TSLS: Time Sensitive, Lightweight and Secure Access Control for Information Centric Networking

Qiudong Xia*, Peixuan He*, Kaiping Xue*§, Jiangping Han*, David S.L. Wei†, Hao Yue‡, Jin Qin*
* Department of EEIS, University of Science and Technology of China, Hefei, Anhui 230027 China
† Department of Computer and Information Science, Fordham University, Bronx, NY 10458, USA
‡ Department of Computer Science, San Francisco State University, San Francisco, CA 94132, USA
§Corresponding author, kpxue@ustc.edu.cn

Abstract—**Information Centric Networking (ICN), a new paradigm of Internet infrastructure, aims to better accommodate users' rapid growing demand for content delivery and optimize bandwidth utilization. Although the in-network cache feature of ICN facilitates the dissemination of content to users, it also poses new challenges on access control for content and network resource. Moreover, it is common that the access privilege of content dynamically change over time. However, existing access control mechanisms in ICN cannot support the publication and distribution of such time-sensitive content. In this paper, we propose a time-sensitive, lightweight, and secure access control mechanism, called TSLS, to solve this problem. We introduce broadcast encryption combined with time tokens for content providers to protect content confidentiality, and only authorized users satisfying the time limitation have capability to decrypt and access the content. Besides, a fast lightweight challenge-response verification is implemented at the edge routers to block unauthorized request from injecting into the network. The responses of authorized users are forwarded to content providers for pre-distribute popular content at in-network caches in advance. Our security analysis shows that TSLS possesses the properties of data confidentiality, unforgeability, anonymity, and DoS/DDoS attacks resistance. Our simulation results indicate that our proposed TSLS is an efficient mechanism with low computation cost and network delay.**

I. Introduction

Information Centric Networking (ICN) is an emerging architecture designed for the next generation Internet, which aims to handle the growing users' demand for the content delivery and has drawn significant attentions in recent years [1]. The new features of ICN, such as in-network cache and remarkable mobility support, benefit the core network and users but also introduce new security challenges, one of which is access control. Due to the fact that contents are cached in the whole network and requests from users can be satisfied by the routers without the permission of content providers (CP), CPs will lose control of their contents in ICN. Therefore, how to achieve an efficient access control mechanism for CPs' contents becomes an important and challenging problem in ICN.

Access control in ICN have been studied in the literature. Some of existing solutions are based on the cryptology mechanisms such as [2]–[4], where only authorized users have the ability to decrypt contents. However, since the service of ICN is indiscriminate, routers will respond to the requests from all users, no matter whether a user is benign or malicious. Therefore, malicious users can exhaust network resourses with excessive requests easily. The others use authentication-based mechanisms [5]–[7], routers need to authenticate received requests before sending data content back to the users. However, these schemes incur heavy computational overhead on routers and increase the response delay of each request. In addition, these solutions cannot satisfy functional requirements for access control in reality, such as access privilege time-releasing. In practice, the access privilege of contents usually needs to be expanded for different users at different time gradually. For example, only very important people (VIP) can obtain a video at the early time when it is just published. Other users can access the video after a pre-determined time period when the access privilege of the video is extended. It is critical to enable a time-sensitive access control for content distribution in ICN.

Moreover, although requests from legitimate users may be refused by the time-sensitive access control mechanism because CPs do not release the content access privilege to intended users until reaching pre-defined timepoints, these requests are still meaningful to CPs for accurately forecasting users' demands for specific contents. CPs can rent some in-network caches and store contents at these caches in advance according to users' demand. When contents are published at the early time, all the requests are satisfied and responded by CPs for the lack of content duplication in caches. So pre-distributed contents can largely reduce the pressure of ICN and CPs and improve users' experience.

Motivated by these observations, in this paper, we propose a time-sensitive, lightweight and secure access control mechanism, named TSLS, for ICN. In TSLS, access control is conducted at the edge of a network where users' requests are launched. Using the broadcast encryption combined with time token mechanism, CPs ensure content confidentiality and can pre-distribute potential popular contents to core router nodes. Before reaching the first releasing time of a certain content, the requests from legitimate users can be pre-recorded and direct the content distribution by implementing a lightweight challenge-response based access privilege verification at the edge routers. When a releasing time has passed, edge routers update the time tokens and users with the allowed access privilege levels can obtain the content quickly. Through releasing time tokens at different time points, the content's access privilege is expanded by CPs and more users can access the content. Our contributions can be summarized as follows:

- We develop an efficient lightweight access control mechanism for ICN, where the edge routers prevent the reqeusts from unauthorized users into the core network. The challenge-response based verification is lightweight with low computational cost and delay.
- We implement a time-sensitive access control mechaism by integrating the time token mechanism into broadcast encryption. The time access policy of contents can be customized by CPs and only users who satisfy the access privilege of contents can obtain contents.
- We reuse the abandoned requests from legitimate users so that CPs can receive the demands from users. Furthermore, CPs set contents near users according to users' demands for efficient contents distribution.

The rest of this paper is organized as follows. The related work is introduced in Section II. In Section III, we describe the system model, threat assumption, and the preliminaries. Our proposal details are presented in Section IV. We then analyze the security and performance in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

Besides some emerging security issues in ICN, such as content poison [8] and privacy preserving [9], it is important to implement efficient access control in ICN. Some schemes have been proposed to address this challenge in recent years. Based on cryptographic technologies, such as proxy re-encryption, broadcast encryption, and attribute-based encryption, researchers have proposed cryptography-based access control mechanisms for ICN. Attribute-based encryption has been used in many access control schemes, such as [10], [11], for its ability to achieve fine-grained access control, and the work in [2] shows that it can also be well applied in ICN. Meanwhile, broadcast encryption is introduced in [3] to flexibly revocate users' privileges, and Zheng *et al.* [4] introduced proxy re-encryption to develop a solution for ICN by re-encrypting the content at edge routers before forwarding it to the users. These schemes have common characteristics where access control is achieved via users' decryption ability. However, they haven't addressed how to prevent potential DoS/DDoS attacks in ICN, as without verification of access capability, interest packets will be straightly forwarded into the core network and corresponding contents will be returned to illegitimate users, which will largely waste network resources, including computing, caching and bandwidth.

Some other schemes verify users' requests first and the illegitimate requests will be dropped. Abdallah *et al.* [5] proposed a mutual authentication protocol between users and routers before providing content. However, in this scheme, requests must be verified by the CP, which loses the advantage of in-network cache. Li *et al.* [6] proposed a capability-based security enforcement architecture, in which a token is generated by CP for each reqeust to represent users' privilege and the token is verified by the routers. These access control mechanisms require every router in the network to participate in the service, which increases the computational workload on them.

Time-sensitive access control has been discussed in cloud computing, such as [12], [13], where Timed-Release Encryption (TRE) [14] is widely adopted as an effective solution. Liu *et al.* [12] proposed a proxy re-encryption based scheme, which can accurately grant the data access privilege to intended users who own a certain attribute set during a specific time period. Hong *et al.* [13] developed a time and attribute combined access control system based on the ciphertext-policy attribute-based encryption with TRE in cloud computing system, which can implement different access structures at different time.

III. System Model, Threat assumptions and Preliminaries

A. System Model

We consider an ICN-based network that consists of three kinds of entities: Content Providers (CPs), an Internet Service Provider (ISP) and users, which is shown in Fig. 1.

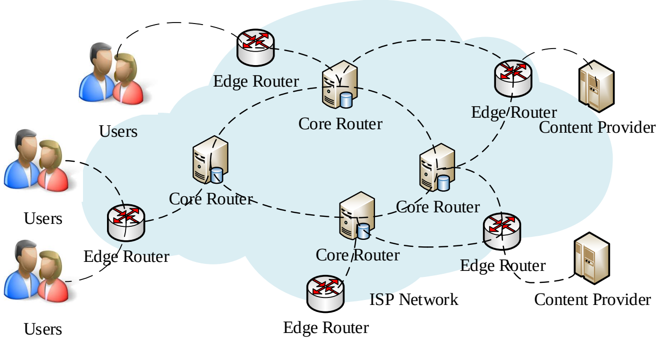

Fig. 1. System Model

CPs are responsible for publishing contents as well as dealing with users' subscriptions. They also provide some necessary information for edge routers , such as time tokens, for verification. Users subscribe to CPs and obtain contents by sending corresponding requests. A user may request contents before the releasing time to show her demand.

Routers in ISP network can be divided into two types: edge routers and core routers. Edge routers, as users' entrances into the core network, play the roles of verifying users' requests before forwarding them, obtaining the updated time tokens from CPs and collecting demand on content from users. Core routers forward data content to neighbor routers according to their Forward Information dataBase (FIB) tables. In addition, they perform in-network caching and respond to the requests if the content is stored in their own cache. The caches in some core routers can be leased by CPs for storing the pre-distribution content.

B. Threat Assumptions

In this work, CPs are assumed to be trusted and undertake the users' authority management. Users are considered to be malicious, who are curious on the content and intend to access it without sufficient access privilege. Meanwhile, we assume there exists an attacker that aims to exhaust network resources by generating illegal interest packets. ISP is assumed to be semi-trusted. On one hand, ISP is honest to obey the pre-designated protocol and maintain this access control mechanism. On the other hand, ISP is curious about the content, meaning that it will make a great effort to access the contents stored in the routers' cache.

C. Preliminaries

1) Bilinear Map: Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be multiplicative cyclic groups of the same prime order of p, and g_1 and g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. A bilinear map can be described as $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ that has the following properties: (a) *Computability:* there is an efficient algorithm to compute the map e; (b) *Bilinearity:* for all $a, b \in \mathbb{Z}_p^*$ and $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$, $e(u^a, v^b) = e(u, v)^{ab}$; (c) *Non-degeneracy:* $e(g_1, g_2) \neq 1$.

2) Complexity Assumption: Broadcast encryption used in our scheme works on the bilinear pairings with Weak Bilinear Diffie-Hellman Exponent(WBDHE) assumption.

Definition 1: WBDHE Assumption: For unknown $a \in \mathbb{Z}_p^*$, given a tuple $(P, P^a, P^{a^2}, ..., P^{a^l}) \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, it is infeasible to compute $e(P, Q)^{\frac{1}{a}}$.

IV. Proposed Scheme

A. Overview

In our proposed access control mechanism, users are divide into groups based on their different access privileges. Advanced subscribers can access more contents and obtain the latest popular contents earlier. We use different numbers as group identifers for users, where users with a larger group identifer have more privileges. The content published by CPs is time sensitive, which can be accessed by users in different groups at different time. To ensure the security of published contents and enable access control, CPs encrypt the content via symmetric encryption and encrypt the symmetric key using broadcast encryption [15] before content distribution. The time token is also integrated into broadcast encryption for the time-sensitive content distribution. Besides, to protect the network against DoS/DDoS attacks, we make edge routers authenticate users' requests before forwarding them into core network. Through a lightweight challenge-response verification, edge routers can block malicious requests while responses from legitimate users can be sent to CPs so as to distribute contents to most needed locations in advance.

B. Initialization

In this step, a CP initializes the public and private parameters as follows: (a) CP generates $I = [p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e]$,where $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a bilinear map. After that, CP chooses two randomly selected generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ such that $q = e(g_1, g_2)$;(b) Given that users are divided into n groups, CP selects a set of n random numbers: $\beta_1, \beta_2, ..., \beta_n \in \mathbb{Z}_p^*$. Then CP defines $b_i = {g_1}^{\beta_i}, i = 1, 2, ..., n$ and let B be $(b_1, b_2, ..., b_n)$; (c) Let $(I, g_2, B, H, Enc(\cdot))$ be the public parameters. Here, H is a standard hash algorithm which is used for the verification of challenge-response and $Enc(\cdot)$ is a symmetric encryption algorithm with a symmetric decryption algorithm $Dec(\cdot)$. CP keeps $(\beta_1, \beta_2, ..., \beta_n, g_1)$ as the master key.

C. User Registration

When user U_i registers to CP with the group identifer m and the identity ID_i, CP generates the decryption keys for U_i. As U_i can access contents with the privilege level no larger than m, CP selects m random numbers: $r_i^1, r_i^2, ..., r_i^m \in \mathbb{Z}_p^*$ and then computes:

$$s_i^j = {g_1}^{r_i^j/(\beta_j + r_i^j)}, u_i^j = {g_2}^{1/(\beta_j + r_i^j)}, \tag{1}$$

for $j = 1, 2, ..., m$. Denote $R_i = (r_i^1, r_i^2, ..., r_i^m)$, $S_i = (s_i^1, s_i^2, ..., s_i^m)$ and $U_i = (u_i^1, u_i^2, ..., u_i^m)$.

Then, CP stores $\Gamma = (ID_i, R_i, S_i, U_i, m)$ in the user list for further use and also sends it to U_i as her private key.

D. Content Publication

Assume that there is a content item M with the access privilege level τ at the time t_1, and later the access privilege level of M changes to τ' after the time t_2, where $\tau > \tau'$. In this case, after the time t_1, M is published and users with group identifer larger than τ can obtain M. When time t_2 comes, CP releases the content access privilege level to τ' and users with group identifer larger than τ' could also access it. We use this example to illustrate our scheme in the rest of this section. Before CP publishes the content item M, it uses symmetric encryption to encrypt M with a randomly selected symmetric key K. Besides, CP encrypts the symmetric encryption key K and computes $CK = Enc_{k'}(K)$, where $k' = q^{k+\alpha t}$. $k, \alpha \in \mathbb{Z}_p^*$ are randomly selected numbers and t is the value related to the time. We define $k_1 = q^k$ and $k_2 = q^{\alpha t}$. Then, CP encrypts k_1 with broadcast encryption by computing $C_1 = {b_{\tau'}}^{k_1}$, $C_2 = {g_2}^{k_1}$. The user with group identifer larger than the lowest access privilege of M, i.e., τ', could recover k_1 from C_1, C_2. After that, CP generates multiple number pairs:

$$D_1 = \{Enc_{k_1}(x_{i'}) | i' = 1, 2, ...\}, D_2 = \{H(x_{i'}) | i' = 1, 2...\},$$

where $\Delta = \{x_{i'} | i' = 1, 2, ...\}$ are randomly selected numbers. D_1, D_2 are used for the challenge-response verification at edge routers, which are also called as verification samples. Then, CP encrypts k_2 by computing $C_3 = {b_\tau}^{\alpha t}$, $C_4 = {g_2}^{\alpha t}$, as a time token for the content item M at the time t_1. It means that the access privilege level of M has been set to τ and the users with group identifier larger than τ can recover k_2. CP also computes:

$$D_3 = \{Enc_{k_2}(y_{i'}) | i' = 1, 2, ...\}, D_4 = \{H(y_{i'}) | i' = 1, 2...\},$$

where $\Lambda = \{y_{i'} | i' = 1, 2, ...\}$ are randomly selected number. D_3, D_4 are also used as samples for challenge-response verification of the time token.

Finally, before the content item M is published at the time t_1, CP sends $(F, PL, CK, C_1, C_2, D_1, D_2)$ to edge routers, where $C_1 = {b_{\tau'}}^{k_1}$ and $C_2 = {g_2}^{k_1}$, F is the name of this content and $PL = \tau$. Each edge router stores those values for verifying users' requests. CP doesn't send the time token of the content item M to edge routers because the content hasn't been published yet. CP also saves Δ and Λ for further use, such as verifying messages returned from edge routers.

E. Request Authentication

In this subsection, we describe the three phases during the access privilege releasing process of a content: before published, being published at first, and privilege expansion.

Phase I: Before Being Published

In this phase, the time is before t_1 and the content item M hasn't been published. Hence, no one has the access privilege to M. However, CP can collect the demand of the content item M from the responses to users' requests. If a user wants to access M, she will send a request for the content to the edge router. When the edge router receives the user's request, it verifies whether the user has the access privilege through a challenge-response mechanism. Only the legitimate users' demand will be forward to CP in order to prevent malicious users from submitting misleading demand information to CP.

We assume that a user U_i with group identifer m wants to access the content item M. The user sends a request to the edge router which includes F, the name of the content item M, and a temporary ID connected with the edge router. The edge router searches F in its content name list and sends the corresponding C_1, C_2 with a randomly selected sample $d_1^\gamma \in D_1$, where $\gamma = 1, 2, ...$, to U_i as a challenge: C_1, C_2, d_1^γ, PL. Also, CP finds $d_2^\gamma \in D_2$ for verifying response. If U_i has the access privilege to the content item M, i.e., $m > \tau'$, the user can compute q^k with C_1 and C_2 by

$$\begin{aligned} e(C_1, u_i^{\tau'})e(s_i^{\tau'}, C_2) =& e(g_1, g_2)^{\frac{k \cdot \beta_{\tau'}}{\beta_{\tau'} + r_i^{\tau'}}} e(g_1, g_2)^{\frac{k \cdot r_i^{\tau'}}{\beta_{\tau'} + r_i^{\tau'}}} \\ =& q^k = k_1. \end{aligned} \tag{2}$$

Then, the user recovers x_γ from d_1^γ by computing $X = Dec_{k_1}(d_1^\gamma)$ and returns X to the edge router as a response.

The edge router compares $H(X)$ with d_2^γ for verifying user's privilege. If they are the same, the edge router saves X as a proof and discards this verification sample so that the response cannot be used again in another verification. When all samples are exhausted, the edge router asks CP for more.

Edge routers regularly report back to CP with proofs before the time t_1 comes. CP randomly checks proofs by verifying whether $X \in \Delta$ and counts the number of them which is an indicator related to the users' demand for M in the region of this edge router. CP could store the content at the local cache near the high-demand area in advance, which will reduce the pressure of the content distribution when it is published and improve users' experience with a low file retrieval delay.

```
Algorithm 1: Privilege Verification Response
Input: The content's access privilege label, $PL = l$;
       The content's lowest access privilege, $l'$;
       The verification parameters, $C_1, C_2$;
       The sample for privilege verification, $d_1^\gamma$;
       The time token of content, $C_3, C_4$;
       The sample for time token verification, $d_3^\lambda$;
       The $U_i$ with group identifier, $m$;
Output: The response, $X, Y$; The secret key, $k_C$
if $m > l$ then
    Select $(s_i^l, u_i^l)$ and $(s_i^{l'}, u_i^{l'})$;
    Compute: $MK_1 = e(C_1, u_i^{l'})e(s_i^{l'}, C_2)$,
    $MK_2 = e(C_3, u_i^l)e(s_i^l, C_4)$;
    Compute: $X = Dec_{MK_1}(d_1^\gamma)$,
    $Y = Dec_{MK_2}(d_3^\lambda)$;
    Set: $k_t = MK_1 * MK_2$;
    Compute: $k_C = Dec_{k_t}(CK)$;
end
Return $X,\ Y,\ k_C$;
```

Phase II: Being Published at First

When the content item M is published at the time t_1, CP sends the time token with samples to the edge routers: C_3, C_4, D_3, D_4, where $C_3 = {b_\tau}^{\alpha t}$ and $C_4 = {g_2}^{\alpha t}$. When the user U_i with group identifer m wants to access this content item and sends the request to the edge router, the edge router returns this challenge with randomly selected unused samples from D_1 and D_3 to U_i:

$$(CK, C_1, C_2, d_1^\gamma, C_3, C_4, d_3^\lambda),$$

where $d_1^\gamma \in D_1$ and $d_3^\lambda \in D_3$. the user can run **Algorithm 1** to obtain the symmetric key K and the response of the challenge. If the user's group identifier m is larger than τ, according to the Eq.(6), she recovers q^k from C_1, C_2 and $q^{\alpha t}$ from C_3, C_4 similarly. The symmetric key can also be decrypted and the user sends the challenge with X and Y back to the edge routers as the response. The edge router compares X, Y with d_2^γ, d_4^λ. If they are the same, it forwards the request into the network and returns encrypted content item M when the item is received from the network. The user can decrypt the content item with a correctly k_c and then obtain M.

If $\tau' < m < \tau$, the user cannot access the content now but will have the access privilege in the future. The user responds the d_1^γ correctly and the edge router can choose to report the demand to CP. The request from this user will be discarded then.

Phase III: Privilege Expansion

In this phase, CP updates the time token to $C_3 = {b_{\tau'}}^{\alpha t}$ and $C_4 = {g_2}^{\alpha t}$. CP also changes the content's privilege label PL to τ'. It means that the content's access privilege is released to τ'. After edge routers update the time token, the authentication process is the same as **Phase II**. At this time, users with group identifier $m > \tau'$ have access to the content. If the content item M has additional access privilege releasing plans, CP only needs to update the time token when the new releasing time comes. For example, if the access privilege level of content item M is released to σ at the time t_3, CP updates the time token by computing $C_3 = {b_\sigma}^{\alpha t}$, $C_4 = {g_2}^{\alpha t}$, and PL of the content. Meanwhile, the lowest access privilege of M has been changed to σ, CP also needs to update C_1 and C_2 by computing $C_1 = {b_\sigma}^{k_1}, C_2 = {g_2}^{k_1}$.

V. Security and Performance analysis

A. Security Analysis

*1) **Data Confidentiality**:* Our proposed scheme protects data confidentiality from both malicious routers and users. Given content item M, before it is published, CP doesn't send the time token into the network. Therefore, there is no information about $q^{\alpha t}$. In other words, no one has the chance to obtain k' and K which are used to access the content.

After the content item M is published, only the users with the satisfied access privilege can obtain the content. Given $C_3 = {g_1}^{\alpha t \beta_\tau}$, $C_4 = {g_2}^{\alpha t}$ and g_2 without the knowledge of β_τ, if an attacker can also compute $q^{\alpha t}$ as follows:

$$e(C_3, g_2)^{1/\beta_\tau} = e(g_1, g_2)^{\alpha t} = q^{\alpha t}, \tag{3}$$

it obviously contradicts with **WBDHE** assumption. Similarly, given $C_3 = {g_1}^{k\beta_{\tau'}}$, $C_4 = {g_2}^k$ and g_2 without the knowledge of $\beta_{\tau'}$, if an attacker can compute q^k as follows:

$$e(C_4, g_2)^{1/\beta_{\tau'}} = e(g_1, g_2)^k = q^k, \tag{4}$$

it also obviously contradicts with **WBDHE** assumption. Without k_1 and k_2, no one can compute k' and K which means the attacker cannot access the content item M.

In the challenge-response process, other verification information that is transmitted includes $d_1^\gamma, d_2^\gamma, d_3^\lambda, d_4^\lambda, X, Y$, where d_1^γ, d_2^γ, d_3^λ ,d_4^λ are randomly selected from D_1, D_2, D_3, D_4. All of them are not related to k'. Thus, with this information, it's still impossible for the users and routers to gain any useful knowledge and obtain the content. In conclusion, the data confidentiality can be ensured.

*2) **Unforgeability**:* The challenge requires users to decrypt q^k and $q^{\alpha t}$ from C_1, C_2, C_3, and C_4. Users need to compute d_2^γ, d_4^λ from d_1^γ, d_3^λ with q^k and $q^{\alpha t}$, where d_1^γ, d_3^λ are randomly selected from D_1, D_3. Computing the x_γ, y_λ without q^k and $q^{\alpha t}$ is impossible because of the security of the symmetrical encryption and hash function. All verification samples are different and the edge routers discard the random samples after successful verification. Hence, each response is used only once, and the attacker cannot forge a response by reusing it.

*3) **Anonymity**:* In the verification process, the edge routers examine whether $H(X) = d_2^\lambda$, $H(Y) = d_4^\lambda$ is established. They only know that users have the access to the content but cannot obtain the accurate group identifers of users. Moreover, the verification process only require the edge routers and users to exchange some randomly selected parameters. Thus, attackers cannot find any private information from them. At the same time, when a user communicate with the network, she uses a temporary identifer that is not related to the real identity of the user. Hence, it also protects users' anonymity.

*4) **DoS/DDoS Attack Resistance**:* In our scheme, each request has to be verified by the edge router and the requests from unauthorized users are blocked from the core network. Therefore, it is hard for attackers to exhaust the resource of the core network with vast amount of requests.

B. Performance Analysis

To evaluate the performance of our scheme, we use GNU Multiple Precision Arithmetic(GMP) library and Pairing-Based Cryptography(PBC) library to implement the encryption process and verification process. We also simulate our scheme with NS-3 and ndnSIM and compare its performance with standard NDN. All experiments are conducted on Ubuntu 16.04 LTS with 2.4GHz Intel Core i7 processor and 16G RAM.

1) Algorithm Implementation: The content is protected by symmetric encryption. The symmetric encryption key K is stored as k' with the broadcast encryption that is implemented using an elliptic curve with 160-bit group order and offers approximately the same security level as 1024-bit RSA. We choose AES-256 and SHA-256 in the sample generation process and verification process. **Table I** shows the time cost for the cryptographic operations.

TABLE I
COMPUTATION COST FOR CRYPTOGRAPHIC OPERATIONS

Operator	Time(ms)
Verification Operator generation	2.781
Time-limit Operator generation	2.805
Users response(for demands)	1.486
Users response(for contents)	4.565
SHA-256	$< 10^{-4}$
AES-256	0.03(1K)

In our scheme, CP generates verification parameters and time tokens in 5.586ms on average. The verification samples use numbers randomly selected in the range of 2^{1024} and are generated with AES-256 and SHA-256 in 0.046ms on average. When the verification process occurs at the edge routers, it doesn't need any extra computation resource. Users respond to the challenge with the time cost 1.486ms or 4.565ms which are different from the feedback of demands or requesting contents, respectively. Fig. 2 illustrates the encryption and decryption time of our scheme for different content size. In our scheme, the symmetric encryption and decryption time increases with the content size. Users obtain the key after the challenge with the computation time cost 4.565ms on the average.

2) Network simulation: We compare our scheme with the standard NDN in ndnSIM 2.6. The topology of the network simulation is generated by using the two-layer top-down hierarchical model in BRITE which has 1000 nodes and 1820 edges. The number of users is 20% of routers and 10 edge routers are linked to users with 100 Mbps bandwidth and 5ms delay on average. The core routers are linked to each other with bandwidth 1Gbps with 10ms delay. We randomly select a CP from the core routers and it responds to all the requests from users through edge routers. The delay of the challenge-response is measured as 5.345ms from the edge routers to users and 5.643ms from users to the edge routers without considering the time for computation. We simulate the file retrieval delay for different file sizes. We randomly select different sizes: 10KB, 20KB, 40KB, 80KB and 100KB.

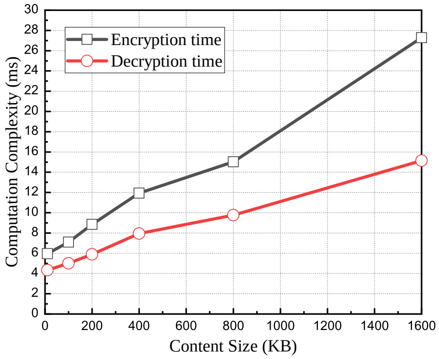

Fig. 2. Computation Complexity & Content Size

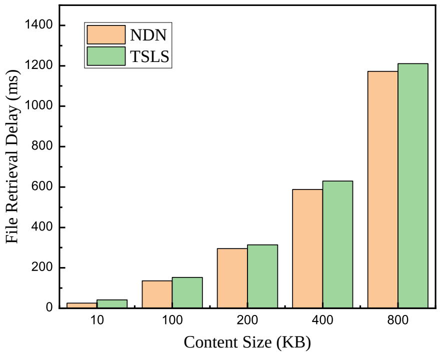

Fig. 3. File Retrieval Delay & Content Size

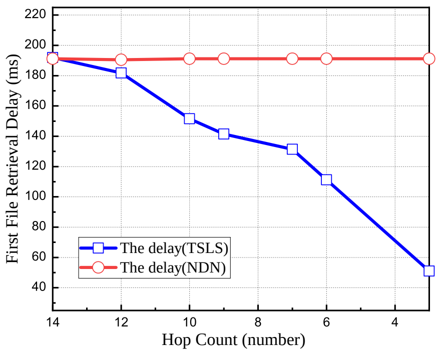

Fig. 4. First File Retrieval Delay & Hop Count

The chunk size we use in the simulation is 10KB. We assume that users use right prefix to request chunks and always send successive chunks' requests belonging to the same file. For a fair comparison, CP doesn't publish contents into the network in advance. Fig. 3 shows the simulation results as compared to the standard NDN. It can be observed that the delay in our scheme is low with only a slight difference of 37.56ms when the content size is 800KB. The cost of our scheme results from the extra computation on the verification. Moreover, compared with the network delay to transmit the content, it is small enough to be ignored. To assess the effectiveness of publishing content in advance, we compare our scheme with standard NDN. When CP receives demand from the users linked with the same edge router, our scheme store content at three core router caches which are closed to users in advance. Fig. 4 shows the delay when users first receive content from the network. The content size is 100KB. It shows that our scheme can reduce the delay for the users to receive the content by up to almost 70%.

VI. Conclusion

In this paper, we proposed a time-sensitive, lightweight, and secure access control solution, called TSLS, for Information Centric Networking. The access control is enabled through a challenge-response based verification at edge routers. We introduced the broadcast encryption integrated with time token mechanism to allow content providers to flexibly release the access privilege of content to different users at different time. Besides, the responses received in verification process at edge routers are used to forecast users' demand and pre-distribute the content to improve users' experience. The experimental evaluation shows that our scheme has the acceptable computational cost and low transmission delay.

Acknowledgment

This work is supported in part by the National Key Research and Development Plan of China under Grant No. 2017YFB0801702, Youth Innovation Promotion Association CAS under Grant No. 2016394 and the National Natural Science Foundation of China under Grant No. 61671420.

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)*. ACM, 2009, pp. 1–12.

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

[3] 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, Available*, 2017.

[4] Q. Zheng, G. Wang, R. Ravindran, and A. Azgin, "Achieving secure and scalable data access control in information-centric networking," in *Proceedings of the 2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 5367–5373.

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

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

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

[8] D. Kim, J. Bi, A. V. Vasilakos, and I. Yeom, "Security of cached content in NDN," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2933–2944, 2017.

[9] 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.

[10] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage," *IEEE Transactions on parallel and distributed systems*, vol. 27, no. 5, pp. 1484–1496, 2016.

[11] K. Xue, J. Hong, Y. Xue *et al.*, "CABE: A new comparable attribute-based encryption construction with 0-encoding and 1-encoding," *IEEE Transactions on Computers*, vol. 66, no. 9, pp. 1491–1503, 2017.

[12] Q. Liu, G. Wang, and J. Wu, "Time-based proxy re-encryption scheme for secure data sharing in a cloud environment," *Information sciences*, vol. 258, pp. 355–370, 2014.

[13] J. Hong, K. Xue, Y. Xue, W. Chen, D. S. Wei *et al.*, "TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud," *IEEE Transactions on Services Computing, Available*, 2017.

[14] R. L. Rivest, A. Shamir, and D. A. Wagner, "Time-lock puzzles and timed-release crypto," in *Massachusetts Institute of Technology*, 1996.

[15] C. Delerablée, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys," in *Proceedings of 2007 International Conference on Pairing-Based Cryptography*. Springer, 2007, pp. 39–59.