# A Secure, Efficient, and Accountable Edge-Based Access Control Framework for Information Centric Networks

Kaiping Xue, *Senior Member, IEEE*, Peixuan He, Xiang Zhang, Qiudong Xia,
David S. L. Wei, *Senior Member, IEEE*, Hao Yue, and Feng Wu, *Fellow, IEEE*

***Abstract*— Information centric networking (ICN) has been regarded as an ideal architecture for the next-generation network to handle users' increasing demand for content delivery with in-network cache. While making better use of network resources and providing better service delivery, an effective access control mechanism is needed due to the widely disseminated contents. However, in the existing solutions, making cache-enabled routers or content providers authenticate users' requests causes high computation overhead and unnecessary delay. Also, the straightforward utilization of advanced encryption algorithms makes the system vulnerable to DoS attacks. Besides, privacy protection and service accountability are rarely taken into account in this scenario. In this paper, we propose SEAF, a secure, efficient, and accountable edge-based access control framework for ICN, in which authentication is performed at the network edge to block unauthorized requests at the very beginning. We adopt group signature to achieve anonymous authentication and use hash chain technique to reduce greatly the overhead when users make continuous requests for the same file. At the same time, we provide an efficient revocation method to make our framework more robust. Furthermore, the content providers can affirm the service amount received from the network and extract feedback information from the signatures and hash chains. By formal security analysis and the comparison with related works, we show that SEAF achieves the expected security goals and possesses more useful features. The experimental results also demonstrate that our design is efficient for routers and content providers and bring in only slight delay for users' content retrieval.**

***Index Terms*— Information centric networking, access control, access privilege level, service accounting, revocation.**

## I. Introduction

To cope with the mismatch between current IP address based Internet architecture and users' demand for content delivery, the Information Centric Networking (ICN) has been proposed as a promising new paradigm [1], [2]. In ICN, the emphasis is shifted from *where* the content is located to *what* the content is. Specifically, the content in ICN (also called *chunk*, a smaller data unit segmented from file) is described by its *name* and users request a content by sending an interest packet containing the desired content name. When the interest packet reaches the origin server or a cache-enabled router which has already cached a copy of the requested content, the content is sent back to the requester in a data packet. Because of these characteristics, ICN can make the best use of network resources, e.g., bandwidth and routers' cache space, and deliver contents to users with lower latency. In addition, ICN supports multicast and mobility inherently.

Despite the above advantages, ICN also poses some new challenges, among which access control is an important one. In the current Internet, when a user makes a request (e.g., an HTTP request) for certain content, a centralized content provider (CP) will decide to approve or deny the request according to an access control list. However, due to the existence of in-network cache in ICN, any request can be satisfied by the routers in the forwarding path while content providers (CPs) have no control over the routers' behavior. In this case, unauthorized users can easily obtain their desired contents from the network without content providers' permission, which has a negative impact on content providers' profits. Therefore, an effective access control mechanism is desperately needed for the successful deployment of ICN.

Overall, the existing access control solutions for ICN can be divided into two categories: authentication-based [3]–[6] and encryption-based [7]–[11]. In the authentication-based schemes, the cache-enabled router where cache hit occurs initiates an authentication process to decide whether to send the requested content back or not. The authentication process either happens right on every single cache-enabled router or requires interactions with CP [4] or an access control server [5] during the content retrieval. Authentication on routers brings in heavy computation overhead and degrades the forwarding performance [3], [6]. Also frequent interactions cause significant delay for users and offset the benefits provided by in-network cache. Moreover, a common flaw is that authentication is performed on every chunk request so that users have to go through multiple authentication processes to retrieve a complete file. This seems clumsy but cannot

Manuscript received January 14, 2019; accepted April 22, 2019; approved by IEEE/ACM Transactions on Networking Editor K. Ren. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0801702, in part by the Key Research Program of the Chinese Academy of Sciences (CAS) under Grant ZDRW-KT-2016-2-5, in part by the Youth Innovation Promotion Association of CAS under Grant 2016394, and in part by the National Natural Science Foundation of China under Grant 61671420. *(Corresponding author: Kaiping Xue.)*

K. Xue, P. He, X. Zhang, Q. Xia, and F. Wu are with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China (e-mail: kpxue@ustc.edu.cn; hnythyq@mail.ustc.edu.cn; mm1201@mail.ustc.edu.cn; xqd95@mail.ustc.edu.cn; fengwu@ustc.edu.cn).

D. S. L. Wei is with the Department of Computer and Information Science, Fordham University, New York, NY 10458 USA (e-mail: wei@cis.fordham.edu).

H. Yue is with the Department of Computer Science, San Francisco State University, San Francisco, CA 94132 USA (e-mail: haoyue@sfsu.edu).

Digital Object Identifier 10.1109/TNET.2019.2914189

be avoided as the requests are satisfied at different routers and the routers are not aware of each other. On the other hand, the encryption-based schemes achieve access control by restricting users' decryption capability to contents [7]–[10]. By adopting cryptographic algorithms such as attribute-based encryption (ABE) [9], only authorized users are able to decrypt the encrypted contents so the confidentiality of the contents is preserved. Unfortunately, though the unauthorized users cannot decrypt the contents, they can still retrieve contents from the network because the routers do not discriminate the requests. As a result, the network resources can be easily exhausted by flooding requests [11], [12]. Besides, only a few of these schemes consider the privilege revocation, which will be the stumbling block to the deployment of ICN in real world.

Therefore, intuitively, the network should take charge of access control when users' requests are satisfied on routers since it would be a detour to bother content providers. But in such case, users' privacy is at risk because routers can know both what they request and who they are through the authentication [13]–[15]. Hence, a well-designed access control scheme should take privacy protection into account. Moreover, due to the existence of cache hit and the aggregation of request packets, it's hard for CPs to know the exact service amount that Internet Service Provider (ISP) provides, which makes it difficult for CPs to pay ISP based on their usage [16]. To convince CPs to pay their bills willingly, ISP should provide indubitable credentials so that CPs can count how many of their users' requests have been served (forwarded or satisfied). Also, it would be preferable that the credentials can contain useful feedback information about users' preferences and content popularity to help CPs improve their content services [17], [18].

Motivated by the above observations, we present SEAF, a secure, efficient and accountable edge-based access control framework in this paper. In SEAF, to separate access control from content provision, we let routers at the network edge authenticate users' requests and only authenticated requests can enter the network, so the bandwidth and cache resources inside the network are only accessible to authorized users. For privacy protection, users authenticate themselves to edge routers by generating a valid group signature, which can keep them anonymous to the edge routers. Nevertheless, signature generation and verification require expensive computation. Thus, a trivial solution by which users generate signatures for every request is impractical. To avoid the heavy construction, SEAF makes full use of the continuity of users' requests and bridges hash chain technique with group signature so that only the first of a series of requests requires signature verification operation and the rest can be authenticated by lightweight hash operation. Since the lengths of hash chains are the same as the numbers of users' requests, signatures and hash chains can be used as service credentials to convince content providers that ISP indeed provides the service it claims. Our contributions can be summarized as follows:

- We propose an effective and efficient access control framework for ICN. By placing access control task at the network edge, unauthorized requests can be blocked at the very beginning, which allows cache-enabled routers to focus on forwarding and cache operation.
- We design a lightweight and privacy-preserving authentication protocol between users and edge routers through a combination of group signature and hash chain techniques. Furthermore, CPs can use the signatures and hash chains to count the service amount provided by ISP.
- We further provide an efficient revocation method for the framework. With the revocation mechanism, it is easy for CPs to revoke users' privileges by restricting their requests to enter into ICN network.
- We formally analyze the security strength and conduct experiments by means of algorithm implementation and network simulation. The experiment results show that our design achieves better access control and only introduces slight content retrieval delay.

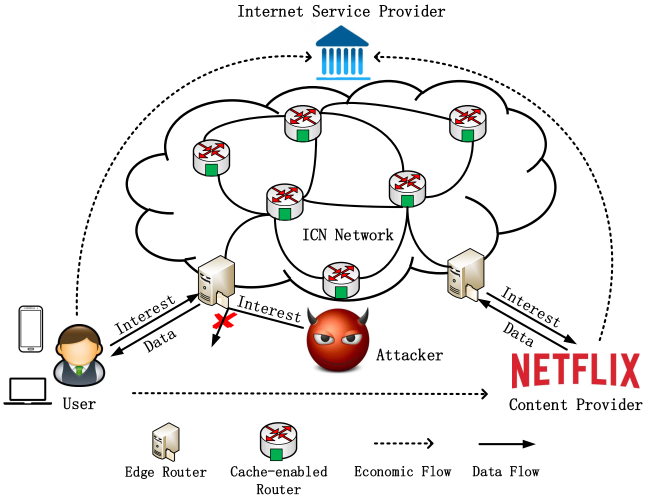


Fig. 1. System model.

This paper inherits the basic idea of our conference paper [19]. They differ mainly in the following aspects: i) In this paper, targeting at the authentication at edge router side, we design an efficient revocation approach to enable CP to revoke users' privileges in time without introducing too much overhead. ii) To make service accounting efficiently, we provide a probabilistic verification method based on batch verification and analyze its security and performance in detail. iii) We restate the security analysis and give a more rigorous version to show that our system is secure enough. iv) We accomplish the revocation mechanism and re-design the simulation sceneries to show the influence of bringing in the mechanism in our system.

The rest of this paper is organized as follows. We state our system model, security assumptions and design goals in Section II. We then present preliminaries in Section III and the construction of our access control scheme in Section IV. Security analysis is shown in Section V and performance evaluation is presented in Section VI. We also discuss the related work in Section VII, and finally conclude this work in Section VIII.

## II. System Model, Security Assumption and Design Goals

### A. System Model

We consider an Information Centric Networking architecture consisting of many content providers (CPs), an Internet Service Provider (ISP) network, and a large number of users, as shown in Fig. 1.

- *Content providers*: they are the producers of contents like Netflix and they are in charge of user registration and key management. To ensure the confidentiality of the published contents, they encrypt all the contents before publication.
- *Internet Service Provider*: it owns an ICN network and provides the other entities in the system with network access service. Besides, it caches contents in ICN network to improve the quality of users' experience and also deals with user authentication and service accounting. ICN network includes two types of routers: cache-enabled routers and edge routers. Specifically, the cache-enabled routers forward the requests and make the responses if the requested contents exist in their cache. Besides the tasks for which cache-enabled routers are responsible, the edge routers also need to authenticate users' requests before injecting them into the network.
- *Users*: they are the contents consumers and obtain desired contents from CPs or ICN network. Users will lose access privileges if their accounts have no money or they are compromised to do something threatening the security of the system like flooding interests.

There are financial relationships among these parties. ISP provides content forwarding and cache services to CPs via ICN network, and CPs should be charged based on the service amount, i.e., the number of requests that are forwarded to CPs or satisfied on the routers. Also, users pay CPs to get different access privileges for subscribed contents and pay ISP for the network access service.

### B. Security Assumption

ISP provides cached contents to users and charges CPs according to the service amount it provides. We assume that ISP is *rational*, *curious*, and *greedy*. By *rational*, we mean that ISP, as an enterprise, concerns about its economic benefits and reputation. Therefore, it will follow the designated protocols honestly to attract more content providers to purchase its service. By *curious*, we mean that ISP is curious about the rich information of the cached contents, and users' access patterns, e.g. who are interested in what kinds of data at what locations and time. By *greedy*, we mean that in order to get more profits from CPs, ISP may lie about the provided service amount or forge more accounting credentials. However, by considering the fact that in real world, ISP is always a reputed enterprise, like AT&T, that treasures its reputation, we assume that it dares not to cheat when this behavior can be detected by CP with a non-negligible possibility like 0.1%.

As content owners, CPs are assumed to be trusted in our system. They pay ISP according to the service amount, i.e. the number of users' requests served. Users are assumed to be untrusted. On the one hand, they try to get unauthorized data by tampering, replaying, or forging. On the other hand, they may collude with each other, even with ISP, to perform above attacks or deceive CPs.

### C. Design Goals

In this paper, we are intended to design a secure access control framework for ICN with high efficiency and desired security features. Specially, the proposed scheme achieves the following design goals:

- **Data confidentiality.** The proposed scheme should make sure both unauthorized users and cache-enabled routers in ICN are incapable of learning something from the encrypted contents.
- **Privacy preservation.** Our designed scheme needs to protect users' private information like users' identities from being leaked to anyone during content retrieve process.
- **Privilege revocation.** The content providers in our scheme must be able to revoke any user's access privilege to contents without affecting other authorized users.
- **Accountability.** The requirement of accountability is that CPs are able to conduct service accounting to confirm the number of requests satisfied and extract useful information from service credentials to improve their services.
- **High efficiency.** The proposed scheme is required to be very efficient for users to obtain the desired contents. Besides, user revocation should be achieved without letting other users update their private keys.

## III. Preliminaries

### A. Bilinear Map

Let $G_1$, $G_2$, and $G_T$ be multiplicative cyclic groups of the same prime order of $p$, and $g_1$ and $g_2$ be generators of $G_1$ and $G_2$, respectively. A bilinear map is a map $e : G_1 \times G_2 \rightarrow G_T$ that has the following properties:

- *Computability*: there is an efficient algorithm to compute the map $e$;
- *Bilinearity*: for all $a, b \in \mathbb{Z}_p^*$ and $u \in G_1, v \in G_2$, $e(u^a, v^b) = e(u, v)^{ab}$;
- *Non-degeneracy*: $e(g_1, g_2) \neq 1$.

Our scheme implements group signature [20] and broadcast encryption [21] which work on the bilinear pairings with $q$-Strong Diffie-Hellman (SDH), Decision Linear Diffie-Hellman (DLIN), Weak Bilinear Diffie-Hellman Exponent (WBDHE), and Basic Diffe-Hellman Problem (BDHP) assumptions as follows:

*Definition 1: q-SDH Assumption:* Given a $(q+2)$-tuple $(g_1, g_2, {g_2}^{x}, {g_2}^{x^2}, \ldots, {g_2}^{x^q})$ as input, to output a pair $({g_1}^{1/(x+c)}, c)$ where $c \in \mathbb{Z}_p^*$ is difficult. Formally, we say an adversary $\mathcal{A}$ has a non-negligible advantage $\epsilon$ to solve q-SDH if

$$Pr\Big[\mathcal{A}(g_1, g_2, {g_2}^{x}, {g_2}^{x^2}, \ldots, {g_2}^{x^q}) = ({g_1}^{\frac{1}{x+c}}, c)\Big] \geq \epsilon.$$

*Definition 2: DLIN Assumption:* Given $(u, v, h, u^a, v^b, h^c) \in {G_1}^6$ as input, it is hard to distinguish whether $c = a + b$ or $c$ is random in $\mathbb{Z}_p^*$, even in bilinear groups where Decisional Diffie-Hellman (DDH) assumption can not hold. More precisely, we say that an adversary $\mathcal{A}$ has a non-negligible advantage $\epsilon$ to solve DLIN if

$$\bigg| Pr\Big[\mathcal{A}(u, v, h, u^a, v^b, h^{a+b}) = (u, v, h, u^a, v^b, h^{a+b})\Big] - Pr\Big[\mathcal{A}(u, v, h, u^a, v^b, \eta) = (u, v, h, u^a, v^b, h^{a+b})\Big] \bigg| \geq \epsilon,$$

where $\eta$ is random in $G_1$.

*Definition 3: WBDHE Assumption:* For unknown $a \in \mathbb{Z}_p^*$, given a tuple $(P,\ P^a,\ P^{a^2},\ \ldots,\ P^{a^l} \in G_1,\ Q \in G_2)$, it is infeasible to compute $e(P, Q)^{\frac{1}{a}}$.

*Definition 4: BDHP Assumption:* For unknown $x \in \mathbb{Z}_p^*$, given $P, P^x \in G_1$, it is infeasible to compute $x$ because of the discrete logarithm problem.

### B. Linear Encryption

Linear encryption is a securer extension of ElGamal encryption which can be secure even in groups where DDH can not hold. The algorithm procedure can be described as follows:

- *Key Generation:* A user's public key is three random elements $u, v, h \in G_1$ and his/her private key is two numbers $x, y \in \mathbb{Z}_p^*$ such that $u^x = v^y = h$.
- *Encryption:* For a message $M \in G_1$, a user chooses secure session keys $a, b \in \mathbb{Z}_p^*$ randomly, and the corresponding ciphertext is $(u^a, v^b, M \cdot h^{a+b})$.
- *Decryption:* Given a ciphertext $(T_1, T_2, T_3)$, a user can recover the message by computing $T_3/({T_1}^x \cdot {T_2}^y)$.

### C. Hash Chain

The property of hash function is that its forward computation is efficient and the backward computation is generally infeasible. Apply a one-way hash function $h(\cdot)$ to a random seed $s$ for $l-1$ times and we can get a hash chain of $l$ elements:

$$s,\ h(s),\ h^2(s),\ \cdots,\ h^{l-1}(s).$$

Hash chain has been used as a lightweight authentication method in many literatures, e.g., [22]. Suppose a user owns a hash chain and shares the last element $h^{l-1}(s)$ with the verifier who has already authenticated the user once, then the user can be simply authenticated again by showing $h^{l-2}(s)$ to the verifier because no one except the user can compute $h^{l-2}(s)$. By repeating this process, the user can be authenticated $l-1$ times before the hash chain is used up. In this paper, we denote a hash chain as $(H_{head}, H_{tail}, l)$ where $H_{head}$ is the first element of the hash chain, $H_{tail}$ is the last element of the hash chain, and $l$ is the length of the hash chain.

## IV. Construction of SEAF

### A. Overview

In our access control scheme, CPs manage their users by dividing them into groups. Users in different groups have different access privileges to contents. For example, VIP users can access more contents than normal registered users. Without loss of generality, CPs use numbers as group identifers and a larger group identifier means a higher access privilege. This manner of privilege management is efficient and has been widely applied.

We enforce a two-layer access control in our scheme for availability and robustness. Specifically, in CP side, we let CP encrypt the contents before publication through broadcast encryption to guarantee that only the users with corresponding access privileges can decrypt the contents. But only encryption is not enough considering that anyone can obtain contents from ICN network, which makes the system vulnerable to DoS attacks. So at edge router, group signature is introduced to allow edge routers to authenticate users without revealing their real identities, which ensures that the resources in ICN network are only available to authorized users. Besides, SEAF uses hash chains to relate continuous requests so that edge routers can replace most of the expensive signature verification operation with lightweight hash operation. Furthermore, signatures and hash chains generated during authentication can be stored as service credentials for future accounting.

For simplicity, the following description of our scheme considers only one content provider and its users, but it can be easily extended to the one with multiple content providers. The whole construction includes system setup, user registration, content generation, request authentication, content decryption, service accounting, and user revocation.

### B. System Setup

In this step, CP generates the necessary public and private parameters as a group manager. Assume that the users are divided into $m$ groups according to the access policy, CP initializes the system as follows:

- Generating a bilinear map group system $S = (p, G_1, G_2, G_T, e(\cdot,\cdot))$ with two randomly selected generators $g_1 \in G_1, g_2 \in G_2$ and $q = e(g_1, g_2)$;
- Selecting two random elements $h \in G_1, h' \in G_2$ and two random numbers $\xi_1, \xi_2 \in \mathbb{Z}_p^*$ and setting $u, v \in G_1, \mathcal{H}_1, \mathcal{H}_2 \in G_2$ such that $u^{\xi_1} = v^{\xi_2} = h, \mathcal{H}_1 = {h'}^{\xi_1}, \mathcal{H}_2 = {h'}^{\xi_2}$;
- Selecting $m$ random numbers $\gamma_1, \gamma_2, \ldots, \gamma_m \in \mathbb{Z}_p^*$ and setting $w_i = {g_2}^{\gamma_i}$ where $i = 1, 2, \ldots, m$. Denoting $\Gamma$ as $(\gamma_1, \gamma_2, \ldots, \gamma_m)$ and $W$ as $(w_1, w_2, \ldots, w_m)$;
- Selecting $m$ random numbers $\lambda_1, \lambda_2, \ldots, \lambda_m \in \mathbb{Z}_p^*$ and setting $y_i = {g_1}^{\lambda_i}$ where $i = 1, 2, \ldots, m$. Denoting $\Lambda$ as $(\lambda_1, \lambda_2, \ldots, \lambda_m)$ and $Y$ as $(y_1, y_2, \ldots, y_m)$;
- Publishing the public parameters including
  $$(S, g_2, h, h', u, v, q, \mathcal{H}_1, \mathcal{H}_2, W, Y, H_1, H_2, E(\cdot)),$$
  where $H_1$ is a one-way hash function: $(0,1)^* \to \mathbb{Z}_p^*$, $H_2$ is a hash function used to generate hash chains and $E_K(\cdot)$ is a secure symmetric encryption algorithm with the secret key $K$. At the same time, CP keeps $(\Gamma, \xi_1, \xi_2, \Lambda, g_1)$ as its master key.

### C. User Registration

For the registration of user $j$ with identity $ID_j$ to be a member of group $n$, CP randomly selects a number $x_j \in \mathbb{Z}_p^*$, computes

$$A_j = {g_1}^{1/(\gamma_n + x_j)}, \qquad (1)$$

and adds $(A_j, ID_j)$ into the user list. Then CP selects other $n$ random numbers $z_j^1, z_j^2, \ldots, z_j^n \in \mathbb{Z}_p^*$ and computes

$$b_j^k = {g_1}^{z_j^k/(\lambda_k + z_j^k)}, \quad d_j^k = {g_2}^{1/(\lambda_k + z_j^k)}, \qquad (2)$$

for $k = 1, 2, \ldots, n$. Denote $Z_j = (z_j^1, z_j^2, \ldots, z_j^n)$, $B_j = (b_j^1, b_j^2, \ldots, b_j^n)$, and $D_j = (d_j^1, d_j^2, \ldots d_j^n)$. After the registration, user $j$ gets its secret key $(x_j, A_j, Z_j, B_j, D_j)$ where $x_j$ and $A_j$ are used for signature generation, and $Z_j$, $B_j$ and $D_j$ are used to decrypt the contents with different access privilege levels.

### D. Content Generation

To make the access privilege level of the contents explicit, we propose a minor modification to the ICN naming mechanism. For example, a content has the name of */com/example/subdir/abc.mp4/chunk_1* where */com/example/* represents CP's domain name (example.com), *subdir/abc.mp4/* is the directory path of the file to which it belongs and *chunk_1* is used to specify the content in this file. If the content has an access privilege level of 3, then the content name is changed to */com/example/**3**/subdir/abc.mp4/chunk_1*. The inserted access privilege level label can help edge routers easily decide which users can access the content. This modification has no side effect except a little increase of the content name length.

Before CP's raw contents are disseminated into the network, CP uses broadcast encryption to preserve data confidentiality. We suppose that the content to be published is binded with the access privilege level label $n$, and then it can be accessed by the users in group $n, n+1, \ldots, m$. First, CP randomly selects $k \in \mathbb{Z}_p^*$, computes $K = q^k$ and encrypts content $M$ with $K$: $C = E_K(M)$. Then CP selects $y_n$ and encrypts the symmetric key $K$ by computing

$$C_1 = {y_n}^k, \quad C_2 = {g_2}^k. \tag{3}$$

So content $M$ is stored as $(C, C_1, C_2)$ in CP and some cache-enabled routers.

### *E. Request Authentication*

Edge routers authenticate users' requests based on group signatures and hash chains. Through a valid group signature, edge routers can verify that the user is authorized to access the first chunk of certain file. If the user exhibits the tail of a hash chain in the signature, he/she can authenticate themselves by revealing the generated hash chain to the edge router in reverse order one element a time for the following chunks. Due to the one-way property of hash function, the requests with correct hash chain elements are also regarded as authorized. The details are as follows.

When user $j$ requests the first chunk of a file, he/she generates a hash chain with the proper length $(H_{head}, H_{tail}, l)$ and sends an interest packet with the group identifier $n$, the filename $f$, the timestamp *TS*, the hash chain tail $H_{tail}$ and a signature $\sigma$. Note that the filename $f$ is a prefix of the chunk name. The signature $\sigma$ is generated as shown in **Algorithm 1**.

```
Algorithm 1 Signature Generation
Input: User $j$'s private key $(A_j, x_j)$, system parameters
       $(g_2, h, u, v, w_n)$, requested file name $f$, timestamp
       $TS$ and a hash chain tail $H_{tail}$.
Output: A valid group signature $\sigma$ on $f \parallel TS \parallel H_{tail}$.
Select random numbers $\alpha, \beta, r_\alpha, r_\beta, r_x, r_{\delta_1}, r_{\delta_2} \in \mathbb{Z}_p^*$;
Set $M' = f \parallel TS \parallel H_{tail}$;
Set $\delta_1 = x_j\alpha,\ \ \delta_2 = x_j\beta$,
    $T_1 = u^\alpha,\ T_2 = v^\beta,\ T_3 = A_j h^{\alpha+\beta}$;
Set $R_1 = u^{r_\alpha},\ R_2 = v^{r_\beta}$,
    $R_3 = e(T_3, g_2)^{r_x} e(h, w_{n_j})^{-r_\alpha - r_\beta} e(h, g_2)^{-r_{\delta_1} - r_{\delta_2}}$,
    $R_4 = T_1^{r_x} u^{-r_{\delta_1}},\ R_5 = T_2^{r_x} v^{-r_{\delta_2}}$;
Set $c = H_1(M', T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$;
Set $s_\alpha = r_\alpha + c\alpha,\ s_\beta = r_\beta + c\beta$,
    $s_x = r_x + cx_j,\ \ s_{\delta_1} = r_{\delta_1} + c\delta_1,\ s_{\delta_2} = r_{\delta_2} + c\delta_2$;
return $\sigma = (T_1, T_2, T_3, R_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$;
```

After receiving the interest packet, the edge router first checks the validity of the timestamp *TS* and then verifies the signature $\sigma$ using **Algorithm 2**. The public parameters for the verification are selected according to the group identifier $n$. If the signature is valid, the edge router believes that the user is a legitimate user of group $n$. Then the edge router extracts the access privilege level label $n'$ from the chunk name and compares it with $n$. If $n \geq n'$, it means that the user has permission to the chunk and the edge router injects the request into the ICN network. Otherwise, the edge router discards the request.

```
Algorithm 2 Signature Verification
Input: System parameters $(g_2, h, h', u, v, w_n)$, signed
       info $f, TS, H_{tail}$ and a signature $\sigma$.
Output: Valid or Invalid.
Set $M' = f \parallel TS \parallel H_{tail}$;
Set $R_1 = u^{s_\alpha} T_1^{-c},\ \ R_2 = v^{s_\beta} T_2^{-c}$,
    $R_4 = T_1^{s_x} u^{-s_{\delta_1}},\ \ R_5 = T_2^{s_x}\ v^{-s_{\delta_2}}$,
    $t_1 = -s_\alpha - s_\beta,\ \ t_2 = -s_{\delta_1} - s_{\delta_2}$;
if $R_3 \neq e(T_3, g_2)^{s_x} e(h, w_n)^{t_1} e(h, g_2)^{t_2} (\frac{e(T_3, w_n)}{e(g_1, g_2)})^c$ then
    return Invalid;
else if $c = H_1(M', T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$ then
    return Valid;
end
return Invalid;
```

In order to authenticate requests for the chunks of the same file with hash chains, the edge router maintains an authentication state table (AST). The AST has three fields: filename $f$, last received hash value $H_{last}$, and a counter $l$. After the edge router validates a signature, it adds a new entry to the table and the fields are initialized to the received filename $f$, the hash chain tail $H_{tail}$, and "1".

When the user requests other chunks of the file, he/she only needs to attach the new element $H_{new}$ of the hash chain to the interest packet. The edge router extracts filename $f'$ from the content name and computes $H_2(H_{new})$. Then the edge router searches tuple $< f',\ H_2(H_{new}) >$ in the AST table to find an entry satisfying $f = f'$ and $H_{last} = H_2(H_{new})$. If the entry is found, the edge router updates the hash value of the entry to $H_{new}$ and increases the value in the counter field.

When the user retrieves all the chunks of the file or loses interest halfway, he/she can send a termination message with the filename and a new hash chain element. Similarly, the edge router tries to locate the corresponding entry. If the entry exists, the edge router stores a credential, which includes the three fields in the entry (i.e., $< f, H_{last}, l >$), the timestamp *TS*, the group identifier $n$, the signature $\sigma$, and the hash chain tail $H_{tail}$ received at the beginning. The storage should be in the ascending order of the timestamp *TS* so that duplicate credentials can be easily detected.

### *F. Content Decryption*

When the requested content $(C, C_1, C_2)$ is obtained, user $j$ first selects the right decryption key $(b_j^{n'}, d_j^{n'})$ according to the access label $n'$ in the content name. Then user $j$ recovers $K$ from $C_1$ and $C_2$ as follows:

$$\begin{aligned} e(C_1, d_j^{n'}) e(b_j^{n'}, C_2) &= e(g_1, g_2)^{\frac{k\lambda_{n'}}{\lambda_{n'} + z_j^{n'}}} e(g_1, g_2)^{\frac{k \cdot z_j^{n'}}{\lambda_{n'} + z_j^{n'}}} \\ &= q^k = K. \end{aligned} \tag{4}$$

After that, user $j$ can decrypt $C$ to get content $M$ with the symmetric encryption algorithm $E(\cdot)$ and recovered $K$.

### G. Service Accounting

To prove the amount of served requests and provide feedback for CP, the edge routers send the stored service credentials to CP periodically. The service credentials have the form of

$$< f, TS, H_{tail}, n, \sigma, H_{last}, l >,$$

where $f$ is the requested filename, *TS* is the time when the user starts to request the file, $H_{tail}$ is the tail of the received hash chain, $n$ is the identifier of the user's group, $\sigma$ is the user's signature on $f \parallel TS \parallel H_{tail}$, $H_{last}$ is the last received hash chain element, and $l$ is the length of the received hash chain indicating how many chunks of the file have been requested.

Before paying bills to ISP and extracting feedback information, CP needs to verify all the service credentials. For every credential, CP first checks the validity of *TS* and whether $H_{tail}$ equals $H_2^{l-1}(H_{last})$. If the cases are satisfied, CP is required to verify the group signatures in the credentials. To reduce the huge computation overhead in CP for verifying the signatures and accelerate the process, we provide the batch verification algorithm. Besides, CP can divide the credentials into $m$ subsets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m$ according to group identifier $n$. CP conducts a probabilistic verification by selecting a random subset of the credentials in a certain proportion, and for the chosen subset, CP verifies the group signatures using **Algorithm 3**. As long as the credentials in the chosen subset can make a successful verification, CP can trust that all the credentials are legitimate.

**Algorithm 3** Batch Verification

```
Input: System parameters $(g_2, h, u, v, w_n)$ and $\|\mathcal{S}_n\|$
       service credentials
       $< f_i, TS_i, H_{i,tail}, n, \sigma_i, H_{i,last}, l_i >$ where $\sigma_i =$
       $(T_{i,1}, T_{i,2}, T_{i,3}, R_{i,3}, c_i, s_{i,\alpha}, s_{i,\beta}, s_{i,x}, s_{i,\delta_1}, s_{i,\delta_2})$
       for $i = 1, 2, \ldots, \|\mathcal{S}_n\|$.
Output: Valid or Invalid.
Set $P_1 = P_4 = Q = 1$, $P_2 = P_3 = P_5 = 0$;
for $i = 1$ to $\|\mathcal{S}_n\|$ do
    Set $M_i' = f_i \parallel TS_i \parallel H_{i,tail}$,
        $R_{i,1} = u^{s_{i,\alpha}} {T_1}^{-c_i}$, $R_{i,2} = v^{s_{i,\beta}} {T_2}^{-c_i}$,
        $R_{i,4} = {T_{i,1}}^{s_{i,x}} u^{-s_{i,\delta_1}}$, $R_{i,5} = {T_{i,2}}^{s_{i,x}} v^{-s_{i,\delta_2}}$,
        $c_i' =$
    $H_1(M_i', T_{i,1}, T_{i,2}, T_{i,3}, R_{i,1}, R_{i,2}, R_{i,3}, R_{i,4}, R_{i,5})$;
    if $c_i \neq c_i'$ then
        return Invalid;
    end
    Set $P_1 = P_1 {T_{i,3}}^{s_{i,x}}$,  $P_4 = P_4 {T_{i,3}}^{c_i}$,
        $P_2 = P_2 - s_{i,\alpha} - s_{i,\beta}$,  $P_3 = P_3 - s_{i,\delta_1} - s_{i,\delta_2}$,
        $P_5 = P_5 - c_i$,  $Q = QR_{i,3}$;
end
Set $Q' = e(P_1 \cdot h^{P_3} \cdot {g_1}^{P_5}, g_2) e(h^{P_2} \cdot P_4, w_n)$;
if $Q' \neq Q$ then
    return Invalid;
end
return Valid;
```

After the verification, CP pays bills to ISP according to the sum of $l$ in all of the credentials, which equals to the amount of served requests.

TABLE I
THE REVOCATION LIST

| Group Identifer | Revocation Parameter |
|---|---|
| $n_1$ | $RP_1$ |
| $n_2$ | $RP_2$ |
| $n_2$ | $RP_3$ |

Then CP processes all credentials to reveal the signers' identities. For a partial credential $< f, TS, \sigma, l >$, CP computes

$$A = T_3/({T_1}^{\xi_1} \cdot {T_2}^{\xi_2}) \tag{5}$$

with $\xi_1$ and $\xi_2$ and gets the signer's identity $ID_j$ by looking up $A$ in the user list. Therefore, the partial credentials can be transformed into the form of $< f, TS, ID_j, l >$, which is equivalent to users' access record. With data analysis techniques applied, CP can extract important information such as users' preferences and content popularity, which can be very useful for the improvement of CP's service quality.

### H. User Revocation

A trial solution of user revocation is letting CPs update new contents encrypted with new keys when revocation occurs. However, this solution, introduced in many existing schemes, such as [23] and [24], brings too much overhead to update private keys when the number of authorized users or revoked users is large. Therefore, it is unsuitable for ICN where users are in quantity. Here, to avoid the significant overhead for updating private keys for all authorized users, we achieve user revocation by letting edge routers conduct revocation verification to make revoked users unable to pass the authentication at the very beginning. For the sake of simplicity, we do not consider the situation that revoked users collude with authorized users or ISP to obtain encrypted contents.

In our access control scheme, if users' accounts run out of money or users abuse their privileges (e.g., flooding interests) causing some damage to CPs, CPs have rights to revoke the users' privileges, making their requests unable to enter ICN network. User revocation is implemented by maintaining a revocation list like TABLE I, and the revocation list is public and can be obtained by edge routers and users.

The revocation list is something like a blacklist of our system, which is used by edge routers in request authentication process. Once a user is added into the revocation list, it means that he/she loses the opportunity to request any content from the corresponding CP. Due to the anonymous authentication, edge routers are incapable of knowing users' identities, so we can not achieve revocation by recording the revoked users' identities and certificates like traditional PKI architecture. Here we record a revocation parameter in the revocation list, which is a bilinear map value related to the signature key $A_i$ of the revoked user satisfying $RP_i = e(A_i, h')$.

In our system, to better reflect the revocation information, we let CP update the revocation lists every day. Besides, when users or edge routers request the revocation list, CP will bind the revocation list with a signature used to verify its validity. Here CP uses the BLS signature proposed in [25] to compute the signature as: $Sig = f(RL)^{\xi_1}$, where $f$ is a hash function: $\{0,1\}^* \rightarrow G_2$. The edge routers can easily verify the integrity of the revocation list by checking $e(h, f(RL)) \overset{?}{=} e(u, Sig)$ after receiving it.

To make the privilege revocation mechanism work effectively, we need to modify our scheme on request authentication phase. In this phase, the edge router is required to check whether the user is added into the revocation list before verifying the signature $\sigma$, as shown in **Algorithm** 4.

**Algorithm 4** Revocation Verification

```
Input: System parameters $(h', \mathcal{H}_1, \mathcal{H}_2)$, a signature $\sigma$
       and the revocation list $RL$.
Output: Valid or Invalid.
Set $temp = e(T_1, \mathcal{H}_1)e(T_2, \mathcal{H}_2)/e(T_3, h')$;
for $i = 1$ to $num(RL)$ do
    if $RL.RP_i = temp$ then
        return Invalid;
    end
end
return Valid;
```

If the edge router finds an item $RP_i$ that $RP_i = temp$ by traversing the revocation list, it rejects the request as the user's privilege has been revoked. It is noted that our scheme can achieve user revocation efficiently without affecting other authorized users, which means authorized users need not to update their private keys to support user revocation.

## V. Security Analysis

In this section, we first analyze the security features of our scheme in terms of data confidentiality, anonymity, traceability, unforgeability, replay resistance, and accountability. Then we compare our scheme with several representative access control schemes in some important aspects.

### A. Data Confidentiality

Our proposed scheme protects data confidentiality from both malicious routers and unauthorized users. Especially, edge routers prevent unauthorized users from obtaining contents by demanding a valid group signature. Here we consider the situation in which the routers are compromised.

*Lemma 1:* Unauthorized entities such as routers and unauthorized users including revoked users cannot learn any information from the encrypted contents.

*Proof:* As shown in Eq. (3), a content $M$ is stored as $(C_1, C_2, C)$ where $C_1 = {y_n}^k$, $C_2 = {g_2}^k$, $K = q^k$, and $C = E_K(M)$. Suppose that routers or unauthorized users can compute $K = q^k$, i.e., given $C_1 = {g_1}^{k\lambda_n}$, $C_2 = {g_2}^k$ and $g_2$, without the knowledge of $\lambda_n$, they can compute

$$e(C_1, g_2)^{\frac{1}{\lambda_n}} = e(g_1, g_2)^k = K.$$

This obviously contradicts WBDHE assumption.

Besides, edge routers can get some private key information from the revocation list. But without obtaining the decryption keys $B_i$ and $D_i$, edge routers are unable to get the symmetric key $K$ by using the content decryption algorithm.

In our system, user revocation is implemented by maintaining a revocation list and the security of the revocation list guarantees that revoked users cannot obtain the encrypted contents to threaten the data confidentiality. The revocation list is signed by CP as: $Sig = f(RL)^{\xi_1}$. Suppose that a revoked user can forge the signature, which means that given $u, h = u^{\xi_1}$, he/she can compute $\xi_1$. However, this contradicts BDHP assumption. So any revoked user cannot pass the revocation verification to obtain the encrypted contents by modifying the revocation list.

In conclusion, the correctness of Lemma 1 can be ensured. ■

### B. Anonymity

*Lemma 2:* If DLIN assumption holds on group $G_1$, the Linear encryption is semantically secure.

*Proof:* Suppose we have an adversary $\mathcal{A}$ with a non-negligible advantage $Adv_{\mathcal{A}}$ to break the semantical security of Linear encryption. Then we can find an algorithm $\mathcal{B}$ that plays DLIN game with a non-negligible advantage as follows.

**Setup.** The challenger $\mathcal{C}$ of the DLIN game tosses a secure random coin $r \in (0, 1)$ and generates a tuple $(u, v, h, u^a, v^b, Z)$, where $u, v, h \in G_1$ and $a, b \in \mathbb{Z}_p^*$. If $r = 0$, $\mathcal{C}$ sets $Z = h^{a+b}$; otherwise, $Z$ is set as $h^c$ for random $c \in \mathbb{Z}_p^*$. Then $\mathcal{C}$ sends the tuple to $\mathcal{B}$.

**Challenge.** $\mathcal{A}$ submits two challenge message $m_0, m_1$ to $\mathcal{B}$. Then $\mathcal{B}$ tosses a secure coin $b \in (0, 1)$ and returns the ciphertext $(u^a, v^b, m_b \cdot Z)$.

**Guess.** Eventually, $\mathcal{A}$ outputs its guess $b'$ of $b$. If $b = b'$, $\mathcal{B}$ outputs the guess $r' = 0$ to indicate that $Z = h^{a+b}$; otherwise $\mathcal{B}$ outputs $r' = 1$ to indicate that $Z$ is random in $G_1$.

When $r = 1$, i.e. $Z$ is random in $G_1$, the challenge ciphertext is independent of $b$ and $\mathcal{A}$ can not get any information on it. So we have $Pr[b' = b | r = 1] = \frac{1}{2}$. Otherwise when $r = 0$, i.e. $Z = h^{a+b}$, the challenge ciphertext is a valid encryption of $m_b$ and $\mathcal{A}$ has an advantage $Adv_{\mathcal{A}}$ to break the semantical security of Linear encryption. So we have $Pr[b' = b | r = 0] = \frac{1}{2} + Adv_{\mathcal{A}}$. Since $r' = 0$ when $b' = b$, we have $Pr[r' = r | r = 0] = \frac{1}{2} + Adv_{\mathcal{A}}$ and $Pr[r' \neq r | r = 1] = \frac{1}{2}$. Finally, we can get the overall advantage of $\mathcal{B}$ in DLIN game:

$$\begin{aligned} Adv_{\mathcal{B}} &= \Big| Pr[r' = r | r = 0] - Pr[r' \neq r | r = 1] \Big| \\ &= \Big| (\frac{1}{2} + Adv_{\mathcal{A}}) - \frac{1}{2} \Big| = Adv_{\mathcal{A}}. \end{aligned}$$

We can conclude that if there is an adversary who can break the semantical security of Linear encryption with advantage $Adv_{\mathcal{A}}$, an adversary can be found in DLIN game with a non-negligible advantage $Adv_{\mathcal{A}}$. So the lemma is proved. ■

*Lemma 3:* The group signature in our scheme is anonymous if Linear encryption is semantically secure on $G_1$.

*Proof:* Suppose there is an adversary $\mathcal{A}$ that can break the anonymity of the group signature with a non-negligible advantage $Adv_{\mathcal{A}}$. Now we show how we construct an algorithm $\mathcal{B}$ that breaks the semantical security of Linear encryption with a non-negligible advantage.

**Setup.** The challenger $\mathcal{C}$ of the Linear encryption security game sends a public key $(u, v, h) \in G_1^3$ to $\mathcal{B}$. Then $\mathcal{B}$ generates the remaining elements in the group public key by following the steps in Section IV and sends the group public key $(g_2, h, u, v, W)$ to $\mathcal{A}$. Besides, there is a hash function $H_1$ programmed as a random oracle by maintaining a table.

**Hash Queries.** Considering that $\mathcal{A}$ asks for the hash value $H_1(M', T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$, if it has already been defined in the table, then $\mathcal{B}$ returns the value in the table. Otherwise, $\mathcal{B}$ chooses a random number $k \in \mathbb{Z}_p^*$, inserts $k$ into the table, and returns $k$ to $\mathcal{A}$.

**Private Key Queries.** When adversary $\mathcal{A}$ requests for user $i$'s private key, $\mathcal{B}$ sends $(A_i, x_i)$ back to it.

**Challenge.** $\mathcal{A}$ submits its challenge, including two indexes, $i_0$ and $i_1$, and a message $M'$, to $\mathcal{B}$. After that, $\mathcal{B}$ also submits its challenge by providing two private keys, $A_{i_0}$ and $A_{i_1}$, to $\mathcal{C}$.

Then $\mathcal{C}$ tosses a secure coin $b \in (0, 1)$ and returns the ciphertext $(T_1, T_2, T_3)$ of the private key $A_{i_b}$ to $\mathcal{B}$. According to the received ciphertext, $\mathcal{B}$ generates the corresponding signature as follows:

Supposing user $i_b$ belongs to group $n$, $\mathcal{B}$ chooses six random number $c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2} \in \mathbb{Z}_p^*$. Then it computes $R_1, R_2, R_3, R_4, R_5$ as:

$$\begin{aligned} R_1 &= u^{s_\alpha} \cdot T_1{}^{-c}, R_2 = v^{s_\beta} \cdot T_2{}^{-c}, \\ R_4 &= T_1{}^{s_x} \cdot u^{-s_{\delta_1}}, R_5 = T_2{}^{s_x} \cdot v^{-s_{\delta_2}}, \\ t_1 &= -s_\alpha - s_\beta, t_2 = -s_{\delta_1} - s_{\delta_2}, \\ R_3 &= e(T_3, g_2)^{s_x} e(h, w_n)^{t_1} e(h, g_2)^{t_2} (\frac{e(T_3, w_n)}{e(g_1, g_2)})^c. \end{aligned} \tag{6}$$

And $\mathcal{B}$ sets the value $H_1(M', T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$ as $c$. If there is a collision, i.e., the oracle at this point has been programmed as some other value, $\mathcal{B}$ reports failure and aborts. Otherwise, it sends the group signature $(T_1, T_2, T_3, R_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$ to $\mathcal{A}$.

**Guess.** Finally, $\mathcal{A}$ outputs its guess $b'$ of b. Algorithm $\mathcal{B}$ also gives its guess $b'$ to its challenger $\mathcal{C}$.

Since the group signature of user $i_b$ is derived from the Linear encryption of $A_{i_b}$, $\mathcal{B}$ gets a correct guess whenever $\mathcal{A}$ does. The adversary $\mathcal{A}$ can break the anonymity of the group signature with a non-negligible advantage $Adv_{\mathcal{A}}$. So it is easy for us to conclude that $\mathcal{B}$ can break the semantical security of Linear encryption with the same non-negligible advantage, i.e., $Adv_{\mathcal{B}} = Adv_{\mathcal{A}}$. The lemma is proved. ■

According to Lemma 2 and Lemma 3, we can say that the group signature in our scheme is anonymous if DLIN assumption holds on group $G_1$. In other words, inheriting from the group signature, the authentication messages will not leak any information about the signer's identity.

### *C. Traceability*

*Lemma 4:* If q-SDH assumption holds on $(G_1, G_2)$, then the traceability can be guaranteed in our system.

*Proof:* Suppose an adversary $\mathcal{A}$ can break the traceability of our system. Then we will show how to interact with $\mathcal{A}$ to break the q-SDH assumption.

**Setup.** The system generates the system parameter $(S, g_1, g_2, W)$ as shown in section IV-B and gives them to the challenger. Meanwhile, it generates a $(q + 2)$-tuple $(g_1, g_2, g_2{}^{\gamma_n}, g_2{}^{\gamma_n^2}, \ldots, g_2{}^{\gamma_n^q})$ and a set of pairs $(A_i, x_i)$ for $i = 1, \ldots, q - 1$, which are also given to the challenger. Some of the pairs are the SDH pairs satisfying $e(A_i, w_n g_2{}^{x_i}) = e(g_1, g_2)$, where $n$ is the group identifer of the users. The others are the pairs where the $x_i$ corresponding to $A_i$ is unknown. Then the challenger picks a $h \in G_1$ randomly and chooses two random numbers $\xi_1, \xi_2 \in \mathbb{Z}_p^*$ making $u^{\xi_1} = v^{\xi_2} = h$. Finally, the challenger gives the adversary $\mathcal{A}$ public key $(S, g_1, h, u, v, W)$ and master key $(g_2, \xi_1, \xi_2)$. Besides, there is a hash function $H_1$ considered as a random oracle by maintaining a table.

**Hash Queries.** When adversary $\mathcal{A}$ queries the hash value of $(M', T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$, if it has the corresponding entry in the table, then the challenger returns the value in the entry. Otherwise, the challenger chooses a number $k \in \mathbb{Z}_p^*$ randomly, inserts $k$ into the table and returns $k$ to $\mathcal{A}$.

**Signature Queries.** When $\mathcal{A}$ asks for a signature on message $M'$ with index $i$, if $x_i$ belongs to an SDH pair, the challenger generates a signature $\sigma$ with key $(A_i, x_i)$ using the signature generation algorithm and returns to $\mathcal{A}$. Otherwise, the challenger picks $\alpha, \beta \in \mathbb{Z}_p^*$ randomly and sets $T_1 = u^\alpha, T_2 = v^\beta, T_3 = A_i g_1{}^{\alpha+\beta}$. Afterwards, the challenger chooses six random number $c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2} \in \mathbb{Z}_p^*$. Then it computes $R_1, R_2, R_3, R_4, R_5$ using Eq.(6).

Finally, the challenger sets the hash oracle of $(M', T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$ as $c$. If there is a collision, the challenger reports failure and aborts. Otherwise, it sends the group signature $(T_1, T_2, T_3, R_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$ to $\mathcal{A}$.

**Private Key Queries.** The challenger returns $(A_i, x_i)$ when $\mathcal{A}$ asks for the private key of user $i$, if the private key is an SDH pair. Otherwise, it reports failure and aborts.

**Output.** If adversary $\mathcal{A}$ succeeds in breaking the traceability, it outputs a forged group signature $\sigma = (T_1, T_2, T_3, R_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$ on a Message $M'$ with $c = H_1(M', T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$, which can be verified correctly. The challenger uses the master key, $\xi_1, \xi_2$, to trace the signature by computing $A = T_3/(T_1{}^{\xi_1} \cdot T_2{}^{\xi_2})$. If $A$ doesn't equal to any $A_i$ in the set of pairs $(A_i, x_i)$, the challenger outputs $\sigma$. Otherwise, $A = A_j$, where $j$ is an integer between 1 to $q - 1$. If the corresponding $x_j$ is unknown, the challenger outputs $\sigma$, and if not, the challenger reports failure and aborts.

According to the Forking Lemma proposed by David Pointcheval and Jacques Stern in [26], the challenger can get another forgery $(T_1, T_2, T_3, R_3, c', s'_\alpha, s'_\beta, s'_x, s'_{\delta_1}, s'_{\delta_2})$ on message $M'$ with a non-negligible probability, where it has the same $T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5$ as the former forgery.

We can learn that the two forged signatures satisfy the following equations from the signature verification algorithm:

$$R_1 = u^{s_\alpha} T_1{}^{-c} = u^{s'_\alpha} T_1{}^{-c'}, \tag{7}$$

$$R_2 = v^{s_\beta} T_2{}^{-c} = v^{s'_\beta} T_2{}^{-c'}, \tag{8}$$

$$R_4 = T_1{}^{s_x} u^{-s_{\delta_1}} = T_1{}^{s'_x} u^{-s'_{\delta_1}}, \tag{9}$$

$$R_5 = T_2{}^{s_x} v^{-s_{\delta_2}} = T_2{}^{s'_x} v^{-s'_{\delta_2}}. \tag{10}$$

$$\begin{aligned} R_3 &= e(T_3, g_2)^{s_x} e(h, w_n)^{-s_\alpha - s_\beta} \\ &\quad \times e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} (\frac{e(T_3, w_n)}{e(g_1, g_2)})^c, \\ &= e(T_3, g_2)^{s'_x} e(h, w_n)^{-s'_\alpha - s'_\beta} \\ &\quad \times e(h, g_2)^{-s'_{\delta_1} - s'_{\delta_2}} (\frac{e(T_3, w_n)}{e(g_1, g_2)})^{c'}, \end{aligned} \tag{11}$$

From Eq. (7), the challenger can obtain the following equation

$$T_1 = u^{\Delta s_\alpha / \Delta c}, \tag{12}$$

by dividing the two instances in it, where we denote $\Delta s_\alpha = s_\alpha - s'_\alpha, \Delta c = c - c'$. Similarly, from Eq. (8), the challenger obtain the equation

$$T_2 = v^{\Delta s_\beta / \Delta c}. \tag{13}$$

Considering Eq. (9), the challenger get $T_1{}^{\Delta s_x} = u^{\Delta s_{\delta_1}}$ by dividing the two instances in the equation. Combining with Eq. (12), the challenger can obtain

$$(\Delta s_\alpha / \Delta c) \Delta s_x = \Delta s_{\delta_1}. \tag{14}$$

In the same way, from Eq. (10), the challenger deduces that

$$(\Delta s_\beta / \Delta c) \Delta s_x = \Delta s_{\delta_2}. \tag{15}$$

Finally, the challenger divides the two instances in Eq. (11) and get the following equation combing with Eq. (12) $\sim$ Eq. (15):

$$e(g_1, g_2) = e(T_3\ h^{-\Delta s_\alpha/\Delta c - \Delta s_\beta/\Delta c}, w_n g_2{}^{\Delta s_x/\Delta c}),$$

where we denote $\widehat{A} = T_3\ h^{-\Delta s_\alpha/\Delta c - \Delta s_\beta/\Delta c}$, $\hat{x} = \Delta s_x/\Delta c$ respectively. We can see that the challenger may obtain an SDH pair $(\widehat{A}, \hat{x})$ according to the forged signatures generated by adversary $\mathcal{A}$.

Therefore, if there exits an adversary can break the traceability of our system with a non-negligible probability, we can find an algorithm to solve the q-SDH problem. The lemma is proved. ■

### *D. Unforgeability*

*Lemma 5:* If q-SDH assumption holds on $(G_1, G_2)$, there is no unauthorized user able to forge the authentication message to access the network.

*Proof:* Suppose an adversary $\mathcal{A}$ succeeds to forge a valid group signature with a non-negligible probability in polynomial time. We also assume that $H_1$ is a random oracle. Then adversary $\mathcal{A}$ can obtain two valid signatures $(M', \delta_0, c, \sigma_1)$ and $(M', \delta_0, c', \sigma_1')$ as follows:

$$\begin{cases} \delta_0 = (T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5), \\ c = H_1(M', T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5), \\ c' = H_1'(M', T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5), \\ \sigma_1 = (s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2}), \\ \sigma_1' = (s_\alpha', s_\beta', s_x', s_{\delta_1}', s_{\delta_2}'), \end{cases}$$

where the elements hold the following equations:

$$\begin{cases} s_\alpha = r_\alpha + c\alpha, & s_\alpha' = r_\alpha + c'\alpha, \\ s_\beta = r_\beta + c\beta, & s_\beta' = r_\beta + c'\beta, \\ s_x = r_x + cx, & s_x' = r_x + c'x, \\ s_{\delta_1} = r_{\delta_1} + c\delta_1, & s_{\delta_1}' = r_{\delta_1} + c'\delta_1, \\ s_{\delta_2} = r_{\delta_2} + c\delta_2, & s_{\delta_2}' = r_{\delta_2} + c'\delta_2. \end{cases}$$

The probability that $c = c'$ can be omitted. Thus, $\mathcal{A}$ can compute an SDH tuple $(\hat{x} = \frac{s_x - s_x'}{c - c'}, \hat{A} = T_3/h^{\frac{s_\alpha + s_\beta - s_\alpha' - s_\beta'}{c - c'}})$, such that $\hat{A} = g_1{}^{1/(\gamma_n + \hat{x})}$. Obviously, this contradicts q-SDH assumption. ■

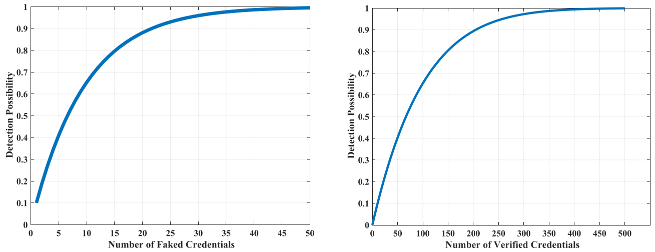
### *E. Replay Resistance*

*Lemma 6:* No adversary can proceed a replay attack to gain unauthorized access.

*Proof:* Since a timestamp *TS* is included in the request, the request will be invalid when the replay attack is launched. If an adversary attempts to alter *TS*, he/she will have to alter the signature $\sigma$ as well. This is equivalent to forging a valid signature, which has been proved infeasible in Lemma 5. ■

### *F. Accountability*

Our proposed scheme provides a novel accounting mechanism for CPs. On the one hand, CPs can confirm how many of their users' requests have been served (forwarded or satisfied). On the other hand, CPs can gather necessary feedback information through the mechanism, such as content popularity and users' preferences.



(a) (b)

Fig. 2. Detection possibility with different parameters. (a) Detection possibility vs. number of faked credentials ($\beta = 0.1$). (b) Detection possibility vs. number of verified credentials ($\alpha = 0.01$).

Since edge routers authenticate users' requests by group signatures and hash chains, the amount of the served request is equal to the sum of the lengths of the hash chains. Because hash chain tails are included in the group signatures, the validity of the hash chains are dependent on the signatures. And as analyzed above, forging valid group signatures is infeasible. Hence, after verifying the signatures, CPs can get the accurate amount of served requests by adding up the lengths of hash chains.

The signed information in group signature includes file name $f$, timetamp *TS*, and hash chain tail $H_{tail}$. Based on the traceability of group signature, CPs can extract signers' real identities. As a result, from every pair of group signature and hash chain, CPs can learn a piece of message about which user requests which file at what time for how many chunks. By analyzing these messages, CPs can obtain information about content popularity and users' preferences.

However, to reduce the computation overhead in CPs, the verification conducted by CPs is probabilistic by selecting a random subset of the credentials to verify. The edge routers may risk generating some faked credentials to get more profits. In our system, the edge routers dare not do this and we will prove it as follows.

*Lemma 7:* When CPs verify the credentials with a non-negligible probability, edge routers dare not risk generating faked credentials to get more profits.

*Proof:* Consider that edge routers generate faked credentials with the proportion $\alpha$ and CPs verify the credentials with the probability $\beta$. Then the detection possibility $p$ that CPs succeed in finding the misbehavior of edge routers is as follows:

$$p = \begin{cases} 1, & if\ \beta \geq 1 - \alpha, \\ 1 - \binom{n(1-\alpha)}{n\beta} / \binom{n}{n\beta}, & if\ \beta < 1 - \alpha. \end{cases}$$

where $n$ is the number of received credentials. Even when edge routers generate one faked credential each time, the detection possibility is:

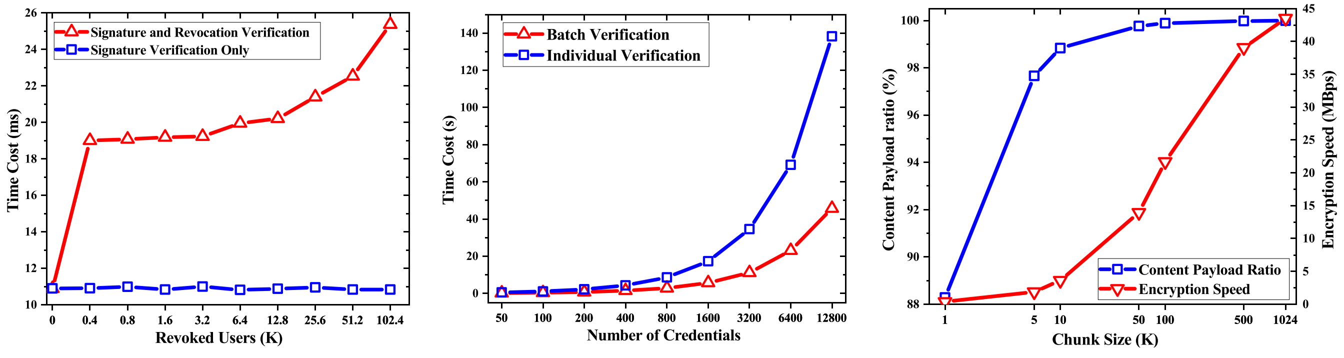
$$p = 1 - \binom{n-1}{n\beta} / \binom{n}{n\beta} = \beta,$$

which is sufficient to deter edge routers.

To give a deeper understanding, we visualize how detection possibility changes with $\alpha$ and $\beta$. Suppose there are $n = 1000$ credentials. Fig. 2(a) shows result when CPs chooses 100 credentials to verify, i.e., $\beta = 0.1$. The detection

TABLE II
COMPARISON WITH OTHER ACCESS CONTROL SCHEMES

| Scheme | Data Confidentiality | DoS Resistance | Offline CP | Privilege Revocation | Privacy Protection | Accountability |
|---|---|---|---|---|---|---|
| DACPI [4] | No | Yes | No | No | No | Yes |
| AccConf [7] | Yes | No | Yes | Yes | Yes | No |
| FTP-NDN [27] | Yes | Yes | Yes | No | No | No |
| SEAF | Yes | Yes | Yes | Yes | Yes | Yes |



(a) (b) (c)

Fig. 3. The result of algorithm implementation. (a) Verification time cost vs. the number of revoked users. (b) Verification time cost vs. the number of credentials. (c) Encryption speed and content payload ratio vs. chunk size.

possibility increases rapidly when the number of faked credentials grows. Even though edge routers only forge a small proportion of credentials like $\alpha = 0.005$, CPs can detect the misbehavior with a high possibility about 40%. Fig. 2(b) illustrates the detection possibility when edge routers generate 10 ($\alpha = 0.001$) credentials. With the growing number of verified credentials, the detection possibility increases quickly and nearly reaches 1 when $\beta = 0.35$. It can easily reaches a high possibility when CPs choose a very small part of credentials to verify.

Such a high risk of being detected is sufficient to stop edge routers owned by a reputed company from generating faked credentials to earn more profits. ■

### G. Comparison

We further compare the proposed SEAF with several other representative access control schemes in the aspects of data confidentiality, DoS resistance, offline CP, privilege revocation, privacy protection and accountability. As summarized in TABLE II, only our proposed SEAF can achieve all of these security features simultaneously.

## VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our access control scheme through algorithm implementation and network simulation. First, using GNU Multiple Precision Arithmetic(GMP) library[1] and Pairing-Based Cryptography(PBC) library,[2] we implement the described broadcast encryption, group signature, and privilege revocation mechanism. Then we evaluate the computation and storage overhead for CP and routers crosswise. Finally, we use NS-3 and ndnSIM [28] to simulate our protocol integrated in standard NDN and show that SEAF only introduces slight content retrieval delay.

TABLE III
COMPUTATION COST FOR CRYPTOGRAPHIC OPERATIONS

| Category | Operation | Time (ms) |
|---|---|---|
| Signature-related | Generation (Without Precomputation) | 8.4 |
| | Generation (With Precomputation) | 0.03 |
| | Verification | 10.8 |
| | Batch Verification | 3.5 |
| | Opening | 0.8 |
| | SHA-256 | $< 10^{-4}$ |
| Encryption-related | Broadcast Encryption | 2.5 |
| | Broadcast Decryption | 1.5 |
| | AES-256 | 0.02 (1K) |

All the experiments are conducted on a Linux system (Ubuntu 16.04 LTS) with a 3.6GHz Intel Core i7 processor and 20G RAM.

### A. Algorithm Implementation

Group signature and broadcast encryption are both implemented using an elliptic curve with 160-bit group order, which offers approximately the same security level with 1024-bit RSA. Because of the necessity of symmetric encryption and hash function, we also test AES-256 and SHA-256 in OpenSSL. TABLE III shows the computation cost for the involved cryptographic operations except the one-time user registration.

**Verification:** In our protocol, both the routers and CP need to execute the verification of users' signatures and hash chains. The difference is that the edge routers do the verification online in real time while CP can verify them off-line periodically. Besides, edge routers are also needed to conduct revocation verification before verifying signatures.

Fig. 3(a) illustrates the computation overhead of edge routers for verification. As shown in Fig. 3(a), the time cost of signature verification has no relationship with the number of

[1] https://gmplib.org/

[2] https://cropto.stanford.edu/pbc/

TABLE IV
AVERAGE TIME COST FOR SIGNATURE VERIFICATION (S)

| Number of Received Signatures | 800 | 1600 | 3200 | 6400 | 12800 |
|---|---|---|---|---|---|
| Only Probabilistic Verification | 0.86 | 1.73 | 3.46 | 6.91 | 13.82 |
| Only Batch Verification | 2.81 | 5.67 | 11.15 | 23.08 | 45.71 |
| Our Scheme | 0.28 | 0.57 | 1.12 | 2.31 | 4.57 |
| Check Probability $\beta = 0.1$ | | | | | |

revoked users. It takes 10.8 ms for verifying every signature on average. But once there exists revoked users, the total verification time cost (time cost of revocation verification and signature verification) will steeply increase by about 8 ms and it grows slowly as the number of revoked users increases (about 1 ms for every 20000 additional revoked users). Compared to the total verification overhead, the overhead of verifying the hash chains ($< 10^{-4}$ ms) is negligible.

The verification operation in edge routers is only required for the first request of every demanded file and the rest requests can be authenticated efficiently with a negligible hash operation. For example, if a user requests 100 chunks of a file and there are 102400 revoked users, the average verification time for every request is merely 0.34 ms and the user does not need to wait long for the verification.

For CP, it executes the batch verification to speed up the verification. As shown in Fig. 3(b), the time cost of the batch verification increases with the increasing number of credentials. But compared to individual verification, batch verification decreases much overhead on CP. In particular, when there are 12800 revoked users, batch verification brings 68.8% less time cost than individual verification and the advantage will increase continuously as the number of credentials rises. To further reduce the overhead in CP for verification, we let CP conduct probabilistic verification by choosing a small proportion of the received signatures based on the batch verification. Table IV represents the average time cost of only probabilistic verification, only batch verification, and our scheme with the check probability $\beta = 0.1$. Our scheme achieves high efficiency and decreases 67% and 90% overhead of only probabilistic verification and batch verification, respectively.

Besides the verification, CP also needs to open the signatures to get the real identities of the signers. Although every signature has to be opened, the opening operation is very fast (0.8 ms per signature) and can be executed in parallel.

**Broadcast Encryption:** Content encryption is comprised of a symmetric encryption (e.g., AES256) with a random key and a broadcast encryption for the random key. So every chunk cached has two parts: the ciphertext of the content and the ciphertext of the symmetric key. Due to the more expensive computation and extra ciphertext storage (120 bytes per chunk) of broadcast encryption, chunk size has a great impact on the overhead. We measure CP's computation overhead with *encryption speed* (the data size that can be encrypted in unit time) and the routers' storage overhead with *Content Payload Ratio* (the ratio of useful payload size to the full chunk size).

As shown in Fig. 3(c), both the encryption speed and content payload ratio increases with the growth of chunk size. Specifically, the encryption speed with the chunk size of 1MB (43.51 MBps) is 110 times faster than with the chunk size of 1KB (0.39 MBps). And the content payload ratio increases from 88.28% to 99.99% while the chunk size varies from 1KB to 1MB. Hence, in order to save encryption time before publishing contents and make the best use of the routers' cache space, chunk size should be large. Considering the transmission protocol and application requirements, values between 100KB and 500KB would be feasible, where the payload ratio is close to 100% and encryption speed is fast enough.

### B. Network Simulation

To illustrate user experience in ICN with our access control solution integrated, we measure the users' average content retrieval delay increased by our scheme. Specifically, one-time content retrieval delay is defined as the elapsed time between a user sending out an interest packet and receiving the corresponding data packet. The average content retrieval delay is the average value when different users request multiple contents respectively. Since NDN is a popular architecture among ICN proposals, we do the simulations with ndnSIM in NS-3. The results should be similar in other ICN architectures.

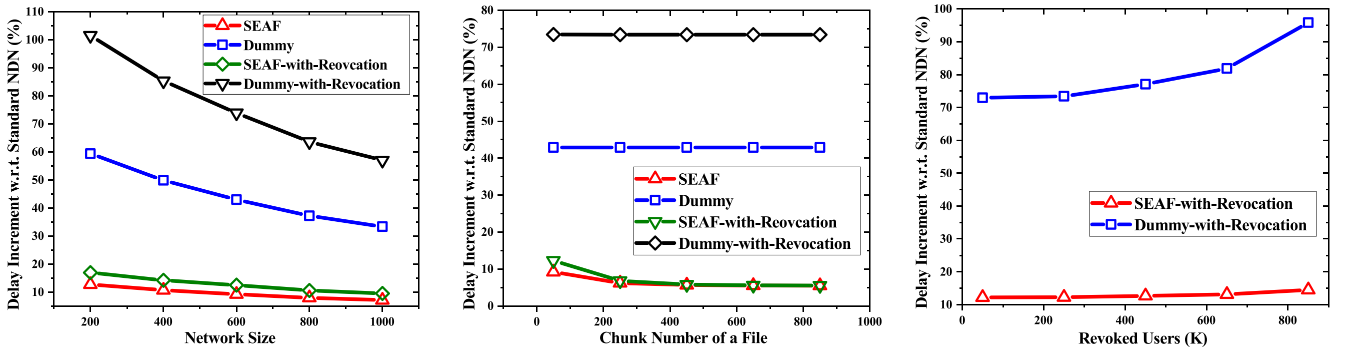
The network topologies in the simulations are generated using the two-layer top-down hierarchical model in BRTIE.[3] The autonomous system (AS) layer is generated using the Waxman model and the router layer for each AS is generated using the BarabasiAlbert model. The links between any two routers have bandwidth selected randomly from 1 to 5 Gbps and delay selected randomly from 1 to 5 ms. The number of user nodes is 20% of the routers. Each user node connects an edge router through a link with 100 Mbps bandwidth and 1 ms delay, distributed uniformly in the ASs. The cache-enabled routers are equipped with cache space for 200 chunks and LRU cache policy.

CP is located centrally in the network and able to respond to every interest packet containing its prefix. Like in the real scenarios, CP publishes new contents regularly, e.g., 10 new files per second. After publication, user nodes can request either the new contents or the old ones. Each user requests the chunks from the same file continuously and does not request the next chunk until the current request is satisfied. Also, to simulate the access control operations, user nodes and edge routers delay the corresponding time before sending or forwarding an interest packet. Based on TABLE III, by executing the precomputation, users can generate a signature in 0.03 ms instead of 8.4 ms before requesting the first chunk of a file. Additionally, the signature verification time for edge routers is 10.8 ms, the revocation verification time is related to the number of revoked users, and the decryption time for user nodes is 1.5 ms. The time for the generation and verification of hash chains is omitted because it is negligible compared to other operations.

We first test the performance of standard NDN protocol. Then we implement our proposed SEAF (denoted as *SEAF*) and a dummy protocol in which edge routers verify a signature for every interest packet (denoted as *Dummy*). Besides, we accomplish the revocation mechanism on SEAF and dummy protocol, denoting them as *SEAF-with-Revocation* and *Dummy-with-Revocation*, respectively. And under different simulation scenarios, we compare the performance of these four protocols with standard NDN protocol to get corresponding ratios.

First, we simulate all these four protocols on different sizes of network topologies, from 200 to 1000 routers. In this

[3] Boston University Representative Internet Topology Generator: https://www.cs.bu.edu/brite/



(a) (b) (c)

Fig. 4. The result of network simulation. (a) Average content retrieval delay increment vs. network size. (b) Average content retrieval delay increment vs. chunk number. (c) Average content retrieval delay increment vs. revoked user number.

simulation, the chunk size is set to 1MB, every file consists of 10 chunks, and there are 3.2K revoked users. Fig. 4(a) illustrates the average content retrieval delay introduced by all these four schemes compared with that in standard NDN (without access control). We can see that the extra delay introduced by all the protocols decreases as the network size increases due to the increasing transmission delay. And because of the overhead brought by revocation verification, both *SEAF-with-Revocation* and *Dummy-with-Revocation* need more time to retrieve the contents. But we can see that by using hash chains, *SEAF* is much better than *Dummy*, which only introduces around 10% delay. Even in the situation with revocation, the advantage of our scheme is more significant and *SEAF-with-Revocation* brings 63% less delay than *Dummy-with-Revocation*. This increment on users' content retrieval delay is insignificant for the achievement of the access control mechanism.

Then we measure the average content retrieval delay with the different chunk number of a file ranging from 10 to 1000. We do this simulation on the network topologies of 600 routers and set the chunk size as 1MB and the number of revoked users as 3.2K. As shown in Fig. 4(b), the average content retrieval delay increment of *SEAF* and *SEAF-with-Revocation* decreases with the increasing chunk number. But in *Dummy* and *Dummy-with-Revocation*, chunk number has very little influence on content retrieval performance because they verify a signature for every interest packet instead of using the hash chain to deal with the continuous requests. As the chunk number increases, the delay increment introduced by both *SEAF* and *SEAF-with-Revocation* is very small, which is about 5.5%.

Finally, we evaluate the influence of the number of revoked users on the retrieval delay. In this simulation, the network size is 600, the chunk size is set to 1MB, and every file consists of 10 chunks. Fig. 4(c) presents that the average content retrieval delay increment grows with the increasing number of revoked users and the number of revoked users has more influence on the performance of *SEAF-with-Revocation*. *SEAF-with-Revocation* and *Dummy-with-Revocation* bring the delay from 12.2% to 14.5% and 72.9% to 95.8%, respectively, when the number of revoked users varies from 0.8K to 102.4K. We can see that the use of hash chain in SEAF facilitates the performance a lot.

According to all of these simulations, we conclude that SEAF is effective and efficient enough, and using hash chain for authentication indeed helps to decrease the overhead of our scheme.

## VII. Related Work

**Access Control.** In fact, access control in distributed environments, such as sensor networks, has been studied in [29], [30]. However, the intermediate nodes in sensor networks have no possession of data owners' contents and users in ICN have stricter requirements for content delivery, thus the solutions would not work in ICN. Chen *et al.* proposed an encryption and probability access control model [11] in which authorized users obtain encryption keys of the contents from CPs, and routers pre-filter requests via a bloom filter of users' public keys to resist DoS attacks. But the scheme is impractical because of the tremendous storage overhead. Similarly, in [4], every content is related to a secret and only authorized users can obtain the secret from CP and prove it to the router. Though it is a feasible solution, the requirement of an always-online CP makes it less attractive. Fan *et al.* proposed proxy re-encryption based access control scheme [27]. This scheme is inefficient because the routers have to perform the re-encryption for every forwarding. Li *et al.* [3] proposed a capability-based security enforcement architecture that enables access control through the tokens in packets, which is similar to the use of capabilities in classical computing systems. Besides, there are other works that achieve access control by adopting advanced cryptographic algorithms such as attribute-based encryption [9] and broadcast encryption [7], to restrict users' decryption capabilities. But these schemes have no resistance to DoS attacks.

**Privilege Revocation.** The revocation problem is intractable in many situations and few literatures are focused on it. In the proxy re-encryption based access control scheme proposed by Zheng *et al.* [31], users need to ask the publisher for the decryption keys of the contents they request. Once their privileges are revoked, the publisher will refuse to return the decryption keys. Similarly, there is a access control provider in the Fotiou *et al.*'s scheme [5] and the access control provider does not allow the routers to satisfy the revoked users' requests. The revocation is enforced either in extra servers or in CP in these two schemes, which is not efficient enough. Misra *et al.* proposed an efficient revocation scheme in [7] using Shamir $(t, n)$ threshold. However, this scheme can only revoke limited users' privileges.

**Privacy Protection.** Chaabane *et al.* [32] discussed the potential privacy issues in Content-Oriented Networking and proposed the possible solutions on users' anonymity, untraceability, etc. As an effective manner, timing attacks which can infer nearby users' access history through the shorter RTT for cached contents, have drawn increasing attention as shown in [14], [15], [33]. Mohaisen *et al.* [14] solved the problem by making routers wait a random delay before sending the requested contents back to blur the response time. Acs *et al.* [33] extended the attack to local and distributed adversaries and gave complete proofs for the privacy-preserving cache mechanisms. Also, Wu *et al.* [15] proposed a networking coding based scheme that adopts random forwarding to exploit the potentials of multipath routing and improve the diversity of the anonymity set for consumers.

**Accountability.** Küsters *et al.* [34] proposed a widely applicable definition of accountability to assess the level of accountability that a protocol provides. Pappas *et al.* [35] presented a forwarding accountability mechanism that stimulates ISPs to apply stricter security polices to their customers. When it comes to ICN, accountability also includes ISPs, proving the amount of served requests to CPs and providing necessary feedback information to CPs. Ma and Towsley [16] proposed two pricing models in which CPs pay for the cache service provided by ISPs based on cache occupancy or request times. However, how to avoid the controversy between CPs and ISPs on the service is not mentioned. Ghali *et al.* [18] proposed a solution for gathering feedback information, in which routers send a notice message when cache hit occurs on routers so that CPs can collect information about the requested contents. Tourani *et al.* [17] also proposed a manifest-based approach to help CPs track their clients' behaviors and preferences more precisely. But these two schemes both rely on the routers to follow the protocol honestly.

## VIII. Conclusion

In this paper, we presented SEAF, a secure, efficient and accountable edge-based access control solution for ICN. Specifically, we showed that the access control functionality can be carried out by authenticating users' requests at the edge routers. We adopted group signature to achieve anonymous authentication and hash chain technique to reduce overhead for continuous requests. Our solution is able to (i) achieve effective access control at the network edge, (ii) provide effective and efficient revocation mechanism, (iii) preserve the data confidentiality and users' privacy from the network, (iv) allow the content providers to account the service provided by the network. Our security analysis and experimental results demonstrate that SEAF is a promising solution for the access control in ICN, which meets the security requirements and also guarantees good enough efficiency.

## References

[1] V. Jacobson *et al.*, "Networking named content," in *Proc. 5th Int. Conf. Emerging Netw. Exp. Technol.*, Dec. 2009, pp. 1–12.

[2] K. Xue *et al.*, "A withered tree comes to life again: Enabling in-network caching in the traditional IP network," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 186–193, Nov. 2017.

[3] Q. Li *et al.*, "Capability-based security enforcement in named data networking," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2719–2730, Oct. 2017.

[4] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "DACPI: A decentralized access control protocol for information centric networking," in *Proc. Int. Conf. Commun.*, May 2016, pp. 1–6.

[5] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures," in *Proc. 2nd Ed. ICN Workshop Inf.-Centric Netw.*, Sep. 2012, pp. 85–90.

[6] N. Fotiou and G. C. Polyzos, "Securing content sharing over ICN," in *Proc. 3rd ACM Conf. Inf.-Centric Netw.*, Sep. 2016, pp. 176–185.

[7] S. Misra *et al.*, "AccConF: An access control framework for leveraging in-network cached data in the ICN-enabled wireless edge," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 1, pp. 5–17, Jan./Feb. 2017. doi: 10.1109/TDSC.2017.2672991.

[8] Q. Li, X. Zhang, Q. Zheng, R. Sandhu, and X. Fu, "LIVE: Lightweight integrity verification and content access control for named data networking," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 308–320, Feb. 2015.

[9] B. Li, D. Huang, Z. Wang, and Y. Zhu, "Attribute-based access control for ICN naming scheme," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 2, pp. 194–206, Mar./Apr. 2018.

[10] M. Mangili, F. Martignon, and S. Paraboschi, "A cache-aware mechanism to enforce confidentiality, trackability and access policy evolution in content-centric networks," *Comput. Netw.*, vol. 76, pp. 126–145, Jan. 2015.

[11] T. Chen, K. Lei, and K. Xu, "An encryption and probability based access control model for named data networking," in *Proc. IEEE Int. Perform. Comput. Commun. Conf.*, Dec. 2014, pp. 1–8.

[12] Y. Xin, Y. Li, W. Wang, W. Li, and X. Chen, "A novel interest flooding attacks detection and countermeasure scheme in NDN," in *Proc. IEEE Global Commun. Conf.*, Dec. 2016, pp. 1–7.

[13] Q. Li, R. Sandhu, X. Zhang, and M. Xu, "Mandatory content access control for privacy protection in information centric networks," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 5, pp. 494–506, Sep./Oct. 2017.

[14] A. Mohaisen, X. Zhang, M. Schuchard, H. Xie, and Y. Kim, "Protecting access privacy of cached contents in information centric networks," in *Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secur.*, May 2013, pp. 173–178.

[15] Q. Wu *et al.*, "Privacy-aware multipath video caching for content-centric networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2219–2230, Aug. 2016.

[16] R. T. Ma and D. Towsley, "Cashing in on caching: On-demand contract design with linear pricing," in *Proc. 11th ACM Conf. Emerg. Netw. Exp. Technol.*, Dec. 2015, p. 8.

[17] R. Tourani, S. Misra, and T. Mick, "Application-specific secure gathering of consumer preferences and feedback in ICNs," in *Proc. 3rd ACM Conf. Inf. Centric Netw.*, Sep. 2016, pp. 65–70.

[18] C. Ghali, G. Tsudik, C. A. Wood, and E. Yeh, "Practical accounting in content-centric networking," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2016, pp. 436–444.

[19] K. Xue *et al.*, "SEAF: A secure, efficient and accountable access control framework for information centric networking," in *Proc. Int. Conf. Comput. Commun.*, Apr. 2018, pp. 2213–2221.

[20] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 3152, M. K. Franklin, Ed. Berlin, Germany: Springer, 2004, pp. 41–55.

[21] C. Delerablëe, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys," in *Proc. 1st Int. Conf. Pairing-Based Cryptogr.*, 2007, pp. 39–59.

[22] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981.

[23] Z. Zhu and R. Jiang, "A secure anti-collusion data sharing scheme for dynamic groups in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 40–50, Jan. 2016.

[24] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: Secure multi-owner data sharing for dynamic groups in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1182–1191, Jun. 2013.

[25] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* New York, NY, USA: Springer, Nov. 2001, pp. 514–532.

[26] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, 2000.

[27] C.-I. Fan, I.-T. Chen, C.-K. Cheng, J.-J. Huang, and W.-T. Chen, "FTP-NDN: File transfer protocol based on re-encryption for named data network supporting nondesignated receivers," *IEEE Syst. J.*, vol. 12, no. 1, pp. 473–484, Mar. 2018.

[28] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2: An updated NDN simulator for NS-3," Univ. California, Los Angeles, Los Angeles, CA, USA, Tech. Rep. NDN-0028, Nov. 2016.

[29] R. Zhang, Y. Zhang, and K. Ren, "DP$^2$AC: Distributed privacy-preserving access control in sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, Aug. 2009, pp. 1251–1259.

[30] D. He *et al.*, "Distributed privacy-preserving access control in a single-owner multi-user sensor network," in *Proc. IEEE Int. Conf. Comput. Commun.*, Apr. 2011, pp. 331–335.

[31] Q. Zheng, G. Wang, R. Ravindran, and A. Azgin, "Achieving secure and scalable data access control in information-centric networking," in *Proc. IEEE Int. Conf. Commun.*, Jul. 2015, pp. 5367–5373.

[32] A. Chaabane, E. De Cristofaro, M. A. Kaafar, and E. Uzun, "Privacy in content-oriented networking: Threats and countermeasures," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 25–33, 2013.

[33] G. Acs *et al.*, "Privacy-aware caching in information-centric networking," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 2, pp. 313–328, Mar. 2019.

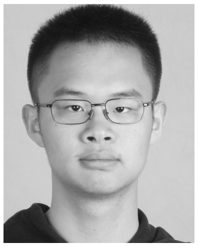
[34] R. Küsters, T. Truderung, and A. Vogt, "Accountability: Definition and relationship to verifiability," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, Jul. 2010, pp. 526–535.

[35] C. Pappas, R. M. Reischuk, and A. Perrig, "FAIR: Forwarding accountability for internet reputability," in *Proc. 23rd IEEE Int. Conf. Netw. Protocols*, Jul. 2015, pp. 189–200.

**Kaiping Xue** (M'09–SM'15) received the B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003, and the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. From 2012 to 2013, he was a Post-Doctoral Researcher with the Department of Electrical and Computer Engineering, University of Florida. He is currently an Associate Professor with the Department of Information Security and the Department of EEIS, USTC. His research interests include next-generation Internet, distributed networks, and network security.

**Peixuan He** received the B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2017. He is currently pursuing the master's degree in information security from the Department of Electronic Engineering and Information Science (EEIS), USTC. His research interests include network security protocol design and analysis.

**Xiang Zhang** received the B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2015, and the master's degree in communication and information system from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2018. His research interests include ICN architecture design, access control in ICN, and smart grid security.



**Qiudong Xia** received the B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2018. He is currently pursuing the master's degree in information security from the Department of Electronic Engineering and Information Science (EEIS), USTC. His research interests include architecture design and security protection in ICN.

**David S. L. Wei** (SM'07) received the Ph.D. degree in computer and information science from the University of Pennsylvania in 1991. From 1993 to 1997, he was with the Faculty of Computer Science and Engineering, University of Aizu, Japan (as an Associate Professor and then a Professor). He has authored or coauthored over 100 technical papers in various archival journals and conference proceedings. He is currently a Professor with the Computer and Information Science Department, Fordham University. His research interests include cloud computing, big data, IoT, and cognitive radio networks. He was a Guest Editor or a lead Guest Editor of several special issues in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, and the IEEE TRANSACTIONS ON BIG DATA. He also served as an Associate Editor for the IEEE TRANSACTIONS ON CLOUD COMPUTING from 2014 to 2018 and an Associate Editor of the *Journal of Circuits, Systems and Computers* from 2013 to 2018.

**Hao Yue** received the B.Eng. degree in telecommunication engineering from Xidian University, Xi'an, China, in 2009, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2015. He is currently an Assistant Processor with the Department of Computer Science, San Francisco State University, San Francisco, CA, USA. His research interests include cyber-physical systems, cybersecurity, wireless networking, and mobile computing.

**Feng Wu** (M'99–SM'06–F'13) received the B.S. degree in electrical engineering from Xidian University in 1992, and the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology in 1996 and 1999, respectively. He is currently a Professor with the University of Science and Technology of China (USTC). Before that, he was a Principle Researcher and the Research Manager of Microsoft Research Asia. He has authored or coauthored over 200 high-quality papers. His research interests include computational photography, image and video compression, media communication, media analysis and synthesis, multimedia communications, image and video processing, and artificial intelligence. As a coauthor, he has received the Best Paper Award from SPIE VCIP 2007, PCM 2008, and the IEEE TCSVT 2009. He has received the IEEE Circuits and Systems Society 2012 Best Associate Editor Award. He has also served as the TPC Chair for PCM 2009, VCIP 2010, and MMSP 2011, and the Special Sessions Chair for ICME 2010 and ISCAS 2013. He serves as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEM FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, and several other international journals.