SEAF: A Secure, Efficient and Accountable Access Control Framework for Information Centric Networking

Kaiping Xue[1,*], Xiang Zhang[1], Qiudong Xia[1], David S.L. Wei[2], Hao Yue[3], Feng Wu[1]

[1] Department of EEIS, University of Science and Technology of China, Hefei, Anhui 230027, China
[2] Department of Computer and Information Science, Fordham University, New York, NY 10458, USA
[3] Department of Computer Science, San Francisco State University, San Francisco, CA 94132, USA
*kpxue@ustc.edu.cn

***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 delivery service, an effective access control mechanism is needed due to wide dissemination of contents. However, in the existing solutions, making cache-enabled routers or content providers authenticate users' requests causes high computation overhead and unnecessary delay. Also, straightforward utilization of advanced encryption algorithms increases the opportunities for DoS attacks. Besides, privacy protection and service accountability are rarely taken into account in this scenario. In this paper, we propose a secure, efficient, and accountable access control framework, called SEAF, 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 greatly reduce the overhead when users make continuous requests for the same file. 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 introduces only slight delay for users' content retrieval.**

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) paradigm has been proposed as a promising alternative [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 a 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 requests can be satisfied by the routers in the forwarding path while 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. Such scenario poses serious threat to CPs' interests. Therefore, an effective access control mechanism is 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, which brings heavy computation overhead and degrades the forwarding performance [3], [6], or requires interactions with CP [4] (or an access control server [5]) during the content retrieval, which causes significant delay for users and offsets the benefits provided by in-network cache. 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 be avoided because 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 that the confidentiality of the contents is preserved. Unfortunately, though the unauthorized

978-1-5386-4128-6/18/$31.00 ©2018 IEEE

users cannot decrypt the contents, they still can retrieve the encrypted contents from the network because the routers do not discriminate the requests. As a result, the network can be easily exhausted by the flooding of the requests [11], [12].

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 from 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, CPs cannot 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 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 service [17], [18].

Motivated by the above observations, we present a secure, efficient, and accountable access control framework, called SEAF, in this paper. In SEAF, to separate access control from content provision, we let routers at the network edge authenticate users' requests so that the bandwidth and cache resources inside the network are only accessible to authorized users. For privacy protection, users authenticate themselves by generating a valid group signature to keep them anonymous to the edge routers. Nevertheless, signature generation and verification require expensive computation. Thus, a trivial solution that uses signature on every request is impratical. 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 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. This 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 the 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 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.

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

II. Problem Statement

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.

CPs produce contents and the contents are disseminated into ISP network because of its cache functionality. Users obtain desired contents from CPs or ISP network. ISP network includes two types of routers: edge routers and cache-enabled routers. Specifically, the edge routers, though have no caching capability, authenticate users' requests before forwarding them into the network. The cache-enabled routers forward the requests and responds to them if the requested contents exist in their cache.

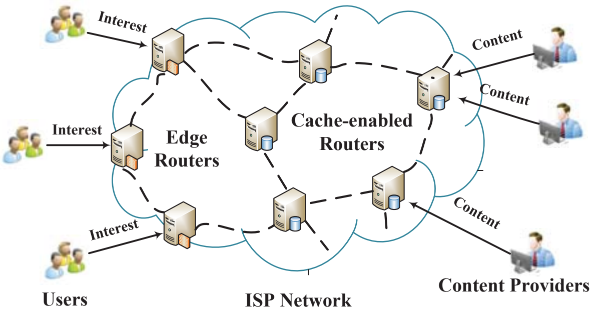

Fig. 1. System Model

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

B. Trust Assumptions and Threat Model

ISP provides cached contents to users and charge 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 CPs to purchase its service. By *curious*, we mean that ISP is curious about the rich information of the cached contents, and the 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.

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' served requests. Users are assumed to be malicious. 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 the said attacks or deceive CPs.

C. Preliminaries

1) 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 [19] and broadcast encryption [20] which work on the bilinear pairings with q-Strong Diffie-Hellman (SDH) and Weak Bilinear Diffie-Hellman Exponent (WBDHE) assumption as follows:

Definition 1: q-SDH Assumption: Given a $(q + 2)$-tuple $(g_1, g_2, g_2^x, g_2^{x^2}, \dots, g_2^{x^q})$ 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 ϵ to solve q-SDH if

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

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

2) 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 $h()$ 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 [21]. 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 simply authenticate himself 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.

III. 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 users. Without loss of generality, CPs use numbers as group identifers and a larger group identifer means a higher access level. This manner of privilege management is efficient and has been widely applied.

To enable access control to be enforced outside CPs, we introduce group signature and broadcast encryption. Specifically, group signature allows edge routers to authenticate users without revealing their real identities and broadcast encryption guarantees only the users with corresponding access privileges can decrypt the contents. 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 and service accounting.

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 its access policy, CP initializes the system as follows:

- Generate 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 $q = e(g_1, g_2)$;
- Select a random element $h \in G_1$ and two random numbers $\xi_1, \xi_2 \in \mathbb{Z}_p^*$, and let $u, v \in G_1$ such that $u^{\xi_1} = v^{\xi_2} = h$;
- Select m random numbers $\gamma_1, \gamma_2, \dots, \gamma_m \in \mathbb{Z}_p^*$, and let $w_i = {g_2}^{\gamma_i}$ where $i = 1, 2, \dots, m$. Denote Γ as $(\gamma_1, \gamma_2, \dots, \gamma_m)$ and W as $(w_1, w_2, \dots, w_m)$;
- Select m random numbers $\lambda_1, \lambda_2, \dots, \lambda_m \in \mathbb{Z}_p^*$, and let $y_i = {g_1}^{\lambda_i}$ where $i = 1, 2, \dots, m$. Denote Λ as $(\lambda_1, \lambda_2, \dots, \lambda_m)$ and Y as $(y_1, y_2, \dots, y_m)$;
- Publish the public parameters:

  $$(S, g_2, h, u, v, z, W, Y, H_1, H_2, E(\cdot)),$$

  where H_1 is a one-way hash function: $(0, 1)^* \rightarrow \mathbb{Z}_p^*$, 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 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

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

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

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

for $k = 1, 2, ..., n$. Denote $Z_j = (z_j^1, z_j^2, ..., z_j^n)$, $B_j = (b_j^1, b_j^2, ..., b_j^n)$ and $D_j = (d_j^1, d_j^2, ...d_j^n)$. 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 level.

D. Content Generation

To make the access 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 it belongs to and *chunk_1* is used to specify the content in this file. If the content has an access level of 3 then the content name is changed to */com/example/**3**/subdir/abc.mp4/chunk_1*. The inserted access 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. CP first randomly selects $k \in \mathbb{Z}_p^*$, computes $K = q^k$, and encrypts the content M with K: $C = E_K(M)$. Suppose that the users in group $n, n+1, \ldots, m$ can access the content, CP selects y_n and encrypts the symmetric key K by computing

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

Then, the content M is stored as (C, C_1, C_2) in CP and cache-enabled routers.

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 $\sigma = (T_1, T_2, T_3, R_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$
```

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 reveal 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 right hash chain elements are also regarded as authorized. The details are as follows.

When user j requests the first chunk of some file, he/she generates a hash chain with 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 σ. Note that the filename f is a prefix of the chunk name. The signature σ is generated by the algorithm shown in **Algorithm 1**.

After receiving the Interest packet, the edge router first checks the validity of the timestamp *TS* and then verifies the signature σ 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 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.

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, she/he only needs to attach the new element H_{new} of the hash chain in the Interest packet. The edge router extracts the filename f' from the content name and computes $H_2(H_{new})$. Then the edge router searches the 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 counter field.

When a user retrieves all the chunks of the file or loses interest halfway, the user 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 σ 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:

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

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

Algorithm 2 Signature Verification

```
Input: System parameters $(g_2, h, u, v, w_n)$, signed info
    $f, \mathit{TS}, H_{tail}$ and a signature $\sigma$.
Output: Valid or Invalid.
Set $M' = f \parallel \mathit{TS} \parallel H_{tail}$
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
else
    return Invalid
end if
```

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

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

As described above, 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, σ is the user's signature on $f \parallel \mathit{TS} \parallel H_{tail}$, 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 divides the credentials into m subsets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m$ according to group identifier n. For each subset, CP verifies the group signatures using **Algorithm 3**. To reduce the computation overhead and accelerate the process, CP can select a random subset of the credentials in a certain proportion to verify. As long as the credentials in the smaller subset can make a successful verification, CP can trust that all the credentials are legitimate.

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.

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

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

with ξ_1 and ξ_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 $< f, \mathit{TS}, ID_j, l >$, 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 content service.

Algorithm 3 Batch Verification

```
Input: System parameters $(g_2, h, u, v, w_n)$ and $\|\mathcal{S}_n\|$ service
    credentials $< f_i, \mathit{TS}_i, H_{i,tail}, n, \sigma_i, H_{i,last}, l_i >$ 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 $M'_i = f_i \parallel \mathit{TS}_i \parallel H_{i,tail}$,
        $R_{i,1} = u^{s_{i,\alpha}} T_1^{-c_i},\ R_{i,2} = v^{s_{i,\beta}} T_2^{-c_i}$,
        $R_{i,4} = T_{i,1}^{s_{i,x}} u^{-s_{i,\delta_1}},\ R_{i,5} = T_{i,2}^{s_{i,x}}\ v^{-s_{i,\delta_2}}$,
        $c'_i = H_1(M'_i, T_{i,1}, T_{i,2}, T_{i,3}, R_{i,1}, R_{i,2}, R_{i,3}, R_{i,4}, R_{i,5})$
    if $c_i \neq c'_i$ then
        return Invalid
    end if
    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 for
$Q' = e(P_1, g_2) e(h, w_n)^{P_2} e(h, g_2)^{P_3} e(P_4, w_n) q^{P_5}$
if $Q' \neq Q$ then
    return Invalid
end if
return Valid
```

IV. Security Analysis

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

A. Data Confidentiality

Our proposed scheme protects data confidentiality from both malicious routers and malicious 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, or there exists colluding with malicious users.

Lemma 4.1: Unauthorized entities, including routers and malicious users cannot learn any information from the encrypted contents, individually or in collusion.

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 an adversary 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 λ_n, it can compute

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

This obviously contradicts with WBDHE assumption. Thus the correctness of Lemma 4.1 can be ensured. ■

B. Unforgeability

To access the network, a user needs to compute a group signature on the request information. The demonstration of

TABLE I
COMPARISON WITH OTHER ACCESS CONTROL SCHEMES

Scheme	Data Confidentiality	DoS Resistance	Offline CP	Privacy Protection	Accountability
DACPI [4]	No	Yes	No	No	Yes
AccConf [7]	Yes	No	Yes	Yes	No
FTP-NDN [22]	Yes	Yes	Yes	No	No
SEAF	Yes	Yes	Yes	Yes	Yes

this property can be derived from the following two lemmas:

Lemma 4.2: An unauthorized user is unable to forge the authentication message to access the network based on q-SDH problem assumption.

Proof: Suppose that an adversary $\mathcal{A}$ succeeds to forge a valid group signature with a non-negligible probability in polynomial time, and we assume H_1 is a random oracle. Then the 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 with q-SDH assumption. ■

Lemma 4.3: Any adversary cannot 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 σ as well. This is equivalent to forging a valid signature, which has been proved infeasible in Lemma 4.2. ■

C. Anonymity

Inheriting from the group signature, the authentication messages will not leak any information about the signer's identity. Actually, a group signature can be regarded as a non-interactive zero-knowledge proof for SDH problem. The completeness and soundness guarantee that the network can be convinced that the user belongs to his/her asserted group. But it will not reveal anything about the user's secret key.

Additionally, unlinkability is also preserved. It means that given two group signatures, σ_1 and σ_2, a verifier cannot determine whether they are signed by the same user. We can revisit **Algorithm 1** and notice that in a signature σ, T_3 and s_x are the only elements embedded with user's secret key. But these elements are blinded by two independent random secrets. Thus, T_3 and s_x from two signatures generated with the same secret key, can also be simulated by two different keys. Therefore, unlinkability is achieved.

D. Traceability

Given a valid group signature σ and the private parameters ξ_1, ξ_2, CP can compute the private key of the signer through $A_j = T_3/(T_1^{\xi_1} \cdot T_2^{\xi_2})$, the equation of Eq. (5). The correctness holds based on the following relation: $T_3 = A_j \cdot h^{\alpha+\beta} = A_j \cdot u^{\xi_1 \cdot \alpha} \cdot v^{\xi_2 \cdot \beta} = A_j \cdot T_1^{\xi_1} \cdot T_2^{\xi_2}$. The traceability guarantees that users' identities can be revealed for CP's accounting operation afterwards.

E. 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.

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. Also, 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.

F. Comparison

We compare the proposed SEAF with several other representative access control schemes in terms of the aspects of data confidentiality, DoS resistance, offline CP, privacy protection, and accountability. As summarized in TABLE I, only SEAF can achieve all of the features.

V. 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 and group signature. Then we evaluate the computation and storage overhead for CP and routers crosswise. Finally, we use NS-3 and ndnSIM [23] to simulate our protocol integrated in standard NDN and show that SEAF only introduces slight content retrieval delay. 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 II shows the computation cost for the involved cryptographic operations except the one-time user registration.

TABLE II
Computation cost for cryptographic operations

Category	Operation	Time (ms)
Signature-related	Generation (Without Precomputation)	5.1
	Generation (With Precomputation)	0.03
	Verification	10.3
	Batch Verification	8.1
	Opening	0.8
	SHA-256	$< 10^{-4}$
Encryption-related	Broadcast Encryption	2.5
	Broadcast Decryption	1.5
	AES-256	0.02 (1K)

Signature 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 offline periodically. It takes 10.3 ms to verify a single signature, compared to which, the overhead of verifying the hash chains ($< 10^{-4}$ ms) is negligible.

For edge routers, the verification operation 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, the average verification time for every request is merely 0.13 ms and the user does not need to wait long for the verification.

For CP, it can choose a small proportion of the received signatures and execute the batch verification to speed up the verification simultaneously. 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 (160 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. 2, both the encryption speed and content payload ratio increase with respect to the growth of chunk size. Specifically, the encryption speed of the chunk size of 1MB (43.29 MBps) is 110 times faster than that of the chunk size of 1KB (0.38 MBps). Also, the content payload ratio increases from 86.48% to 99.98% 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.

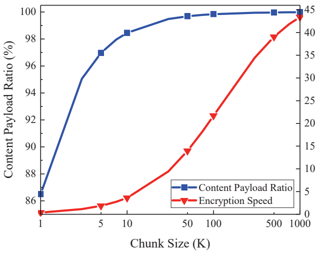

Fig. 2. Encryption speed and content payload ratio vs. chunk size

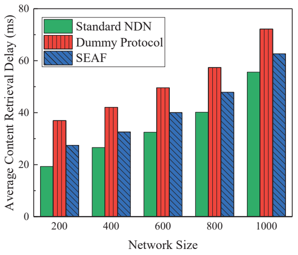

Fig. 3. Average content retrieval time vs. network size

B. Network Simulation

To illustrate user experience in ICN with our access control solution integrated, we measure the users' average content retrieval delay. 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 ICN proposal among ICN architectures, we do the simulations with ndnSIM in NS-3. The results should be similar in other ICN architectures.

We simulate the standard NDN protocol, our proposed SEAF and a dummy protocol in which edge routers verify a signature for every Interest packet on different sizes of network topologies, from 200 to 1000 routers. The topologies are generated using the two-layer top-down hierarchical model

[1] https://gmplib.org/, accessed on Dec. 31st, 2017.

[2] https://cropto.stanford.edu/pbc/, accessed on Dec. 31st, 2017.

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 the bandwidth selected randomly from 1 to 5 Gbps and the 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.

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. Different from the standard NDN protocol in which every router is cache-enabled, in our protocol edge routers do not cache contents. 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 II, by executing the precomputation of random numbers, users can generate a signature in 0.03 ms instead of 5.1 ms before requesting the first chunk of a file. Additionally, the signature verification time for edge routers is 10.3 ms 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.

According to the aforementioned discussion, the chunk size is set to 100K, and every file consists of 10 chunks. The cache-enabled routers are equipped with cache space for 200 chunks and LRU cache policy. We run every simulation for 1000 seconds. Fig. 3 presents the average content retrieval delay in standard NDN (without access control), dummy access control protocol, and the proposed SEAF. The results are simulated in different sizes of network. Compared to standard NDN, the dummy protocol which does signature verification for every request increases about 20 ms delay. However, in SEAF, due to the use of hash chains, only about 8 ms delay is introduced, which can be further reduced for the files with more chunks. Such increase on users' content retrieval delay is insignificant for the achievement of the access control mechanism. We are thus confident that SEAF is effective and efficient enough.

VI. Related Work

Access Control. In fact, access control in distributed environments, such as sensor networks, has been studied in [24], [25]. However, since the intermediate nodes in sensor networks have no possession of data owners' contents and users in ICN have stricter requirements for content delivery, 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 [22]. However, 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.

Privacy Protection. Chaabane *et al.* [26] discussed the potential privacy issues in Content-Oriented Networking and proposed the possible solutions on users' anonymity, untraceability, and so on. As an effective manner, timing attacks which can infer nearby users' access history through the shorter RTT for cached contents, have drawn increasing attention in [14], [15], [27]. Mohaisen *et al.* [14] solved the problem by making routers wait for a random delay before sending the requested contents back to blur the response time. Acs *et al.* [27] 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.* [28] proposed a widely applicable definition of accountability that enables assess to the level of accountability that a protocol provides. Pappas *et al.* [29] 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 request to CPs and providing necessary feedback information to CPs. Ma *et al.* [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.

VII. Conclusion

In this paper, we presented a secure, efficient, and accountable access control framework, called SEAF for Information Centric Networking. Specifically, we showed that the access

[3] Boston University Representative Internet Topology Generator: https://www.cs.bu.edu/brite/, accessed on Dec. 31st, 2017.

control functionality can be carried out by authenticating users' requests at the edge routers. We adopt 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) preserve the data confidentiality and protect users' privacy from the network, (iii) 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 various security requirements and also guarantees good enough efficiency.

ACKNOWLEDGEMENT

The authors sincerely thank the anonymous referees for their invaluable suggestions that have led to the present improved version of the original manuscript. This work is supported in part by the National Key R&D Program of China under Grant No. 2017YFB0801702, the National Natural Science Foundation of China under Grant No. 61379129 and No. 61631017, the Key Research Program of of the Chinese Academy of Sciences (CAS) under Grant No. ZDRW-KT-2016-2-5, Youth Innovation Promotion Association of CAS under Grant No. 2016394, and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT 2009)*. ACM, 2009, pp. 1–12.

[2] K. Xue, T. Hu, X. Zhang, P. Hong, D. S. Wei, and F. Wu, "A withered tree comes to life again: Enabling in-network caching in the traditional IP network," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 186–193, 2017.

[3] Q. Li, P. P. Lee, P. Zhang, P. Su, L. He, and K. Ren, "Capability-based security enforcement in named data networking," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2719–2730, 2017.

[4] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "DACPI: A decentralized access control protocol for information centric networking," in *Proceedings of 2016 International Conference on Communications (ICC 2016)*. IEEE, 2016, pp. 1–6.

[5] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures," in *Proceedings of the 2nd Edition of the ICN Workshop on Information-Centric Networking*. ACM, 2012, pp. 85–90.

[6] N. Fotiou and G. C. Polyzos, "Securing content sharing over ICN," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking (ICN 2016)*. ACM, 2016, pp. 176–185.

[7] S. Misra, R. Tourani, F. Natividad, T. Mick, N. E. Majd, and H. Huang, "AccConF: An access control framework for leveraging in-network cached data in the ICN-enabled wireless edge," *IEEE Transactions on Dependable and Secure Computing, Avaliable online*, 2017, https://doi.org/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 Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 308–320, 2015.

[9] B. Li, D. Huang, Z. Wang, and Y. Zhu, "Attribute-based access control for ICN naming scheme," *IEEE Transactions on Dependable and Secure Computing, Avaliable online*, 2016, https://doi.org/10.1109/TDSC.2016.2550437.

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

[11] T. Chen, K. Lei, and K. Xu, "An encryption and probability based access control model for named data networking," in *Proceedings of 2014 IEEE International Performance Computing and Communications Conference (IPCCC 2014)*. IEEE, 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 *Proceedings of 2016 IEEE Global Communications Conference (Globecom 2016)*. IEEE, 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 Transactions on Dependable and Secure Computing*, vol. 14, no. 5, pp. 494–506, 2017.

[14] A. Mohaisen, X. Zhang, M. Schuchard, H. Xie, and Y. Kim, "Protecting access privacy of cached contents in information centric networks," in *Proceedings of the 8th ACM SIGSAC symposium on Information, Computer and Communications Security (ASIACCS 2013)*. ACM, 2013, pp. 173–178.

[15] Q. Wu, Z. Li, G. Tyson, S. Uhlig, M. A. Kaafar, and G. Xie, "Privacy-aware multipath video caching for content-centric networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2219–2230, 2016.

[16] R. T. Ma and D. Towsley, "Cashing in on caching: On-demand contract design with linear pricing," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT 2015)*. ACM, 2015, p. 8.

[17] R. Tourani, S. Misra, and T. Mick, "Application-specific secure gathering of consumer preferences and feedback in ICNs," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking (ICN 2016)*, 2016, pp. 65–70.

[18] C. Ghali, G. Tsudik, C. A. Wood, and E. Yeh, "Practical accounting in content-centric networking," in *Proceedings of 2016 IEEE/IFIP Network Operations and Management Symposium (NOMS 2016)*. IEEE, 2016, pp. 436–444.

[19] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proceedings of the 24th Annual International Cryptology Conference (Advances in Cryptology - CRYPTO 2004)*, vol. 3152. Springer, 2004, pp. 41–55.

[20] C. Delerablée, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys," *Proceedings of the 1st international conference on Pairing-Based Cryptography (Pairing 2007)*, pp. 39–59, 2007.

[21] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[22] 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 Systems Journal, Avaliable online*, 2016, https://doi.org/10.1109/JSYST.2016.2580299.

[23] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2: An updated NDN simulator for NS-3," NDN, Technical Report NDN-0028, Revision 2, November 2016.

[24] R. Zhang, Y. Zhang, and K. Ren, "DP^{2}AC: Distributed privacy-preserving access control in sensor networks," in *Proceedings of 2009 IEEE International Conference on Computer Communications (INFOCOM 2009)*. IEEE, 2009, pp. 1251–1259.

[25] D. He, J. Bu, S. Zhu, M. Yin, Y. Gao, H. Wang, S. Chan, and C. Chen, "Distributed privacy-preserving access control in a single-owner multi-user sensor network," in *Proceedings of 2011 IEEE International Conference on Computer Communications (INFOCOM 2011)*. IEEE, 2011, pp. 331–335.

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

[27] G. Acs, M. Conti, P. Gasti, C. Ghali, G. Tsudik, and C. Wood, "Privacy-aware caching in information-centric networking," *IEEE Transactions on Dependable and Secure Computing*, 2017.

[28] R. Küsters, T. Truderung, and A. Vogt, "Accountability: definition and relationship to verifiability," in *Proceedings of the 17th ACM conference on Computer and Communications Security (CCS 2010)*. ACM, 2010, pp. 526–535.

[29] C. Pappas, R. M. Reischuk, and A. Perrig, "FAIR: forwarding accountability for internet reputability," in *Proceedings of the 23rd IEEE International Conference on Network Protocols (ICNP 2015)*. IEEE, 2015, pp. 189–200.