 2019, Misra et al. (2019) proposed a secure content delivery framework for ICN using Shamir's (t+1, n)-threshold secret sharing together with broadcast encryption without a third party. A content provider encrypts the content using a symmetric key, then broadcasts the encrypted content and key generation materials to the network. Only the authorized users can use these keying materials and their individual keys to decrypt the encrypted content. In this scheme, the content provider sends one share of secret to each authorized requester, and the requester can fetch the content from the ICN router. Their scheme frees the content provider from the online processing for each content access compared with the scheme in (Abdallah et al., 2016) and avoids the synchronization procedure in (Li et al., 2015). In addition, this scheme provides threshold-based revocation for users. However, this scheme does not keep an account of each content access or the history of keying materials updating which makes the auditing impossible, and does not provide multilevel access .

2.3. *Blockchain-based access control schemes*

In 2015, Zyskind et al. (2015) presented a distributed personal data management scheme with blockchain for enhancing privacy. The blockchain is introduced as the automatic access control manager without any trusted third party, and two novel transactions types are used, of which T_{access} is used for the access control management and T_{data} is served as data content. Only the data address is stored in the blockchain, and the data storage is implemented by DHT, which greatly reduces the risk of data leakage. However, this paper does not propose a specific access control model.

In 2017, Maesa et al. (2017) proposed a blockchain-based access control scheme in which a resource owner defines a policy for the resource and stores it in a blockchain in the form of a policy creation transaction (PCT), the resource's policy can be assigned to users as an access right by rights transfer transactions (RTT). In 2018, Fan et al. (2018) put forward a similar blockchain-based access control solution where the encrypted data is uploaded to the cloud and data access policies stored as blockchain transactions. Although all the access policies or access rights are recorded in the blockchain which achieves tamper-proof and is easy to audit, it leads to leakage of access policy since the blockchain is publicly visible. To prevent the leakage of access policies, Ouaddah et al. (2016, 2017) designed an access control model based on access tokens. They implemented the operations (grant, get, delegate and revoke) of tokens in the form of blockchain transactions. However, the proposed scheme allows one token to access one resource. As a result, it does not make full use of the tokens, and brings the challenge of managing tokens when the resources increasing.

3. Problem statement

3.1. *System model and problem definition*

The system model of the proposed blockchain-based access control for ICN is demonstrated in Fig.1, which contains a content requester (CQ), a content provider (CP), content routers (CRs) and blockchain. The CQ can request content from the CP or CRs after he requests and signs access tokens of content from blockchain, as shown in step ①, ④, ⑤ of Fig.1. The CP generates access tokens for the CQ, grants and verifies access tokens by blockchain,

Journal Pre-proof

and grants content when he receives content requests from CRs, as shown in step ②, ③, ⑥, ⑦ of Fig.1. The CRs provide name-based routing, and perform content transmitting and content caching according to caching policy. The details about how to find content and implement content routing in ICN are not the focus of this paper, and the interested readers can refer to (Li et al., 2017, 2015) to get more information. Blockchain is introduced as a distributed ledger and the Proof-of-Stake (PoS) (Kiayias et al., 2017) consensus algorithm is recommended in our blockchain taking the efficiency and cost issues into consideration.

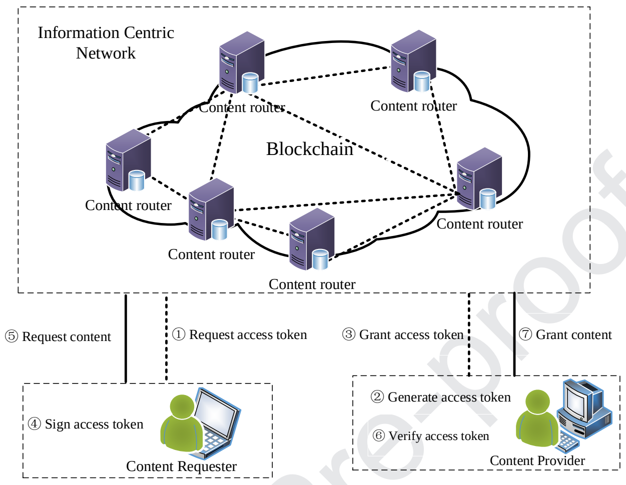


Fig. 1. Blockchain-based access control system model in ICN

According to the system model, we define the problem as follows. Suppose there is an ICN of p content providers $CPs = \{CP_1, CP_2, \cdots, CP_p\}$ and q content requesters $CQs = \{CQ_1, CQ_2, \cdots, CQ_q\}$. Each $CQ_i \in CQs$ wants to access CP_j's content with an access token *AccToken* which is a structured token generated by CP_j based on the CQ_i's attribute score, where an *AccToken* maps a batch of allowed hierarchical resources *RSet*. The match-based access control scheme achieves dynamic and high efficiency since the *AccToken* does not need to update with the resources set *RSet* of adding, deleting, or modifying. The *AccToken* is securely transferred on the decentralized blockchain in a form of transaction and the activity of content access off-blockchain is recorded as an *AccToken*'s access transaction to provide accurate and convenient access auditing for CP_j, while also ensuring each user's (CP_j and CQ_i) identity anonymity.

3.2. *Adversarial model*

The access control framework mainly consists of four components: content providers (CPs), content requesters (CQs), content routers (CRs) and blockchain. Here, we define the adversarial model by discussing the role of the various components in the framework .

- **Content Providers** (CPs). CPs are assumed to be trusted in this framework. We assume that CPs are fair and trustworthy in assigning attribute scores to CQs, and the access content provided by the CPs is also true and reliable.

- **Content Requesters** (CQs). We assume that CQs are potentially malicious in content access. Firstly, the malicious CQs may steal content cached in the content routers of ICN to get unauthorized access or tamper it to perform cache poison attack (Tourani et al., 2018); secondly, the malicious CQs may steal or forge access tokens to illegally access content; thirdly, the malicious CQs may launch the man-in-the-middle attack to intercept or tamper access tokens to make legitimate CQs fail to access content; finally, they may collude with each other to perform the distributed denial of service(DDoS) attack to crash the system.

- **Content Routers** (CRs). We assume that the CRs are honest-but-curious (HBC). It means that CRs perform content caching and forwarding honestly, but CRs are curious about the rich information of the cached content.

Journal Pre-proof

- **Blockchain**. We assume that blockchain is honest-but-curious (HBC). By honest, it means that the blockchain based distributed ledger system performs operation correctly so that it can resist various inherent attacks, e.g. 51% power attack, double spending attack, poisoning data of contract attack; and the addresses of all users (CPs, CQs and CRs) on the blockchain are securely guarded. By curious, it means that any entity involved may try to dig in the access information about CPs and CQs since the recorded transactions on blockchain are open and transparent.

3.3. *Design objectives*

- **Support hierarchical access.** Through providing multiple permitting operations for each resource, and classifying the CQ according to their attributes, we utilize coarse-grained matching together with fine-grained matching to support hierarchical access.

- **Balance privacy with audit.** By means of confining access policy to the owned CP, and designing a specific format for access tokens, we keep the privacy of the access policy. With the blockchain's pseudonym, the CP's identity and the CQs' identities are hidden. At the same time, we formalize the transmission of access tokens and off-blockchain utilization of them as transactions of blockchain for providing the CP with audit.

- **Be compatible with ICN's pervasive caching.** According to the permitting operations of the content, a CR performs different caching policies to be compatible with the ICN's pervasive caching.

3.4. *Blockchain preliminaries*

In order to express clearly, we summarize a few conceptions of blockchain used in this article, such *blockchain*, *smart contracts*, *address*, *transactions*, *wallets*.

Blockchain: In the blockchain-based distributed ledger system, the log is implemented as a series of transactions blocks, where each block contains the hash of the previous block, committing this block as its sole antecedent. It is referred to as the blockchain (Bonneau et al., 2015; Chen et al., 2018). In this paper, blockchain used as a short name of the blockchain-based distributed ledger system. It can not only record the delivering flow of access tokens in the form of transactions but also store the history of off-blockchain content accessing .

Smart Contracts: The concept of smart contract was proposed firstly by Szabo (1997), now it has been extended to the blockchain, and is a common protocol for an editable auto-execution program that is accepted by all nodes on the blockchain. The smart contract in this paper is mainly designed and executed for the transferring and utilization of the access token.

Address: In the blockchain-based distributed ledger system, each user has a key pair: a private key (SK) and a public key (PK). The private key is generated by a random algorithm, the public key is derived from the private key (usually using an elliptic curve algorithm), and the hash value of the public key is the address of the user in the blockchain-based distributed ledger system. In our framework, all users (the CPs, the CQs and the CRs) adopt the same addresses as the ones' in the blockchain-based distributed ledger system.

Transactions: The concept of transaction is introduced in the Bitcoin blockchain for transferring Bitcoin between addresses (Bonneau et al., 2015; Nakamoto, 2008). And it is generalized in the blockchain-based distributed ledger, where the *unspent transaction output* (UTXO) could be any data that is locked with the public key (PK) of the owner. When making a transfer transaction, the user consumes his own UTXO and generates a new UTXO which is locked with the recipient's PK. Similar to (Ouaddah et al., 2016, 2017), the access tokens are transferred between users in our framework. In addition, the access tokens are created by the CP, not the smart contract.

Wallets: Each user has a wallet to manage the user's keys and addresses in the blockchain-based distributed ledger system. In our framework, we take the wallet as an access control manager (ACM), which can be a web-based or mobile application. With it, a CP can not only designate the attribute information required for the access control but also define and label permission operations for the content. The main functions of our wallet are as follows: (1) generate keys and addresses for users. (2) request, generate and verify an access token. (3) generate transfer transactions and access transactions. (4) match content to a requester.

Journal Pre-proof

4. Secure blockchain-based access control framework

The proposed secure blockchain-based access control framework (SBAC) consists of two levels: *matching-based access control model* and *blockchain-based access token mechanism*, as shown in Fig.2.

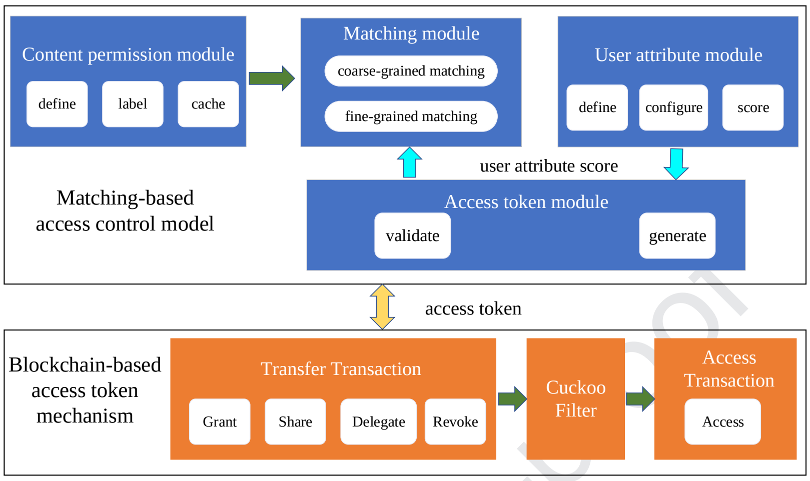


Fig. 2. Blockchain-based access control framework

The *matching-based access control model* implements the hierarchical access, which is composed of the following four components: the *content permission module*, the *user attribute module*, the *matching module* and the *access token module*. The *content permission module* mainly helps a CP to define operations for resources that are allowed remotely access, label the detail operations for each resource, and design caching policy in the ICN. The *user attribute module* allows a CP to define attributes for users, configure attributes for all his potential CQs, and give the current CQ an attribute score. The *matching module* compares the score of content and the one of a CQ to map the allowed content. The *access token module* generates and verifies the access token for a CQ. The *blockchain-based access token mechanism* provides secure access token transfer and reliable access audit, which is divided into two major categories: *transfer transaction* and *access transaction*. The *transfer transaction* deals with the access tokens' granting, sharing, delegating and revoking. And the *access transaction* records the history of content access with the access tokens. For the convenience of description, the notations of the SBAC are illustrated in Table 1.

Table 1: Notations

Notation	Description
CP/CQ	Content provider/content requester in ICN
PK/SK	Public/private key of CP(or CQ)
POp_i	Permitting operations, i is the score of POp_i
$RSet$	Resources set
$RSet_{allow}$	Resources set which is allowed to access by CQ
$MaxS_{op}/MinS_{op}$	The max/min score of POp_i
S_{R_j}	The score of the resource R_j
$AttrSet$	User attribute set
$Type_i$	The i^{th} base-type of user attributes, $i = 1, 2 \ldots, n$
$(Attr_j, Score_j)$	The j^{th} sub-type attribute name and responding score, $j = 1, 2 \ldots, m$
S_{attr_i}	The score of the i^{th} base-type of user attributes
S_{CQ}	The score of the CQ's attributes
φ	The *In-script* used in obtaining an access token from previous transaction
ω	The *Out-script* which gives the condition for obtaining the access token in the transaction
$AccToken_x$	The x type of access token , $x = g,s,d$, where g,s,d are the acronyms of *grant, share, delegate*, respectively.

Journal Pre-proof

4.1. *Matching-based access control model*

The proposed *matching-based access control model* achieves a quick match between a content requester (CQ) and the resources allowed in a content provider (CP), as shown in Fig.3. Specifically, CP configures the permitting operations for content with content permission module at first (refer to step ① of Fig.3), and when receiving the CQ's access token request, CP computes and assigns the attribute score to the CQ and generates a structured access token for him with user attribute module and access token module, respectively (refer to step ②, ③ of Fig.3). Then, once the CQ posts a content access request with an access token, the matching module of CP is responsible for determining to allow or refuse the CQ's content access (refer to step ④ of Fig.3) according to the content's score and the CQ's attribute score. If it is allowed, the CQ can access a bunch of resources off-blockchain of the CP with the access token from blockchain. In such a way, the matching-based access control model fulfills high efficiency in managing access tokens and gains hierarchical access.

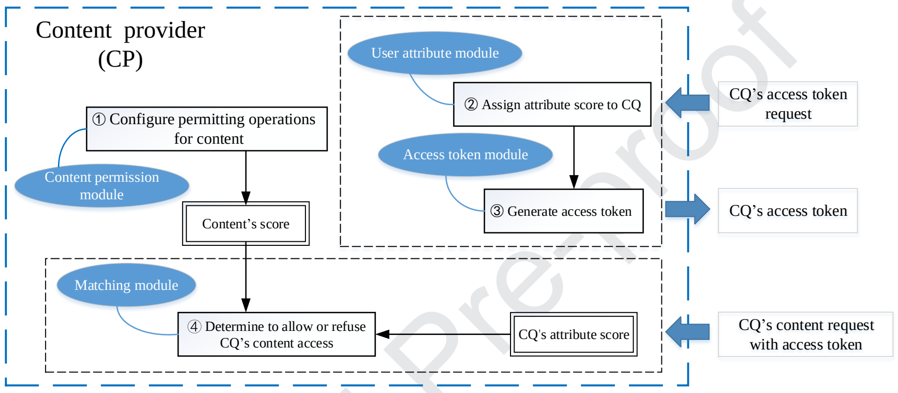

Content provider (CP)
User attribute module
② Assign attribute score to CQ
CQ's access token request
① Configure permitting operations for content
Access token module
Content permission module
③ Generate access token
CQ's access token
Content's score
Matching module
④ Determine to allow or refuse CQ's content access
CQ's attribute score
CQ's content request with access token

Fig. 3. Main steps of matching-based access control model

4.1.1. *Content permission module*

In this module, permitting operations of content are defined firstly, then a CP can label the permitting operations for resources, and caching policies are designed for the ICN. The main object of the module is to provide hierarchical resources set by assigning different permitting operations to each resource.

A. Define permitting operations for content

For clearly specifying the detail operations of each resource, we introduce atomic operation set, see Def.1.

Definition 1. *Atomic Operation Set.*

$$AOp = \{...Modify, Execute, Delegate, Share, Read\} \tag{1}$$

According to the existing remotely access need for resources, we enumerate five atomic operations: *modify*, *execute*, *delegate*, *share*, *read*. The atomic operation *modify* refers to the ability of updating or deleting a resource, the *execute* indicates that a resource can be executed, the *delegate* means that a resource can be delegated to a second-level user, the *share* specifies represents that a resource can be shared by multiple users, and the *read* shows that a resource is readable. For the convenience of description, these five atomic operations can be denoted as M, E, D, S, and R, respectively. Of course, there could be more atomic operations for new needs in applications.

In order to define permitting operations for each resource, we introduce a structured byte based on the above five atomic operations. The structured byte, denoted as POp, is shown in Fig.4, where each bit shows if the atomic operation exists (1 for the corresponding bit) or not (0 for the corresponding bit).

As shown in Fig.4, we can conclude that if the resources are not allowed to remotely access, the value of POp would be 0, and we denote the resources'level as POp_0 or POp_{min}. If the resources are allowed to remotely access in all atomic operations, the value of POp would be 31, and we denote the resources' level as POp_{31} or POp_{max} . We also call the value i as the score of POp_i, and select $MaxS_{op}$ to denote 31, $MinS_{op}$ to denote 0. In the existing 5 atomic

Journal Pre-proof

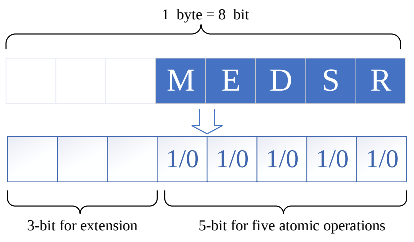


Fig. 4. Structured byte of permitting operations ($POp's$ structure)

operations, we have 17 kinds of permitting operations for resources since the read atomic operation is the prerequisite for other four atomic operations (See Table 2).

B. Label permitting operations for resources and score

When the system starts up, the CP uses one extra byte to label the permitting operations for each resource, according to Def.1. And, all the resources of CP are reorganized to multilevel sets according to their permitting operations, see Def.2.

Definition 2. *Resources Set With Permitting Operations.*

$$RSet = \{(POp_i, \{R_j\}_i)\}, i = 0, 1, 3, 5, ..., n, j = 1, 2, ..., m \tag{2}$$

In the CP's resources $RSet$, the base 2-tuple is $(POp_i, \{R_j\}_i)$, where one POp_i owns a set of resources R_j. Specifically, the i denotes the permitting operations of resources, where the higher i means more permitting operations. In other word, one POp_i indicates one level of resources. An example of 31 resources in different levels are illustrated in Table 2.

Table 2: Permitting operations and caching policy

Level	Permitting operations	Resources set	Caching policy	Level	Permitting operations	Resources set	Caching policy
POp_0	{}	$\{R_1\}$	No-Caching	POp_3	{S,R}	$\{R_{15},R_{19}\}$	All-Caching
POp_1	{R}	$\{R_{13},R_{17}\}$		POp_7	{D,S,R}	{−}	
POp_5	{D,R}	$\{R_{24},R_{31}\}$		POp_{11}	{E,S,R}	$\{R_{26},R_{28}\}$	
POp_9	{E,R}	$\{R_{16},R_{20}\}$		POp_{15}	{E,D,S,R}	{−}	
POp_{13}	{E,D,R}	$\{R_{21},R_{30}\}$		POp_{19}	{M,S,R}	$\{R_{21},R_7\}$	
POp_{17}	{M,R}	$\{R_{18},R_{23}\}$		POp_{23}	{M,D,S,R}	$\{R_6,R_8,R_{22}\}$	
POp_{21}	{M,D,R}	$\{R_{25},R_{27}\}$		POp_{27}	{M,E,S,R}	$\{R_3,R_9\}$	
POp_{25}	{M,E,R}	$\{R_{14},R_{29}\}$		POp_{31}	{M,E,D,S,R}	$\{R_2,R_5,R_{10}\}$	
POp_{29}	{M,E,D,R}	$\{R_4,R_{11}\}$					

In Table 2, the level of R_1 is POp_0, and the level of R_3, R_9 are POp_{27}. And the score of R_1, denoted as S_{R_1}, is 0. Further, we can get S_{R_3} =27, S_{R_9}=27, $S_{R_{13}}$=1, etc. Because the resource R_j with the *share* permitting operation allows a second level of users to share it to others, we generate $sCode$ for it with S_{R_j} and the timestamp (st), where $sCode = H(S_{R_j}||st)$. And the resources with the *delegate* permitting operation, we generate $dCode$ for each of them, where $dCode = H(S_{R_j}||H(S_{R_j}))$.

C. Caching policy for ICN

The ICN's in-network resource caching enables users to efficiently obtain resources. Two levels of resource caching policies are introduced in the SBAC, which are based on whether the atomic operation *share* is included in the permitting operations of resources, as shown in Table 2. If it is not included, the *No-Caching policy* is implemented, where resources are prohibited from being cached by the CRs. Otherwise, the *All-Caching policy* is implemented, where the resource can be cached by the CRs in the ICN.

Journal Pre-proof

4.1.2. *User attribute module*

The user attribute module allows a CP to define and configure the potential CQs' attributes, and to assign the specific attributes to the current CQ. In this way, the CQs are classified into different access levels.

A. Define attributes for users

When a user interacts with others, there will be many types of attributes, such as relational attribute, spatial attribute, time attribute, etc. And each of these types of attributes can be divided into sub-types, for example, the relational attribute can be subdivided into family members, relatives, friends, strangers, organizational partners,$\cdots$, etc. For simplicity, we define a user's attributes as a set.

Definition 3. *User Attribute Set.*

$$AttrSet = \{(Type_i, \overrightarrow{AttrScore_i} \cdot \overrightarrow{M}_i^T)\}, i = 1, 2, \ldots, n \tag{3}$$

In Eq.3, the $Type_i$ represents the i^{th} base-type of the user attribute, the $\overrightarrow{AttrScore_i}$ denotes the sub-types set of the i^{th} base-type which includes 2-tuples of $(Attr_j, Score_j)$, where $Attr_j$ is the j^{th} sub-type attribute name and $Score_j$ denotes the responding score (see Algorithm 1). Generally, for each CQ, the higher score means the closer to the CP, and also means more resources he can get. When a CQ has two or more sub-types attributes in the same base-type, the highest one determines the set of resources. Thus, we only assign one sub-type of base-type (if many, then the highest). So we introduce a one-dimensional matrix $\overrightarrow{M} = (m_1, m_2, ..., m_m)$, where the element is a '0' or '1'. If the CQ is assigned the j^{th} sub-type attribute, then we set a '1' at the j^{th} element and set '0' for other elements.

When we compute an attribute grade for each user, different types of attributes should be taken differently according to the practical application. Therefore the weight $\overrightarrow{W}$ of base-type attributes is introduced to specify the degree of importance in attributes scoring. And for all the base-type attributes (such as k), we have $\sum_{i=1}^{k} w_i = 1$. With this, we can conclude the formula for computing the grade of user attributes, see Def.4.

Definition 4. *User Attribute Score.*

$$S_{user} = \overrightarrow{W} \cdot \overrightarrow{S_{attr}}^T = \sum_i^n w_i * S_{attr_i} = \sum_i^n w_i * (\sum_j^m Score_j * m_j) \tag{4}$$

$$S_{attr_i} = \overrightarrow{Score_i} \cdot \overrightarrow{M}_i^T, i = 1, 2, \ldots, n \tag{5}$$

The score of a user S_{user} can be calculated with $\overrightarrow{W}$ and $\overrightarrow{S_{attr}}$ as Eq.4, where $\overrightarrow{S_{attr}}$ is an array of scores of S_{attr_i} as shown in Eq.5. For scoring the user's attributes, we simplify the $\overrightarrow{AttrScore_i}$ in Def.3 to $\overrightarrow{Score_i}$, which is the score of the i^{th} base-type of the user's attribute and $Score_j$ is the j^{th} element of $\overrightarrow{Score_i}$, where j varies from 1 to m. In the SBAC, only the CQ needs to be assigned the attribute score, so we use the S_{CQ} instead of the S_{user} in the following description.

B. Configure potential users' attributes

When the system starts up, the CP needs to configure the base-types of user attributes for a potential CQ, their corresponding weights and their sub-types, and to designate a grade to each sub-type of attributes according to Algorithm 1.

Algorithm 1 takes $MaxS_{op}$, $Strategy$ and $\{(Type_i, \{Attr_j\}_i)\}$ as input, then outputs the scoring attribute set $\{(Type_i,$ $\{(Attr_j, Score_j)\}_i)\}$, where $MaxS_{op}$ is the maximum score of resource's operation, the $Strategy$ specifies the type of scoring method, and $\{(Type_i, \{Attr_j\}_i)\}$ is the set of CQ's base-type and sub-types of attributes without score. In Algorithm 1, we set minimum score $Score_m = 1$ and the maximum score $Score_1 = MaxS_{op} - 1$ for each base-type of attribute. For the sub-types of attributes' scoring method, there are two generating strategies: arithmetic progression and geometric progression.

Journal Pre-proof

Algorithm 1: Attribute Scoring

```
Input: $MaxS_{op}$, $Strategy$, $\{(Type_i, \{Attr_j\}_i)\}, 1 \le i \le n, 1 \le j \le m$
Output: $\{(Type_i, \{(Attr_j, Score_j)\}_i)\}, 1 \le i \le n, 1 \le j \le m$
1  for $i = 1; i \le n; i++$ do
2      $Score_1 \leftarrow MaxS_{op} - 1$;
3      $m \leftarrow count(\{Attr_j\}_i)$;
4      if Strategy == 1 then // arithmetic progression
5          $comDiff \leftarrow Score_1/(m-1)$ ;
           // comDiff is a common difference of $Score_j$.
6          while $2 \le j \le m-1$ do
7              $Score_j \leftarrow Score_{j-1} - comDiff$;
8              $setAttrScore(\{(Type_i, \{(Attr_j, Score_j)\}_i)\})$;
               // set sub-type attribute score.
9              $j \leftarrow j+1$;
               // increment index j.
10     if $Strategy == 2$ then // geometric progression
11         $ratio \leftarrow 2$ ;// set the initial ratio to 2.
12         $n \leftarrow 1$ ;
13         if $ratio^{m-1} \ge Score_1$ then // enlarge to support a large number of sub-types
14             $Score_1 \leftarrow Score_1 * ratio^{m-1}$ ;
15             $n \leftarrow ratio^{m-1}$ ;
16         else
17             while $(ratio+1)^{m-1} \le Score_1$ do // calculate the ratio.
18                 $ratio \leftarrow ratio + 1$ ;
19         while $2 \le j \le m-1$ do
20             $Score_j = \lfloor Score_{j-1}/ratio \rfloor$;
21             $Score_j \leftarrow Score_j/n$ ;
22             $setAttrScore(\{(Type_i, \{(Attr_j, Score_j)\}_i)\})$;
               // set sub-type attribute score.
23             $j \leftarrow j+1$;
               // increment index j.
24 return $\{(Type_i, \{(Attr_j, Score_j)\}_i)\}$;
```

As mentioned above, a CP has the ability to customize a base-type of attribute's name, weight and its sub-types. For clarity's sake, an example of attribute configuration from an individual CP is illustrated in Table 3, where three base-types of user attributes are configured and the arithmetic progression strategy is adopted in Algorithm 1.

We assume that a CP takes the *Relationship* attribute as the most important one to describe the identity of CQ than the other attributes. Thus its weight is 0.5. And the *Position* attribute possesses more uncertainty compared with *Affiliated institutions* attribute, so the *Position* weight is 0.2, and the weight of *Affiliated institutions* is 0.3, see Table 3. For the sub-types of the *Relationship* attribute, we sort them according to the degree of affinity with the CP. And the sub-types of the *Affiliated institutions* attribute are sorted by the possibility of the data being maliciously used, where the higher score means the lower possibility of data misuse. For the sub-types of the *Position* attribute, we sort them according to the distance, where the farther away the distance is, the more possibility of being impersonated.

C. Assign attributes to requesters and score

When receiving a request from a CQ, the CP needs to assign attributes to the CQ. Specifically, the CP chooses one sub-type of each base-type of attributes according to the requester's information. For example, in the *Relationship* attribute of Table 3, CP can choose family members, or strangers in default for a CQ. After completing attributes assignment, the attribute score of the CQ can be computed according to Def.4. For instance, a friend from the same province posts an access token request to the CP as an individual, and the CP computes the score for him,

Journal Pre-proof

Table 3: An example of user attribute configuration

Base-type of Attribute	Sub-type of Attribute		Weight
	Name	Score	
Relationship	family members	30	0.5
	relatives	22.5	
	friends	15	
	organizational partners	7.5	
	strangers	1	
Affiliated institutions	government	30	0.3
	research institutions	20	
	individuals	10	
	enterprises	1	
Position	face to face	30	0.2
	same community	24	
	same city	18	
	same province	12	
	same country	6	
	same world	1	

see the items with the grey background in Table 3. We can get the *friends* attribute's score is 15, the *individuals* attribute's score is 10, and the *same province* attribute's score is 12. Therefore, we can get the following values from Table 3: $\overrightarrow{Score_1} = (30, 22.5, 15, 7.5, 1)$, $\overrightarrow{Score_2} = (30, 20, 10, 1)$, $\overrightarrow{Score_3} = (30, 24, 18, 12, 6, 1)$, $\overrightarrow{M_1} = (0, 0, 1, 0, 0)$, $\overrightarrow{M_2} = (0, 0, 1, 0)$, $\overrightarrow{M_3} = (0, 0, 0, 1, 0, 0)$, $\overrightarrow{W} = (0.5, 0.3, 0.2)$, and we use Def.4. $\overrightarrow{S_{attr}} = \overrightarrow{Score_i} \cdot \overrightarrow{M_i^T} = (15, 10, 12)$, $S_{CQ} = \overrightarrow{W} \cdot \overrightarrow{S_{attr}}^T = (0, 5, 0, 3, 0, 2)(15, 10, 12)^T = 12.9$, to obtain the total score of CQ's attributes is 12.9.

4.1.3. Access token module

As response to the CQ's requests, an access token is returned to the CQ. The access token is a valid and unique access right identifier which is defined by the CP for the CQ. And we transfer the access token in the form of transactions on the blockchain, see section 4.2.

A. Access token format

The format of an access token, as shown in Table 4, consists of two parts: token header and token payload. The token header includes three fields: a token id, a token type and an encryption algorithm. The token payload contains six fields, including a token data, a source address, a destination address, a generation time, expiration time and an authentication data.

Table 4: Access token format

Access Token Field Name		Symbols	Description
Token Header	Token Id	*tokenId*	Unique identifier for the access token.
	Token Type	typ_x	Can be typ_g,typ_s or typ_d, where *g,s,d* are the acronyms of *grant, share, delegate*, respectively.
	Encryption Algorithm	Enc^x	Enc^x to encrypt $data_{token}$, can be symmetric Enc^1 or asymmetric Enc^2.
Token Payload	Source Address	PK_{CP}	CP's address in the blockchain.
	Destination Address	PK_{CQ}	CQ's address in the blockchain.
	Token Data	$data_{token}$	According to typ_x, $data_{token}$ can be $Enc^1(S_{CQ} \oplus salt)$, $Enc^2(sCode) \parallel H(AccToken_g)$, $Enc^2(dCode) \parallel H(AccToken_g)$
	Generation Time	T_{gen}	Time of generating an access token.
	Expiration Time	T_{exp}	The expiration time of the access token.
	Authentication Data	ϑ	Used to verify the integrity of the token.

Journal Pre-proof

B. Generate access token

There are three types of access tokens can be generated: *grant*, *share* and *delegate*. Taking generating a *grant* type of access token as an example, firstly, a CP randomly chooses a 64-bit number as *tokenId*, sets token type as typ_g and selects a symmetric algorithm Enc^1, such as *AES* or *DES*. Secondly, the CP generates authentication data $\vartheta = H(S_{CQ} \parallel PK_{CP} \parallel PK_{CQ} \parallel T_{gen} \parallel T_{exp} \parallel salt)$, then produces the symmetric encrypting key $symKey = H(\vartheta \parallel SK_{CP})$, and computes $data_{token} = Enc^1_{symKey}(S_{CQ} \oplus salt)$. In a result, the *grant* type of access token $AccToken_g = tokenId \parallel typ_g \parallel data_{token} \parallel PK_{CP} \parallel PK_{CQ} \parallel T_{gen} \parallel T_{exp} \parallel \vartheta$. Note, when the system starts up, it will initialize n salts in an array of strings $[str_0, str_1, ..., str_{n-1}]$, each of them is 32 bytes long. As a result, the chosen $salt = str_{sn}$, where $sn = (T_{exp} - T_{gen}) \mod n$.

If a CQ uses an $AccToken_g$ to fetch resources from the CP, the *sCode* of shareable resources and the *dCode* of delegatable resources are also returned. Then the CQ can generate a $AccToken_s$ or $AccToken_d$ for other CQs. The $AccToken_s$ or $AccToken_d$ generating procedure is the same as $AccToken_g$ except the $data_{token}$ and ϑ. In the $AccToken_s$ and $AccToken_d$, $data_{token} = Enc^2(xCode) \parallel H(AccToken_g)$, where Enc^2 is an asymmetric algorithm, such as *ECC*, and $xCode = sCode\ or\ dCode$ and ϑ is null.

C. Verify access token

For an $AccToken_g$, a CP checks if it is in its valid time or if it is valid on the blockchain. If it is, the CP fetches the ϑ' and generates *symKey* with his private key, where $symKey = H(\vartheta' \parallel SK_{CP})$. Then the CP decrypts $data_{token}$ to get S'_{CQ} with *salt*, where $S'_{CQ} = Dec^1_{symKey}(data_{token}) \oplus salt$. Further, with the S'_{CQ}, PK'_{CP}, PK'_{CQ}, T'_{gen}, T'_{exp} and *salt*, the CP computes $\vartheta = H(S'_{CQ} \parallel PK'_{CP} \parallel PK'_{CQ} \parallel T'_{gen} \parallel T'_{exp} \parallel salt)$ and checks if $\vartheta \overset{?}{=} \vartheta'$ holds. If it is, then the $AccToken_g$ is valid, and the S'_{CQ} is a valid S_{CQ}.

Upon receiving an $AccToken_d$, a CP verifies whether the token is in its valid time and whether it is on the blockchain firstly. If yes, then the CP checks the *dCode* of the $AccToken_d$ is valid or not. If it is, the CP responds the CQ's request. For the $AccToken_s$, any entity of ICN (such as a CR) who caches the resources verifies whether this token is in its valid time and whether it is on the blockchain. If yes, it returns the caching encrypted content to the CQ and only the one who owns the valid *sCode* can decrypt it correctly.

D. Access content in the ICN

For the resources implementing the *No-Caching policy*, the CQ is required to provide the $AccToken_g$ or $AccToken_d$ for the CP to obtain them, in this case, the content router only assumes the content forwarding. For the resources implementing the *All-Caching policy*, the encrypted resources can be cached by any content router when they are forwarded. Any CQ who owns a resource's $AccToken_s$ can obtain the encrypted resource from available neighbor cache and decrypt it with *sCode*.

4.1.4. Matching module

When a CQ requests access to the content with an access token, the CP first computes a current S'_{CQ} for the CQ, and extracts the S_{CQ} from the access token. If $S'_{CQ} \geq S_{CQ}$, the CP takes S_{CQ} to match POp_i to fetch resources from *RSet*, else rejects the CQ. The content matching module is the key to achieving hierarchical access, where the closest even number Z of S_{CQ} is computed at first, and two steps are followed: coarse-grained matching and fine-grained matching. The coarse-grained matching mainly determines the set of resources that are allowed to be accessed for CQ, which is a hierarchical set and called as $RSet_{access}$. The fine-grained matching deals with the detailed permitting operations for the content with the lowest score in $RSet_{access}$, and achieves hierarchical access to the same level of resource based on different S_{CQ}. Finally, CQ obtains the set of allowed access resources $RSet_{allow}$. The pseudocode of requester-content matching is shown in Algorithm 2.

A. Coarse-grained matching

The coarse-grained matching is used to match the available access resources according to the S_{CQ}. Specifically, it first finds the boundary level B of resources, where $B = MaxS_{op} - Z$. And then, it fetches the available resources set $RSet_{access} = \{(POp_B, \{RSet_j\}_B), (POp_{B+2}, \{RSet_j\}_{B+2}), \cdots, (POp_n, \{RSet_j\}_n)\}$, which includes any resource whose score of POp is more than B, see Algorithm 3 for details.

Journal Pre-proof

Algorithm 2: Requester-Content Matching

```
   Input: $S_{CQ}$, RSet
   // $S_{CQ}$ is attributes' score of CQ
   Output: $RSet_{allow}$
   // resources set which is allowed to access by CQ.
1  $Z \leftarrow \lfloor S_{CQ} \rfloor$;
2  if $Z/2 \neq 0 \&\& Z \neq 1$ then // When $Z$ is odd, reduce 1.
3      $Z \leftarrow Z - 1$;
4  $RSet_{access} \leftarrow CGMatch(Z, RSet)$;
   // coarse-grained matching.
5  $RSet_{allow} \leftarrow FGMatch(RSet_{access}, S_{CQ}, Z)$;
   // fine-grained matching.
6  return $RSet_{allow}$;
```

Algorithm 3: *CGMatch*

```
   Input: $Z, RSet$
   // $Z$ is the closest even number no more
      than $S_{CQ}$, $RSet$ is a set of CP's
      resources.
   Output: $RSet_{access}$
   // resources set which is allowed to
      access by CQ.
1  $B \leftarrow MaxS_{op} - Z$;
2  while $B \leq t \leq n$ do
3      $RSet_{access}.put(RSet.get(POp_t))$;
4      $t \leftarrow t + 2$;
5  return $RSet_{access}$;
```

Algorithm 4: *FGMatch*

```
   Input: $RSet_{access}, Z, S_{CQ}$
   Output: $RSet_{allow}$
1  $D \leftarrow S_{CQ} - Z$;
2  $SPOp_B \leftarrow toBinary(2^{\lfloor D \times 10/4 \rfloor + 1} - 1)$;
   // get subset mask of $POp_B$.
3  $POp_{allow} \leftarrow POp_B \& SPOp_B$ // & represents
      binary AND operation
4  $RSet_{allow}.put(POp_{allow}, RSet_{access}.get(POp_B))$;
   // update permitting operation
5  while $B + 2 \leq t \leq n$ do
6      $RSet_{allow}.put(RSet_{access}.get(POp_t))$;
7      $t \leftarrow t + 2$;
8  return $RSet_{allow}$;
```

B. Fine-grained matching

As above, $RSet_{access}$ includes not only the available resources but also their responding permitting operations. However, in order to further differentiate the same user from different environment, or different users whose score are close, the fine-grained matching is introduced. In fine-grained matching, it keeps the same permitting operations as before other than the resources of POp_B. For the resources of POp_B, it fetches the subset of permitting operations according to the difference between S_{CQ} and Z. Finally, the CQ obtains $RSet_{allow}$, see Algorithm 4.

Specifically, if the CQ has the maximum attributes' grade, where $S_{CQ} = MaxS_{op} - 1$, he can get all the CP's resources *RSet*. And if CQ has the minimum attributes' grade, where $S_{CQ} = MinS_{op} - 1$, he can only read the resources of the POp_n level.

4.2. Blockchain-based access token mechanism

Based on blockchain technology, the proposed *access token mechanism* implements secure transmission of access tokens for privacy and accurate record of access activities for access audit. As shown in Fig.2, two kinds of transactions are introduced: the *transfer transaction* and the *access transaction*. The *transfer transaction* is mainly responsible for secure transmission of access tokens, and it can be further divided into four types: *grant, share, delegate, revoke*. The *access transaction* stores the access logs for compliance audit. For improving the efficiency of the access token verifying, Cuckoo filter is introduced.

4.2.1. Transaction format

Generally, each transaction is composed of three parts: identity of the transaction *Tid*, an input array *Tin[]* and an output array *Tout[]* of the transaction. In order to quickly find a transaction record without traversing the entire blockchain, a new field *Transaction Type* is added, which can be denoted as *Ttype*, where *Ttype* $\in$ {*Grant*, *Share*, *Delegate*, *Revoke*, *Access*}. Therefore, the transaction of $AccToken_x$ is defined as T_{Ttype}:

$$T_{Ttype} = (Tid, Ttype, Tin[PK_{CP}, T_{pre}, \varphi], Tout[PK_{CQ}, AccToken_x, \omega]) \tag{6}$$

In Eq.6, T_{pre} represents the last transaction of $AccToken_x$. φ is an *In-script* used in obtaining an access token from previous transaction, $AccToken_x$ is one of three types of access tokens (refer to section 4.1.3), and ω is an Out-script which gives the condition for obtaining the access token in the transaction. For simplicity, Table 5 illustrates a standard transaction.

Table 5: Transaction T_{Ttype}

T_{Ttype}(**in:**T_{pre})	
Transaction Type:	$Ttype$
Access Token:	$AccToken_x$
In-script φ**:**	ϕ or $Sig_{SK_{CP}}(T_{pre})$
Out-script ω $(body, \sigma)$**:**	$Ver_{PK_{CQ}}(body, \sigma)$

Table 6: Grant access token transaction

T_{Grant} (**in:**ϕ)	
Transaction Type:	*Grant*
Access Token:	$AccToken_g$
In-script φ**:**	ϕ
Out-script ω $(body, \sigma)$**:**	$Ver_{PK_{CQ}}(body, \sigma)$

In Table 5, *body* includes T_{pre}, *Ttype*, $AccToken_x$, PK_{CQ} and σ is a signature of CP on T_{Ttype}. If the transaction type *Ttpye* is *Grant*, it would be the original transaction, so the φ is empty, denoted as ϕ.

4.2.2. Transfer transaction of access token

There are four types of access token transfer transactions: *Grant transaction*, *Share transaction*, *Delegate transaction* and *Revoke transaction*.

A. Grant transaction of access token

When a CP responses to the CQ who requests content with his attributes, the CP returns the $AccToken_g$ with *Grant transaction* T_{Grant} on the blockchain, see Table 6. T_{pre}, φ are empty, because there is no previous transaction. The transaction type is *Grant* and the access token is the $AccToken_g$. We can describe the T_{Grant} as the following Eq.7:

$$T_{Grant} = (Tid, Grant, Tin[PK_{CP}, \phi, \phi], Tout[PK_{CQ}, AccToken_g, \omega]) \tag{7}$$

B. Share transaction of access token

The CP uses *sCode* to encrypt the shareable content, and the encrypted shareable content can be arbitrarily cached in the ICN network off blockchain. Any user can decrypt this encrypted shareable content with a sharing token which is transferred in a share transaction. The high level of access token sharing is illustrated in Fig. 5. The content provider *A* grants $AccToken_{g^{AB}}$ to the sharer *B* with the transaction $T_{Grant^{AB}}$. *B* can use this token to access the content in *A*. If *B* satisfies the access rights of a shareable content, *B* can obtain the share code of the content. When *B* wants to share the content with other CQs, he can generate a sharing token $AccToken_{s^{BC}}$ based on the sharing code of the content and a *Share transaction* T_{Share}, as follows:

$$T_{Share} = (Tid, Share, Tin[PK_B, T_{pre}, \varphi], Tout[PK_C, AccToken_{s^{BC}}, \omega]) \tag{8}$$

In Eq.8, T_{pre} is $T_{Grant^{AB}}$.

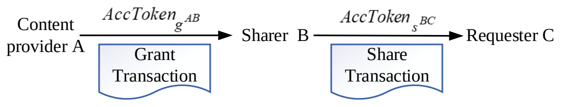


Fig. 5. Sharing process of access token

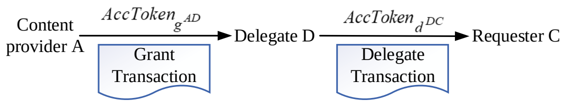


Fig. 6. Delegating process of access token

C. Delegate transaction of access token

The delegate transaction of access token deals with such a situation where a CP has content which can be delegated to a third-party by a CQ. The high level of delegate process is shown in Fig 6. The content provider *A* grants $AccToken_{g^{AD}}$ to a delegate *D* with the transaction $T_{Grant^{AD}}$, then *D* can use the token to access the resources in *A*. If

Journal Pre-proof

user D satisfies the access rights of a delegated content, D can obtain the delegate code of the content. If D delegates a delegated content to requester C, D generates a delegating token $AccToken_{d^{DC}}$ based on the delegation code and then generates *Delegate transaction* $T_{Delegate}$ simultaneously, as follows:

$$T_{Delegate} = (Tid, Delegate, Tin[PK_D, T_{pre}, \varphi], Tout[PK_C, AccToken_{d^{DC}}, \omega]) \tag{9}$$

In Eq.9, T_{pre} is $T_{Grant^{AD}}$.

D. Revoke transaction of access token

The revocation of an access token can be divided into two cases: one is the expiration of access token, the other is that the access token is revoked by the creator. The verifier generates a revoke transaction T_{Revoke} for the expired token. And if a user wants to revoke the issued token explicitly, he will generate the *Revoke transaction* T_{Revoke}, which can be expressed as Eq.10. Note that, since the $AccToken_d$ only can be used once, CP will generate a T_{Revoke} for the token whenever the token is used. In a word, a CP, a CR, a sharer or a delegate all can initiate a T_{Revoke}.

$$T_{Revoke} = (Tid, Revoke, Tin[PK_{user}, T_{pre}, \varphi], Tout[\phi, AccToken_x, \phi]) \tag{10}$$

And in Eq.10, PK_{user} is the public key address of the user who initiated the revocation for the $AccToken_x$. Specifically, in order to ensure that the revoked token is no longer transmitted on the blockchain, the output address and ω of the transaction are set empty, denoted as ϕ.

4.2.3. Access transaction with access token

A CQ posts a request to a CP with the corresponding access token to access content, the CP authenticates him and generates an access transaction while providing the content off blockchain. The details are shown in Fig 7.

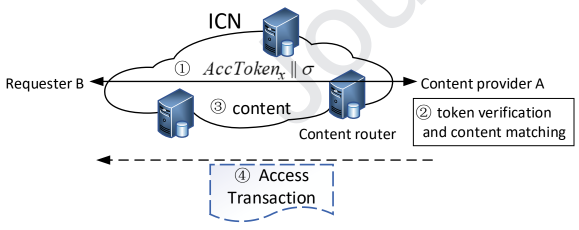


Fig. 7. Access content off blockchain with access token

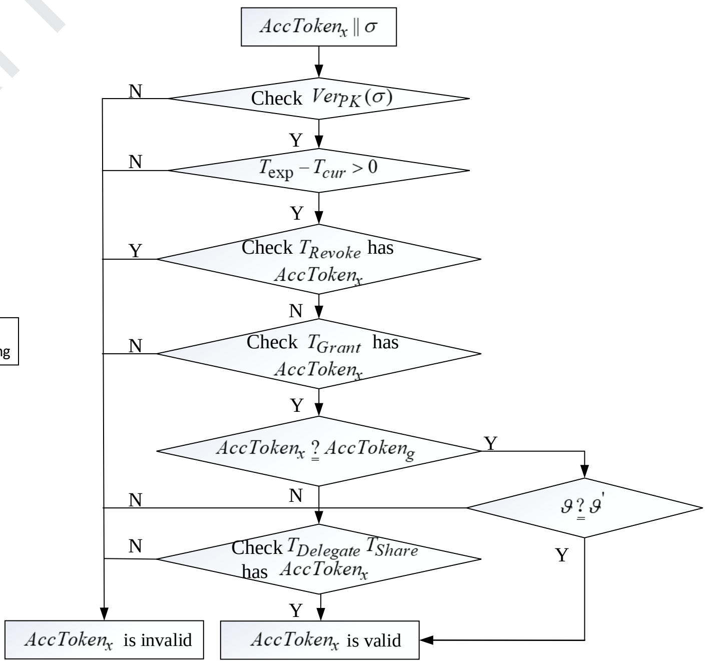

$AccToken_x \| \sigma$
Check $Verp_K(\sigma)$
$T_{exp} - T_{cur} > 0$
Check T_{Revoke} has $AccToken_x$
Check T_{Grant} has $AccToken_x$
$AccToken_x \stackrel{?}{=} AccToken_g$
$\vartheta \stackrel{?}{=} \vartheta'$
Check $T_{Delegate}$ T_{Share} has $AccToken_x$
$AccToken_x$ is invalid
$AccToken_x$ is valid
Y
N

Fig. 8. Verification process of access token

The requester B sends an access token and his own signature, $AccToken_x\|\sigma$, to content provider A, then A verifies its validity according to the following four steps (as shown in Fig.8.): firstly, A checks whether the signature is valid with PK_B in the $AccToken_x$; secondly, A checks the valid time of token; thirdly, A checks the $AccToken_x$ on all the T_{Revoke}, if not find, then check on T_{grant}; fourthly, if the $AccToken_x$ belongs grant tokens, A checks the token payload

Journal Pre-proof

according to section 4.1.3-C, else A checks T_{Share}, $T_{Delegate}$ according to typ_x. If B passes through the verification, A sends B the resources matched off blockchain and generates an *access transaction* T_{Access} as follows:

$$T_{Access} = (Tid, Access, Tin[PK_A, T_{pre}, \varphi], Tout[PK_B, AccToken_x \| H(AccOp), \omega]) \tag{11}$$

In Eq.11, H(AccOp) is the hash of the detailed information of this access activity.

4.2.4. Retrieve transactions with Cuckoo filter

In order to improve the retrieval efficiency of a transaction in the blockchain, Cuckoo filter is introduced. Cuckoo filter is a practical data structure, which supports adding and removing items dynamically while achieving better search performance and less space than Bloom filters (Fan et al., 2014). A Cuckoo filter consists of multiple buckets, where a bucket can have multiple entries, each entry stores one fingerprint (Cui et al., 2017). For inserting a data item x, the hashing function computes the indices of two candidate buckets b_1 and b_2 as follows (let $fingerprint(x)$ be the lowest k bits of $hash(x)$, M be the number of buckets):

$$\begin{aligned} b_1 &= hash(x) \mod M \\ b_2 &= b_1 \oplus hash(fingerprint(x)) \mod M \end{aligned} \tag{12}$$

If there is an empty bucket in the candidate buckets, we save the fingerprint of x into the empty bucket. Otherwise, we just select one of the candidate buckets, remove the existing item, and reinsert this item to its alternate buckets. This procedure is repeated until a vacant bucket is found or exceeds a maximum number of displacements. The lookup process of a Cuckoo filter is that, given an item x, first we calculates x's fingerprint and two candidate buckets according to Eq.12, and then traverse two candidate buckets, if any existing fingerprint in either bucket matches, returns true, otherwise returns false. The deletion process is that, checks both candidate buckets for a given item x, if any fingerprint matches items in any bucket, one copy of that matched fingerprint is removed from that bucket.

In our blockchain, we built two Cuckoo filters, CF_g, CF_{inv}, for grant tokens and invalid token, respectively. When the miner verifies that a new *Grant transaction* is successful, he inserts the hash of the token $H(AccToken_g)$ into the CF_g. When the miner verifies that a new *Revoke transaction* is successful, he simply removes the $H(AccToken_g)$ from the CF_g and insert it into CF_{inv}. In this way, a CP can quickly determine the validity of an $AccToken_g$ on blockchain by looking up $H(AccToken_g)$ in CF_{inv} or CF_g. If it is found in the CF_g, it is valid, else not. If the CP or CR needs to verify $AccToken_s$ or $AccToken_d$, at first, he needs search its corresponding $H(AccToken_g)$ (refer to section 4.1.3-B) in CF_{inv}. If it is found, then the access token is invalid, else he retrieves the address of the corresponding transaction block according to its corresponding $H(AccToken_g)$ in CF_g. Thirdly, he follows the transaction address to find T_{Share} or $T_{Delegate}$ according to typ_x. It is efficient since any one of T_{Share} or $T_{Delegate}$ is directly originated from a T_{Grant}.

5. Security analysis

In this section, we first prove the security features of our SBAC in terms of privacy (user anonymity and hidden policy), anti-counterfeiting, tamper-proof, access audit, access revocation, multi-level content access, completely control and resilience to other attacks. Then, we compare our scheme with several representative access control schemes in terms of some important aspects.

A. **Privacy (user anonymity and hidden policy)**

Proof. With the anonymity of blockchain, the users' (the CQ's and CP's) real identities are hidden. The access policy is confined in the CP since each CP is responsible to generate and maintain the access policy for the potential CQs and only tags its related information (the CQ's attribute score S_{CQ}) in $AccToken_x$. Further, although all the $AccToken_x$s are transmitted and recorded in the blockchain which can be accessed by any entity, an adversary $\mathcal{A}$ can know nothing but the transfer path of the $AccToken_x$. Specifically, $\mathcal{A}$ cannot obtain specific content information such as content name, content address, etc. from observing the transactions on the blockchain. In a word, our SBAC framework can guarantee user anonymity and privacy of access policy.

B. **Anti-counterfeiting**

Proof. Suppose an adversary $\mathcal{A}$ intercepts the access token $AccToken_g = tokenId||typ_g||data_{token}||PK_{CP}||PK_{CQ}||T_{gen}||T_{exp}||\vartheta$, and parses out the token format. Then $\mathcal{A}$ forges a new access token $AccToken_g^{'} = tokenId||typ_g||data_{token}||PK_{CP}||PK_{\mathcal{A}}||T_{gen}||T_{exp}||\vartheta$, in which PK_{CQ} is replaced by $PK_{\mathcal{A}}$. When $\mathcal{A}$ requests the content with $AccToken_g^{'}$, the CP can detect the forged access token by means of verifying the ϑ. Noting that $\vartheta = H(S_{CQ}||PK_{CP}||PK_{CQ}||T_{gen}||T_{exp}||salt)$ and the PK_{CQ} is replaced by $PK_{\mathcal{A}}$ in the $AccToken_g^{'}$, the CP can easily get $\vartheta \neq \vartheta^{'}$. Furthermore, $\mathcal{A}$ is almost impossible to forge the correct ϑ, since S_{CQ} and $salt$ are only known to CP, even if $\mathcal{A}$ launches brute-force attack to $salt$, it is hard for him because the space complexity of $salt$ is 32^{255}. Therefore, the forged access tokens can be detected in SBAC.

C. **Tamper-proof**

Proof. Assume that $\mathcal{A}$ tries to tamper an access token $AccToken_x$ during the period of the token's transmission which is from the token's creator (can be a CP, a sharer or a delegate) to the token's requester. Noting that the transfer of the $AccToken_x$ is based on blockchain transactions and the tamper-proof features of blockchain, any $\mathcal{A}$ fails to tamper the $AccToken_x$. Similarly, the access activities are also recorded in the blockchain, so $\mathcal{A}$ is impossible to tamper them successfully. As a result, our SBAC framework achieves tamper-proof for access tokens and access log.

D. **Access audit**

Proof. In SBAC, the implementation of access control relies on $AccToken_x$, and the transfer and revocation of $AccToken_x$ are realized as transactions on the blockchain. Further, the access activities are also recorded on the blockchain-based distributed ledger. As a result, a CP can audit access of his shared content in a detailed way including access token, time, and even each access activity. In other words, our SBAC framework provides a convenient and effective access audit for a CP.

E. **Access revocation**

Proof. Each CQ's content access requires $AccToken_x$ in our scheme. And the $AccToken_x$ is time limited which can be expired. In addition, the creator (CP, sharer, or delegate) of the $AccToken_x$ can revoke it through a revoke transaction T_{Revoke} at any time. Thus, access revocation of content is realized in SBAC.

F. **Multi-level content access**

Proof. In our scheme, each CQ is assigned attributes and the corresponding score S_{CQ} by a CP. At the same time, the CP reorganizes resources R_j into a hierarchical set of resources $RSet = \{(POp_i, \{R_j\}_i)\}$ according to their labeled permitting operations POp_i. Then, with the help of requester-content matching module (see Algorithm 2), the CP maps the CQ to a set of allowed access content $RSet_{allow}$ with the CQ's S_{CQ} and resources' POp_i. That is, multi-level content access is achieved in our scheme.

G. **Completely control**

Proof. Our framework not only allows a CP to achieve secure content sharing, but also provides him with access audit and access revocation of content, which satisfies the CP completely control of content access. In a word, our SBAC framework guarantees a CP completely control over the shared content.

H. **Resilience to other attacks**

Proof. Our proposed framework can resist the following attacks.

(a) **Cache poison attacks**. Suppose an adversary $\mathcal{A}$ fills of the cache (Roy et al., 2019; Tourani et al., 2018) with well constructed data to attack the requester, he can not succeed since the cached content in a CR is encrypted by a $sCode$ which is only known to the requester. That means, the requester decrypts the well constructed data from $\mathcal{A}$ with a $sCode$ and gets a bunch of meaningless codes. Therefore, $\mathcal{A}$ can not attack the requester through the cache poison in our framework.

(b) **Cache privacy attacks**. Suppose an adversary $\mathcal{A}$ launches a cache privacy attack where he can learn whether the specific content was recently requested by a CQ through probing a CR's cache (Acs et al., 2019), he will fail in our framework since the cached content in the CR is encrypted and only sent to the authenticated requester. In other words, if $\mathcal{A}$ wants to request the cached content, he has to provide a valid access token $AccToken_s$, while he can

Journal Pre-proof

not fetch or forge it since it is secured by blockchain. At the same time, an honset-but-curious CR can not fetch any privacy about the content or the requester because the content is encrypted and the requester uses pseudonym of blockchain. Thus, our framework can resist cache privacy attacks both from $\mathcal{A}$ and the honset-but-curious CR.

(c) **DDoS attacks**. SBAC inherits blockchain's resilience to DDoS attacks, even if some of the nodes are forced to go offline due to DDoS attacks, while other nodes can still work and each blockchain node can run a consensus algorithm. When these attacked nodes come back online, they can resynchronize the data to ensure the consistency and integrity. Moreover, ICN's distributed features enhance SBAC's defense against DDos attacks.

(d) **Man-in-the-middle attacks**. From the above, it is clear that SBAC provides secure access token transfer via the blockchain transactions. Thus, it is also man-in-the-middle attacks resilience.

I. Comparison

The important aspects of security and privacy features of our proposed framework and several representative access control schemes are compared in Table 7.

Table 7: Security comparison

scheme	User anonymous	Hidden policy	Anti-counterfeiting	Tamper -proof	Access audit	Access revocation	Multi-level content	Degree of content control
ABAC (Li et al., 2018)	-	√	√	×	×	×	×	1/3
Live (Li et al., 2015)	-	-	√	√	×	×	two-level	1/3
BPDS (Fan et al., 2018)	√	×	-	√	×	×	×	1/3
BBAC (Maesa et al., 2017)	√	×	-	√	×	√	×	2/3
FairAccess (Ouaddah et al., 2016, 2017)	√	√	-	√	×	√	-	2/3
SEAF (Xue et al., 2018)	√	×	√	√	*limited*	×	√	2/3
SBAC	√	√	√	√	√	√	√	3/3

Journal Pre-proof

As shown in Table 7, the ABAC (Li et al., 2018) focuses on read-only content, achieves privacy with hiding the access policy in the content name and also has the ability of anti-counterfeiting. But the ABAC (Li et al., 2018) does not provide user anonymity, tamper-proof, access audit, client revocation, and multilevel content access. The Live (Li et al., 2015) also deals with read-only content, but it only provides two-level content access (public and private), tamper-proof and anti-counterfeiting. The FairAccess (Ouaddah et al., 2016, 2017) utilizes the blockchain to hide policy into the smart contract, designs a client revocation transaction, and inherits the user anonymity and tamper-proof of blockchain, but it does not take the hierarchical access into consideration. Compared with the FairAccess (Ouaddah et al., 2016, 2017), the access policy in the BBAC (Maesa et al., 2017) and the BPDS (Fan et al., 2018) has been exposed, and the BPDS (Fan et al., 2018) does not provide client revocation either. For the degree of content control, the ABAC (Li et al., 2018), Live (Li et al., 2015) and BPDS (Fan et al., 2018) only provide the content provider with the ability of sharing the content, the FairAccess (Ouaddah et al., 2016, 2017) and BBAC (Maesa et al., 2017) further present the content provider with the ability of revoking the content access. The SEAF (Xue et al., 2018) achieves user anonymity, tamper-proof, anti-counterfeiting, multilevel content access and limited access audit, while the privacy of access policy and client revocation are ignored. Compared to the SEAF (Xue et al., 2018), our framework not only further achieves the privacy of access policy and client revocation, but also offers the ability of sharing the content, revoking the content access and auditing the history of access activities.

In addition, the comparison of resilience to known attacks is given in Table 8, where ''√'' denotes defensible and ''×'' means failed. Table 8 shows that our framework can resist the cache poison attacks, cache privacy attacks, DDoS attacks and Man-in-the-middle attacks. None of the other ICN's schemes (ABAC (Li et al., 2018), Live (Li et al., 2015), BPDS (Fan et al., 2018) and SEAF (Xue et al., 2018)) in the Table 8 can resist the cache privacy attacks although they can resist the cache poison attacks. And Live (Li et al., 2015) and SEAF (Xue et al., 2018) only can resist the DOS attacks.

As a result, our framework provides a content provider (CP) with comprehensive access audit and access revocation by introducing the access transaction and revoke transaction to the blockchain system while keeping user anonymity and tamper-proof. Secondly, our framework achieves multi-level content access through dynamic mapping the content requester's (CQ's) attribute score to the content while also preserving hidden policy and anti-counterfeiting. Lastly, our framework can resist cache privacy attacks while existing works failed.

Journal Pre-proof

Table 8: Resilience comparison

scheme	Cache poison attacks	Cache privacy attacks	DDoS attacks	Man-in-the-middle attacks
ABAC (Li et al., 2018)	√	×	×	√
Live (Li et al., 2015)	√	×	DoS	√
BPDS (Fan et al., 2018)	√	×	√	√
BBAC (Maesa et al., 2017)	-	-	√	√
FairAccess (Ouaddah et al., 2016, 2017)	-	-	√	√
SEAF(Xue et al., 2018)	√	×	DoS	√
SBAC	√	√	√	√

6. Implementation and evaluation

6.1. Implementation

A prototype of our proposed SBAC framework is developed and implemented to verify the feasibility. The main processes of matching-based access control model are developed by python3.5 in the platform of windows10 where the python libraries, such as hashlib, json, sha3, base64, etc. are utilized. The implementation of *blockchain-based access token mechanism* is based on Ethereum, web3, Node.js. The prototype contains 5 content providers, 5 content requesters and 15 ICN content routers. All of them are implemented with intel(R) Core(TM) i5-7200U CPU @2.5GHz, 8GB RAM, and the interfaces of content providers or content requesters are Firefox or Chrome with Metamask.

Journal Pre-proof

6.2. Result analysis

Firstly, scoring for the potential CQs' attributes and configuring permissions for resources are the initial important two functions of a CP. A CP lists the base-types of user attributes and their corresponding sub-types, then Algorithm 1 does scoring (refer to section 4.1.2). The graph of the variation of scoring time with base-types and sub-types of user attributes is illustrated in Fig. 9, where the measurements are set as the averages of over 200 test runs.

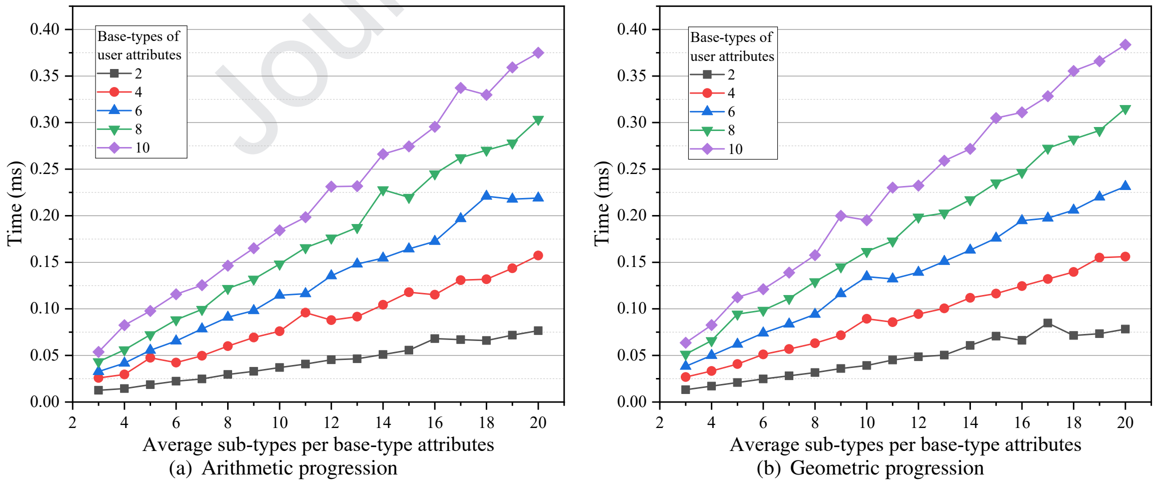


Fig. 9. Process time for scoring attributes

The scoring time of both the arithmetic progression (Fig. 9(a)) and geometric progression (Fig. 9(b)) increases with the number of base-types and sub-types of user attributes. And the longest process time in the graph is less than 0.4ms for 200 sub-types of user attributes (10 base-types of user attributes and 20 sub-types for each on average) which is much more ample than practical needs.

Journal Pre-proof

And for the configuring permissions for resources, the CP labels each of the resources(files) to be shared and classifies them into multilevel sets (refer to section 4.1.1). The performance of configuring permissions for files is shown in Fig. 10, where we have 200 test runs. We can conclude that the process time is acceptable and is only proportional to the number of files. Specifically, nearly 0.03s is taken when configuring 1 file, about 0.25s for 10 files, and around 2.8s for 100files, no matter what size of it (5KB, 500KB, 1MB, 100MB, 500MB, 1GB or 2GB).

Table 9: Gas cost of SBAC

Function	Gas units	Gas cost(eth)
Deploy smart contracts	3174844	0.0130169
Access token transfer	137204	0.0005625
Access with access token	304802	0.0012497

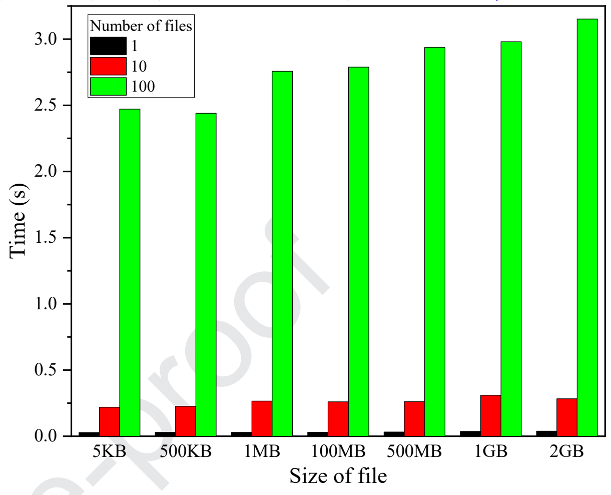


Fig. 10: Time of configuring permissions for files

Secondly, the gas cost of the blockchain-based access token mechanism is evaluated, as shown in Table 9, where the gas price is 0.0041 ether per million gas according to the (ETH, 2018)'s data in December, 2018. As a result, the cost of deploying and executing on Ethereum are small.

Thirdly, the performance of a content access is investigated. When a new CQ launches a content request through the ICN, the CP assigns attributes' score for him and generates the corresponding access token with the proposed matching-based access control model; then after signing the access token which is transmitted from a blockchain transaction, the CQ continues the content request to the CP off blockchain; finally, the CP verifies the access token on blockchain, responds the content to the CQ off blockchain and records this access activity on blockchain. As above, the execution of content access can be divided into processes off blockchain, access token transactions on blockchain and communication. The process time of the main functions off blockchain are illustrated in Table 10, where measurements are also set as the averages of over 200 test runs. From Table 10, the total of process time cost is 1.04096ms for a new CQ's content access and the corresponding cost is 0.46512ms for the one who owned an access token since the first two functions in Table 10 does not need.

Table 10: Process time of SBAC

Subject	Function	Time(ms)
CP	Assign attributes for a requester	0.003
CP	Generate access token	0.67284
CQ	Sign access token	0.011
CP	Verify access token	0.35412
Total		1.040906

Table 11: Transaction storage of SBAC

Transaction	Storage(bytes)
Grant transaction	270
Share transaction	366
Delegate transaction	366
Revoke transaction	302
Access transaction	398

According to (Chen et al., 2018), the size of one block of Ethereum is 76MB and the blocks are generated about every 6.7s on average. And the storage of each type of access token transaction in SBAC is shown in Table 11, thus, the Ethereum can surpport 230,000 access token transactions each block. In other words, a CQ has to wait 6.7s to get an access token. The communication time of the access activity between a CP and a CQ can be omitted since it is often less than 0.009ms for 1KB data in a 100MB/s network. As above, a CQ has to provide an access token to get content from a CP or a CR off blockchain. As a result, the total of time cost is 6.7s+1.04096ms=6.70104096s for a new CQ's content access and the corresponding cost is 0.46512ms for the one who owned an access token, because the access transactions record the access activities after the end of content access.

At last, a comparison of communication cost, computation cost, the number of transactions for one content access

Journal Pre-proof

and token's efficiency of our SBAC and several representative schemes are presented in Table 12.

Table 12: Performance comparison

Scheme	Communication cost	Computation cost	Transactions	Contents Vs Tokens
ABAC(Li et al., 2018)	$6M$	$2E/D + 5Pair + 9Exp + 1Mul$	0	-
Live(Li et al., 2015)	$3M$	$4E/D + 4Sig/Ver + (n + 2)H$	0	$1Vs1$(public content), $1Vs2$(private content)
BPDS(Fan et al., 2018)	$4M$	$4E/D + 2Sig/Ver + 3H + 2Q$	$1Tr$	-
BBAC(Maesa et al., 2017)	$2M$	$4Sig/Ver + 2H + 1Q$	$2Tr$	-
FairAccess (Ouaddah et al., 2016, 2017)	$3M$	$2E/D + 4Sig/Ver + 1H + 1Q$	$2Tr$	$1Vs1$
SEAF(Xue et al., 2018)	$4M$	$2E/D + 2GSig/BVer + nH$	0	-
SBAC	$3M$	$2E/D + 4Sig/Ver + 6H$	$1Tr$	n$Vs1$

[1] M-Message, Tr-Transactions, E/D-Encryption/Decryption, Pair-Pairing, Exp-Exponentiation, Mul-Multiplication, Sig/Ver-Signature/Verify, GSig/BVer-Group Signature/Batch Verify, H-Hash, Q-Query the transactions in blockchain.

As shown in Table 12, the communication cost of our proposed framework is medium (3 messages) which is less than the ones of BPDS (Fan et al., 2018), ABAC (Li et al., 2018), SEAF (Xue et al., 2018) and larger than the one of BBAC (Maesa et al., 2017). Since each scheme adopts different computations, and for comparability, so we take the estimated calculation according to the parameters used in (Lyu et al., 2019), where the exponentiation, multiplication, signature/verify and group signature/batch verify are the heaviest computations, the time of encryption/decryption is one third of the previous ones, and the time of the hash is very little that can be omitted. For simplicity, we take one heaviest computation as 1 cost unit, and we also set the cost of one transaction query in blockchain to 1 cost unit since it is a time consuming operation in the ever-growing blockchain system. Therefore, the total time cost of our framework is 4.7 cost units which is less than the ones using blockchain (BPDS (Fan et al., 2018) is 5.3; BBAC (Maesa et al., 2017) is 5 ; FairAccess (Ouaddah et al., 2016, 2017) is 5.7). The reason that no cost of blockchain transaction's query in our scheme is not we do not need the query of the blockchain but Cuckoo filter we introduced reduces the cost of transaction's query to a level of hashes' cost. And for one content access, our framework and BPDS (Fan et al., 2018) need 1 transaction which is less than the one of other blockchain-based schemes (BBAC (Maesa et al., 2017) and FairAccess (Ouaddah et al., 2016, 2017) are 2.). In the schemes where access token used, each resource needs one or two tokens in FairAccess (Ouaddah et al., 2016, 2017) and Live (Li et al., 2015) , whereas our framework achieves one token matches a bunch of resources, and even when they change or update, no extra computation is introduced.

7. Conclusion

A secure blockchain-based access control framework (SBAC) is proposed to provide a content provider (CP) with complete control of his own content in this paper. Specifically, a blockchain-based access token mechanism is introduced to implement the content access control of sharing, audit, and revocation through the access token transfer and access transactions. A matching-based access control model is designed to achieve hierarchical access and efficient access control, where one access token can fetch a bunch of resources and it still works even when they change (such as a resource is added, deleted or modified) during its period of validity. Our framework inherits the features of blockchain's pseudonym, decentralization and tamper-proof to balance user anonymity and access audit. Furthermore, the Cuckoo filter is introduced to improve the efficiency of access token query in verification and a caching policy is designed to satisfy the pervasive caching of ICN. The security analysis and the evaluation result show that our SBAC achieves better security than the existing schemes and can meet the practical needs.

Acknowledgement

This work was supported by the 2019 Discipline Construction of Zhejiang Provincial Key University-Cyberspace Security (No.GK198800299013), the Cyberspace Security Major Program in National Key Research and Development Plan of China under grant No.2016YFB0800201, the National Natural Science Foundation of China under

Journal Pre-proof

grant No.61401128, 61702150 and 61803135, Natural Science Foundation of Zhejiang Province under grant No. LY19F020039, the Key Research and Development Plan Project of Zhejiang Province under grant No.2017C01062 and 2017C01065.

References

Abdallah, E.G., Zulkernine, M., Hassanein, H.S., 2016. Dacpi: A decentralized access control protocol for information centric networking, in: 2016 IEEE International Conference on Communications (ICC). doi:10.1109/ICC.2016.7511198.

Acs, G., Conti, M., Gasti, P., Ghali, C., Tsudik, G., Wood, C.A., 2019. Privacy-aware caching in information-centric networking. IEEE Trans. Dependable Secur. Comput. 16, 313–328. doi:10.1109/TDSC.2017.2679711.

Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., Ohlman, B., 2012. A survey of information-centric networking. IEEE Commun. Mag. 50, 26–36. doi:10.1109/MCOM.2012.6231276.

Aiash, M., Loo, J., 2015. A formally verified access control mechanism for information centric networks, in: 2015 12th International Joint Conference on e-Business and Telecommunications (ICETE), pp. 377–383. doi:10.5220/0005566303770383.

Anand, A., Muthukrishnan, C., Akella, A., Ramjee, R., 2009. Redundancy in network traffic: findings and implications, in: Eleventh International Joint Conference on Measurement and Modeling of Computer Systems,. doi:10.1145/1555349.1555355.

Azad, M.A., Bag, S., Feng, H., 2018a. Privbox: Verifiable decentralized reputation system for the on-line marketplaces. Futur. Gener. Comp. Syst. 89, 44–57. doi:10.1016/j.future.2018.05.069.

Azad, M.A., Bag, S., Parkinson, S., Feng, H., 2018b. M2m-rep: Reputation system for machines in the internet of things. Comput. Secur. 79, 1–16. doi:10.1016/j.cose.2018.07.014.

Azad, M.A., Bag, S., Parkinson, S., Feng, H., 2018c. Trustvote: Privacy-preserving node ranking in vehicular networks. IEEE Internet Things J. PP, 1–14. doi:10.1109/JIOT.2018.2880839.

Azad, M.A., Bag, S., Tabassum, S., Feng, H., 2017. privy: Privacy preserving collaboration across multiple service providers to combat telecom spams. IEEE Trans. Emerg. Top. Comput. PP, 1–14. doi:10.1109/TETC.2017.2771251.

Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J.A., Felten, E.W., 2015. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies, in: 2015 IEEE Symposium on Security and Privacy SP, pp. 104–121. doi:10.1109/SP.2015.14.

Chen, J., He, K., Du, R., Xiang, Y., 2015. Dominating set and network coding-based routing in wireless mesh networks. IEEE Trans. Parallel Distrib. Syst. 26, 423–433. doi:10.1109/TPDS.2013.303.

Chen, J., Yao, S., Yuan, Q., He, K., Ji, S., Du, R., 2018. Certchain: Public and efficient certificate audit based on blockchain for tls connections, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, pp. 2060–2068. doi:10.1109/INFOCOM.2018.8486344.

Cisco, 2019. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022 White Paper. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html. (accessed 30 April 2019).

Cui, J., Zhang, J., Zhong, H., Xu, Y., 2017. Spacf: A secure privacy-preserving authentication scheme for vanet with cuckoo filter. IEEE Trans. Veh. Technol. 66, 10283–10295. doi:10.1109/TVT.2017.2718101.

ETH, G.S., 2018. Eth gas station: Tx calculator. https://ethgasstation.info/calculatorTxV.php. (accessed 25 December 2018).

Fan, B., Andersen, D.G., Kaminsky, M., Mitzenmacher, M.D., 2014. Cuckoo filter: Practically better than bloom, in: Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies, pp. 75–88. doi:10.1145/2674005.2674994.

Fan, K., Ren, Y., Wang, Y., Li, H., Yang, Y., 2018. Blockchain based efficient privacy preserving and data sharing scheme of content-centric network in 5g. IET Commun. 12, 527–532. doi:10.1049/iet-com.2017.0619.

Fang, C., Yu, F.R., Huang, T., Liu, J., Liu, Y., 2015. A survey of green information-centric networking: Research issues and challenges. IEEE Commun. Surv. Tutorials. 17, 1455–1472. doi:10.1109/COMST.2015.2394307.

Fotiou, N., Marias, G.F., Polyzos, G.C., 2012. Access control enforcement delegation for information-centric networking architectures. ACM SIGCOMM Comp. Commun. Rev. 42, 497–502. doi:10.1145/2377677.2377773.

Gai, K., Wu, Y., Zhu, L., Qiu, M., Shen, M., 2019a. Privacy-preserving energy trading using consortium blockchain in smart grid. IEEE Trans. Ind. Inform. 15, 3548–3558. doi:10.1109/TII.2019.2893433.

Gai, K., Wu, Y., Zhu, L., Xu, L., Zhang, Y., 2019b. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. IEEE Internet Things J. PP, 1–1. doi:10.1109/JIOT.2019.2904303.

Garay, J., Kiayias, A., Leonardos, N., 2015. The bitcoin backbone protocol: Analysis and applications, in: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 281–310. doi:10.1007/978-3-662-46803-6_10.

Ghali, C., Schlosberg, M.A., Tsudik, G., Wood, C.A., 2015. Interest-based access control for content centric networks, in: Proceedings of the 2nd ACM Conference on Information-Centric Networking, pp. 147–156. doi:10.1145/2810156.2810174.

HHamdane, B., El Fatmi, S.G., 2015. A credential and encryption based access control solution for named data networking, in: Proceeding of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 1234–1237. doi:10.1109/INM.2015.7140473.

Jiang, X., Bi, J., Nan, G., Li, Z., 2015. A survey on information-centric networking: Rationales, designs and debates. China Commun. 12, 1–12. doi:10.1109/CC.2015.7188520.

Kiayias, A., Russell, A., David, B., Oliynykov, R., 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol, in: 37th Annual International Cryptology Conference (Crypto), pp. 357–388. doi:10.1007/978-3-319-63688-7_12.

Li, B., Huang, D., Wang, Z., Zhu, Y., 2018. Attribute-based access control for icn naming scheme. IEEE Trans. Dependable Secur. Comput. 15, 194–206. doi:10.1109/TDSC.2016.2550437.

Li, Q., Sandhu, R., Zhang, X., Xu, M., 2017. Mandatory content access control for privacy protection in information centric networks. IEEE Trans. Dependable Secur. Comput. 14, 494–506. doi:10.1109/TDSC.2015.2494049.

Li, Q., Zhang, X., Zheng, Q., Sandhu, R., Fu, X., 2015. Live: Lightweight integrity verification and content access control for named data networking. IEEE Trans. Inf. Forensic Secur. 10, 308–320. doi:10.1109/TIFS.2014.2365742.

Journal Pre-proof

Lyu, Q., Zheng, N., Liu, H., Gao, C., Chen, S., Liu, J., 2019. Remotely access "my" smart home in private: An anti-tracking authentication and key agreement scheme. IEEE Access 7, 41835–41851. doi:10.1109/ACCESS.2019.2907602.

Maesa, D.D.F., Mori, P., Ricci, L., 2017. Blockchain based access control, in: 17th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS) Held as Part of the 12th International Federated Conference on Distributed Computing Techniques (DisCoTec), pp. 206–220. doi:10.1007/978-3-319-59665-5_15.

Misra, S., Tourani, R., Natividad, F., Mick, T., MMajd, N.E., Huang, H., 2019. Accconf: An access control framework for leveraging in-network cached data in the icn-enabled wireless edge. IEEE Trans. Dependable Secur. Comput. 16, 5–17. doi:10.1109/TDSC.2017.2672991.

Nakamoto, S., 2008. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/en/bitcoin-paper.

Ouaddah, A., Abou Elkalam, A., Ouahman, A.A., 2016. Fairaccess: a new blockchain-based access control framework for the internet of things. Secur. Commun. Netw. 9, 5943–5964. doi:10.1002/sec.1748.

Ouaddah, A., Elkalam, A.A., Ouahman, A.A., 2017. Towards a novel privacy-preserving access control model based on blockchain technology in iot, in: Europe, Middle East and North Africa Conference on Technology and Security to Support Learning (EMENA-TSSL), pp. 523–533. doi:10.1007/978-3-319-46568-5_53.

Paul, S., Pan, J., Jain, R., 2011. Architectures for the future networks and the next generation internet: A survey. Comput. Commun. 34, 2–42. doi:10.1016/j.comcom.2010.08.001.

Renault, E., Ahmad, A., Abid, M., 2010. Access control to objects and their description in the future network of information. J. Inf. Process. Syst. 6, 359–374. doi:10.3745/JIPS.2010.6.3.359.

Roy, S., J. Anto Morais, F., Salimitari, M., Chatterjee, M., 2019. Cache attacks on blockchain based information centric networks: An experimental evaluation, in: Proceedings of the 20th International Conference on Distributed Computing and Networking, ACM, pp. 134–142. doi:10.1145/3288599.3288640.

da Silva, R.S., Zorzo, S.D., 2015. An access control mechanism to ensure privacy in named data networking using attribute-based encryption with immediate revocation of privileges, in: 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), pp. 128–133. doi:10.1109/CCNC.2015.7157958.

Szabo, N., 1997. Formalizing and securing relationships on public networks. First Monday. 2. doi:10.5210/fm.v2i9.548.

Tourani, R., Misra, S., Mick, T., Panwar, G., 2018. Security, privacy, and access control in information-centric networking: A survey. IEEE Commun. Surv. Tutor. 20, 556–600. doi:10.1109/COMST.2017.2749508.

Xue, K., Zhang, X., Xia, Q., Wei, David S.L.and Yue, H., Wu, F., 2018. Seaf: A secure, efficient and accountable access control framework for information centric networking, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, pp. 2213–2221. doi:10.1109/INFOCOM.2018.8486407.

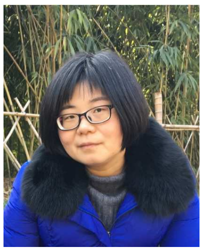
Zhang, M., Luo, H., Zhang, H., 2015. A survey of caching mechanisms in information-centric networking. IEEE Commun. Surv. Tutor. 17, 1473–1499. doi:10.1109/COMST.2015.2420097.

Zhu, L., Wu, Y., Gai, K., Choo, K.K.R., 2019. Controllable and trustworthy blockchain-based cloud data management. Futur. Gener. Comput. Syst. 91, 527–535. doi:10.1016/j.future.2018.09.019.

Zyskind, G., Nathan, O., Pentland, A.S., 2015. Decentralizing privacy: Using blockchain to protect personal data, in: 2015 IEEE Security and Privacy Workshops (SPW), pp. 180–184. doi:10.1109/SPW.2015.27.

Journal Pre-proof

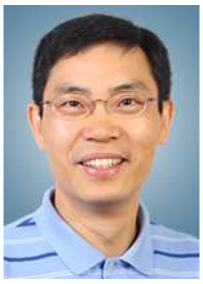
Journal Pre-proof



Qiuyun Lyu is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. She received the bachelor’s and master’s degrees from Chang'an University, in 2000 and 2003, respectively. She is an associate professor of the School of Cyberspace, Hangzhou Dianzi University. Her current research interests include access control framework, privacy enhancing technology and blockchain.

Yizhen Qi received her B.S. degree from Hangzhou Dianzi University, Hangzhou, China, in 2016. She is currently pursuing the master degree in Communication Engineering in Hangzhou Dianzi University. Her research interests include blockchain, privacy protection ,access control and security issues in the information centric networking(ICN), etc.

Xiaochen Zhang is an undergraduate in Hangzhou Dianzi University, Hangzhou, China. He is currently majoring in information security. His research interests include network sec urity, blockchain and data mining, etc.



Huaping Liu received the B.S. and M.S. degrees in electrical engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 1987 and 1990, respectively, and the Ph.D. degree in electrical engineering from New Jersey Institute of Technology, Newark. Since September 2001, he has been with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, where he is currently a professor. His research interests include privacy enhancing technology in communication systems and multiuser communications.

Qiuhua Wang received her B.S. and M.S. degrees in communication engineering from Liaoning Technical University, Fuxin, China, in 2000 and 2003, respectively. She received her Ph.D. degree in communications and information systems from Zhejiang University, Hangzhou, China, in 2013. Now, she is an Associate Professor of the School of Cyber Security, Hangzhou Dianzi University. Her current research interests include network security, security issues in wireless networks, key management and physical layer security, etc.

Ning Zheng is the vice president of Hangzhou Dianzi University, Hangzhou, China. He has authored over 70 referred journal and conference papers. His research interests are privacy enhancing information management system and network information security.

Journal Pre-proof

Journal Pre-proof

Conflict of Interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled “SBAC: A secure blockchain-based access control framework for information-centric networking”.

Journal Pre-proof