Enforcing Access Control in Information-Centric Edge Networking

Danye Wu, Zhiwei Xu, *Member, IEEE*, Bo Chen, *Member, IEEE*,
Yujun Zhang, *Member, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*— By moving computing resources close to where they are needed (i.e., the network edges), edge computing can significantly reduce burden on the centric cloud data centers. However, extreme scale of on-line big data may impose a significant burden on the network backbones. Information-centric edge networking can address this challenge by incorporating in-network caching into edge networks. This however, opens a door for many new security issues and requires various security defenses. One of those is efficient access control design specifically for information-centric edge networking. In this work, we aim to design an efficient and secure access control scheme for information-centric edge networking. In our design, we propose the confidentiality-enhanced network coding which can ensure that, without having access to the authorization key, the attacker will not be able to obtain the original content. And thanks to the properties of confidentiality-enhanced network coding, highly efficient access control can be realized by encrypting only part of the encoding matrix. In addition, our design can allow efficiently revoking users. Security analysis and experimental evaluation on NS3 demonstrate that our scheme can successfully enforce access control in information-centric edge networking with a small overhead.**

***Index Terms*— Edge computing, in-network caching, access control, confidentiality-enhanced network coding.**

Manuscript received February 15, 2020; revised August 1, 2020; accepted September 16, 2020. Date of publication September 24, 2020; date of current version January 15, 2021. This work was supported in whole or in part, by National Key Research and Development Program of China (2018YFB1800403), the research program of Network Computing Innovation Research Institute (E061010003), National Science Foundation of China (61902382, 61972381, 61672500, 61962045), China Scholarship Council (201904910802); This work was carried out with the support of China Environment for Network Innovations (CENI); and partially supported by NSF EARS-1839818, CNS1717454, CNS-1731424, and CNS-1702850. The associate editor coordinating the review of this article and approving it for publication was T. He. *(Corresponding author: Yujun Zhang.)*

Danye Wu and Zhiwei Xu are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: wudanye@ict.ac.cn; xuzhiwei2001@ict.ac.cn).

Bo Chen is with the Department of Computer Science, Michigan Technological University, Houghton, MI 49931 USA (e-mail: bchen@mtu.edu).

Yujun Zhang is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Nanjing Institute of ICT, CAS, Nanjing 210000, China (e-mail: nrcyujun@ict.ac.cn).

Zhu Han is with the Electrical and Computer Engineering Department, University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea (e-mail: zhan2@uh.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2020.3026380

I. Introduction

NOWADAYS, with the growing number of edge devices, the network bears huge burden in transmission overhead. However, the traditional centralized cloud computing cannot satisfy the increasing demand at edge networks. Meanwhile, the computation power and storage space distributed at edge networks can greatly improve the task-processing capability, and computing at edge network can help to reduce pressure. Edge computing takes out a portion of computational resource as well as memory from the data center, moving them closer to the location where they are needed (i.e., the network edge). This would bring great benefits since a large amount of data can now be processed on the edge and the data that must be moved to the central clouds are significantly reduced. This new computing paradigm well suits the need of growing data in an extreme scale in the near future.

The extreme scale of edge devices also makes it a challenging task of exchanging content among computing devices in edge networks as well as delivering content from the central clouds to the computing devices located in the edge networks, which may impose a significant burden on the network backbones. To resolve this challenge, a viable solution would be caching data in close proximity to users [1], such that the popular data cached in the edge networks can be reused asynchronously by many users in the same edge networks. This can bring significant benefits, including: 1) It can boost spectral efficiency and reduce energy consumption of wireless systems [2], improving quality of user experience. 2) It can significantly reduce backhaul offloading [3], alleviating burden on the core Internet as well as data centers. 3) It well supports the content intensive applications like delivering adaptive video streaming [4] and augmented reality. 4) It is compatible with the emerging cellular mobile communications technique 5G, which also incorporates caching [1].

As a promising architectural design for future Internet, information-centric networking (ICN) caches content in routers (i.e., in-network caching) to support efficient content forwarding. Compared to other existing network caching mechanisms like content delivery networks (CDNs) which rely on deploying proxy servers, ICN is more advantageous because: First, ICN deploys cache in the network layer, which usually incurs less overhead compared to a caching mechanism that deploys cache in the application layer. Second, the caching mechanism in the network layer is transparent to applications.

Therefore, it is a promising alternative of deploying cache in the edge networks using ICN. Also, we emphasize that deploying ICN in the edge networks is much less challenging than deploying it in the core Internet [5], since a majority of the edge networks are under construction while the core Internet has been well established. Deploying cache in the edge networks will facilitate content delivery, which however, will bring a side effect that due to caching of data, data publishers will lose direct control of their data, and hence are difficult to enforce access control over them [6]. To the best of our knowledge, there is no access control mechanism specifically designed for edge networks using ICN in literature. *Traditional access control mechanisms for information-centric networking [7]–[10] rely on either expensive encryption/decryption or authentication of packets or complicated message exchange [11], which usually incur large overhead and are not suitable for edge networks with extreme scales*.

In this work, we aim to design an efficient and secure access control scheme for ICN-based edge networking. Our key insights are three-fold: *First*, we use network coding to encode data being delivered in edge networks. There are two main benefits when applying network coding here: 1) Network coding is originally designed for content delivery, and can distribute content more efficiently since the destination can decode and obtain the original content after having received a sufficient number of network coded segments, eliminating unnecessary waiting time of receiving the entire original content. 2) The problem of enforcing access control over large volumes of data can be converted to enforcing access control over encoding vector in the matrix of network coding, which is small and requires a much less number of encryption/decryption operations, thus highly improving the efficiency. *Second*, to prevent the attacker from recovering all or portion of the original content, we propose the confidentiality-enhanced network coding by performing the following steps: 1) We apply a linear all-or-nothing transform (AONT) [12], [13] on the original content. In this manner, the attacker cannot learn anything about the original content without having obtained the entire AONT-transformed content. 2) We encrypt one vector of the encoding matrix for network coding (which will be applied to the AONT-transformed content for network coding), using a secret key only known to both the publisher and the legitimate users. This is advantageous, since we only need to protect a small vector using encryption, which can be efficiently done. The efficiency can be even further optimized by only encrypting a few elements[1] in the vector. The rationale of confidentiality-enhanced network coding is, by preventing the adversary from obtaining full knowledge of the encoding matrix, the adversary is always not able to decode the entire AONT-transformed content, and due to the "all or nothing" nature of AONT, the adversary is not able to learn anything about the original content. *Third*, we design an efficient revocation scheme that can revoke access privilege from the expired users. Our key ideas are: 1) Due to the use of confidentiality-enhanced network coding, we can simply change a portion of the encoding matrix (e.g., one vector), re-encode the original content, and keep this new portion of the encoding matrix secret from the revoked users (i.e., by encrypting it with a new key which will be known only by the legitimate users); 2) Since only a portion of the encoding matrix has been updated, most resulting network coded segments after re-encoding will remain the same, and therefore, most segments cached previously in the routers can be reused, and at most one segment is out-of-date and needs to be updated; 3) This out-of-date segment will be updated in an incremental way from the network coded segments cached in routers by performing Gaussian elimination once expired, and the newly updated segment will be distributed and incorporated into the cached network coded segments afterwards. In this way, the revoked users will not be able to decode the original content from the updated network coded segments since they are not able to obtain the entire encoding matrix that has been updated.

Since in-network caching has been implemented by various ICN architectures (e.g., Named Data Networking (NDN) [14], Content Centric Networking (CCN) [15], Publish/Subscribe Internet Routing Paradigm (PSIRP) [16]), we use NDN as a representation. However, our design can also be adapted to other ICN architectures.

Contributions. Our contributions are summarized as follows:

- We design ACET, the secure and efficient access control framework specifically for ICN-based edge networking. Our design ensures efficiency by utilizing confidentiality-enhanced network coding, such that access control can be enforced by encrypting a small portion of the encoding matrix for network coding.
- We design an efficient revocation mechanism in which an expired user can be revoked efficiently by re-encrypting a small portion of the encoding matrix and re-using most network coded segments cached in each router.
- We analyze security of ACET. In addition, we implement ACET in NS3, and experimentally validate its performance.

This article is an extended version of our previous conference paper [17]. We summarize major differences in the following. 1) A novel network coding is proposed to encode and keep the original content confidential in an efficient way. Compared to the old design which applies AONT to the encoding matrix, the new design can allow efficiently revoking users by re-encrypting one vector of the encoding matrix which only slightly affects the resulting encoding content. 2) The confidentiality-enhanced network coding further optimizes the overall performance by only requiring encrypting a few elements in a coding vector of network coding which is more efficient, and hence more suitable for edge computing. 3) The new design introduces a new timestamp-based revocation scheme, which achieves efficient access authority revocation by incremental coding segment updating.

Paper organization. Section II introduces necessary background of this work. In Section III, we explain our attack model, security definition and assumptions. Sections IV and V describe our main design on access control and authoriza-

[1] The number of elements being encrypted is determined by ensuring it is computationally infeasible for the adversary to brute-force the encrypted elements.

tion revocation, respectively. We provide security analysis and discussion in Section VI and performance evaluation in Section VII. We summarize the related work in Section VIII and conclude in Section IX.

II. Background

A. Information-Centric Networking

Information-centric networking (ICN) is a new Internet architecture which focuses on the name instead of the location of the information. ICN enables in-network caching and replication to facilitate content delivery, which can improve both efficiency and robustness of the network. There are a few implementations of ICN, including CCN and NDN, and we mainly focus on NDN in this work. Unlike traditional IP network architectures, NDN uses a hierarchical name structure instead of an IP address to direct packet routing and transport. There are two types of data packets in the NDN, *interest* and *content*. To request the content, the user will send out an Interest message containing the content name. The requested content will be sent back by the routers (if they cache the content) or the data publisher (if the routers do not have the content in their cache). The NDN network maintains three types of data structures: Forwarding Information Base (FIB), Content Store (CS), and Pending Interest Table (PIT). The FIB retains the next hop interface for the router to reach the data publisher. The CS stores the cached content. The PIT keeps track of the Interest not yet responded as well as its arrival interface so that the requested content can return along the reverse path.

B. All-or-Nothing Transform (AONT)

$AONT$ [12] converts data into an encoded format, with the property that it is hard to invert the encoded format back to the original data unless all of the encoded output is known. Linear AONT [13] is a linear transform which can maintain the property of AONT while being able to further reduce the computational complexity. Stinson [13] defines the linear AONT as follows:

Definition 1: Given a positive integer n, a finite field $\mathbb{F}_q$ with order q, a function π which maps an input of n-tuple $(x_1, \ldots, x_i, \ldots, x_n)$ to an output of n-tuple $(y_1, \ldots, y_i, \ldots, y_n)$, where $x_i, y_i \in \mathbb{F}_q$ and $1 \leq i \leq n$, we say π is a linear $(n, q) - AONT$, if it satisfies the following conditions:

- *π is a bijection;*
- *Each y_i $(1 \leq i \leq n)$ is an $\mathbb{F}_q$-linear function of $x_1, \ldots, x_i, \ldots, x_n$ $(1 \leq i \leq n)$;*
- *If any $n - 1$ out of n output values $y_1, \ldots, y_i, \ldots, y_n$ are fixed, any input value x_i $(1 \leq i \leq n)$ is completely undetermined.*

An $n \times n$ encoding matrix for the linear (n, q)-AONT can be constructed as [13]

$$M = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 \\ 1 & 1 & \cdots & 1 & \lambda \end{pmatrix}.$$

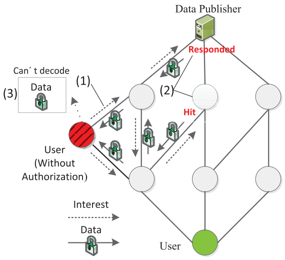


Fig. 1. Unauthorized users.

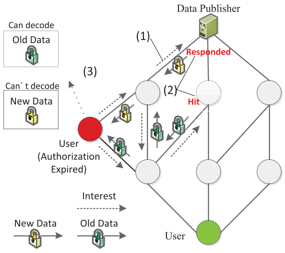


Fig. 2. Expired users.

Each element in M is chosen from the finite field $\mathbb{F}_q$, in which $q = p^k$, and p is a prime number and k is a positive integer. $\lambda \in \mathbb{F}_q$ such that $\lambda \notin \{(n-1)\, mod\ p,\ (n-2)\, mod\ p\}$.

C. Linear Network Coding

Linear network coding [18] is usually used to improve a network's throughput. In a network using linear network coding, the network nodes take several packets and linearly combine them together for further transmission to achieve the maximum possible information flow, instead of simply relaying the packets being received.

Random linear network coding [19]–[21] is a special type of simple yet powerful linear network coding schemes. It works as follows: The content publisher divides the content into a number of segments. He/She then generates an encoding matrix, in which each element is chosen uniformly at random from a sufficiently large finite field. He/She then applies the encoding matrix over the segments, generating a few coded segments which will then be disseminated into the network. The routers linearly combine the received segments from upstream link utilizing coefficients chosen uniformly at random from the same finite field. The generated segments are forwarded to the downstream link. After having received a sufficient number of segments, the user will decode them, obtaining the original content.

III. Attack Model, Security Definition and Assumptions

Attack model. We mainly consider two types of attackers, as shown in Figures 1 and 2 respectively. The first type of attacker captures a user which has not been authorized to access the content, i.e., no access privilege. The attacker will perform the following steps (Figure 1): 1) It sends out an Interest packet to the network to request the content. 2) Either the publisher or the intermediate routers respond the Interest with the "access-control-protected" format of the content following the reverse path where the Interest comes. 3) The

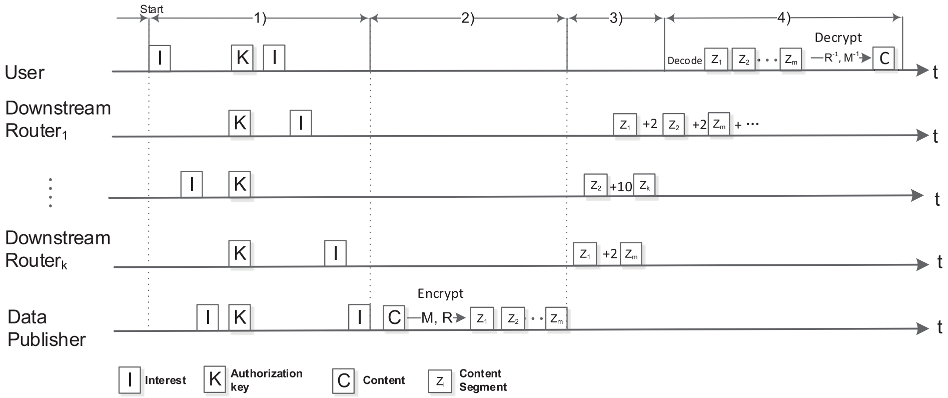


Fig. 3. System Design.

attacker tries to decode the access-control-protected format to extract the original content, which should not be successful if the access control scheme is secure. The second type of attacker captures a user whose authorization is expired, e.g., the user does not pay the subscription and his/her authorization has been revoked. The attacker will perform the following steps (Figure 2): 1) It sends out an Interest packet to the network to request the content. 2) The publisher responds the Interest with an updated "access-control-protected" format. 3) The user tries to extract the original content by decoding the updated "access-control-protected" format, which should not be successful if the revocation scheme is secure.

Security definition. Let $\mathscr{S}$ be a scheme that enforces access control in ICN-based edge networking. Let p_1 be the probability that an attacker or expired user that can successfully obtain the original content published by a publisher, and p_2 be the probability that a legitimate user can successfully obtain this content. We say $\mathscr{S}$ is secure if and only if the following two conditions can be satisfied simultaneously: 1) $p_1 \rightarrow 0$; and 2) $p_2 \rightarrow 1$.

Assumptions. We make a few assumptions. First, we assume there is a secure channel for distributing keys. The secure key distribution is an orthogonal problem and not the focus of this paper. Second, we assume the routers in the network will honestly follow the protocol, e.g., honestly perform network coding, and honestly forward both the content and the Interests. We also assume that only legitimate user can obtain the authorization key to decrypt the content, and legitimate users will not leak their authorization keys.

IV. ACET: An Efficient Access Control Scheme for Edge Networks Using NDN

In this section, we design ACET, an efficient and secure Access Control scheme specifically for Edge neTworks using named data networking for content delivery. We use confidentiality-enhanced network coding to transform the original content. In this way, we can simply encrypt one vector of the network coding matrix and, without being able to decrypt this vector, the adversary is not able to decode and obtain all the transformed content, and hence not able to obtain anything about the original content thanks to the nice property of confidentiality-enhanced network coding. As shown in Figure 3, our scheme mainly consists of four steps: 1) pre-fetching authorization keys (by *the user*) and forwarding Interests (by both *the user the router*); 2) pre-processing content (by *the data publisher*); 3) forwarding network coded segments and Interests (by *the routers*); 4) decoding segments (by *the user*). An optimization of ACET has been provided, which achieves a similar security level while avoiding unnecessary encryption to further improve efficiency. We describe the detailed steps as follows.

A. Pre-Fetching Authorization Keys and Forwarding Interests Initially

Initially, a legitimate user obtains a secret key k for access authorization from the data publisher, before it can request content from the network. The attacker should not be able to obtain this secret key k (see Sec. III).

To request specific content C for the first time, the user will send out an Interest. The Interest will include the name of content C. When a router receives the Interest, it will check its PIT and FIB to forward the Interest to the corresponding data publisher. Note that we assume this user is the first one which retrieves content C and there is no content C cached in any routers yet.

B. Pre-Processing Content Being Requested

Once receiving the Interest request for content C, the data publisher will find C and encode it using confidentiality-enhanced network coding following these steps:

(a) The $m \times n$ matrix content C being protected can be viewed as a collection of m segments $[C_1\ C_2\ \ldots\ C_m]^T$. The data publisher generates a linear $(m, q) - AONT$ matrix M (see Sec. II-B) and applies the linear AONT matrix M on C, obtaining M', an $m \times n$ matrix, which consists of m segments:

$$M' = MC = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 \\ 1 & 1 & \cdots & 1 & \lambda \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ \cdots \\ C_{m-1} \\ C_m \end{pmatrix} = \begin{pmatrix} M'_1 \\ M'_2 \\ \cdots \\ M'_m \end{pmatrix}.$$

In this way, the attacker can not learn anything of C if he/she is not able to learn all the information in M'.

(b) The data publisher constructs an $m \times m$ encoding matrix R on finite field $\mathbb{F}_q$. Each element of R is chosen uniformly at random from $\mathbb{F}_q$. Then the data publisher uses the encoding matrix R to process M' (i.e., network coding), obtaining an $m \times n$ matrix R':

$$R' = RM' = \begin{pmatrix} R_1 \\ R_2 \\ \cdots \\ R_m \end{pmatrix} \begin{pmatrix} M'_1 \\ M'_2 \\ \cdots \\ M'_m \end{pmatrix} = \begin{pmatrix} R'_1 \\ R'_2 \\ \cdots \\ R'_m \end{pmatrix}.$$

where R'_i is a vector of n elements, and $1 \le i \le m$.

(c) The data publisher encrypts one vector in encoding matrix R with the authorization key k. For simplicity of presentation, we assume that R_m is encrypted. Let e be a symmetric encryption, then the data publisher re-computes R as: $R = [R_1\, R_2\, \cdots\, R_{m-1}\, e_k(R_m)]^T$. After processing C, the resulting data Z consists of two components: $Z = \left[R'\ R\right] = \begin{pmatrix} R'_1 & R_1 \\ R'_2 & R_2 \\ \vdots & \vdots \\ R'_m & e_k\left(R_m\right) \end{pmatrix} = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_m \end{pmatrix}$.

The data publisher uses an identity matrix I of order m as the initial encoding coefficients: $I = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} = \begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_{m-1} \\ I_m \end{pmatrix}$. The data publisher will send each vector Z_i as a coded segment to the network along with its initial encoding coefficients vector $I_i\,(1 \le i \le m)$. The generated coded segments should have the same name prefix but different segment indexes.

C. Forwarding Network Coded Segments and Interests

After a downstream router receives a network coded segment, P_{ii}, with its corresponding encoding coefficient vector Q_{ii}, where $P_{ii} = \alpha_1 \cdot Z_1 + \alpha_2 \cdot Z_2 + \cdots + \alpha_m \cdot Z_m$, $Q_{ii} = \alpha_1 \cdot I_1 + \alpha_2 \cdot I_2 + \cdots + \alpha_m \cdot I_m$ and $\alpha_h \in \mathbb{F}_q\,(1 \le h \le m)$, it will perform the following operations:

First, we need to check the content stored in the router whether there is a corresponding Interest PIT entry. If not, we will discard the received data. Otherwise, we will continue the following steps:

- If there are no network coded segments with the same name prefix present in the CS, P_{ii} will be cached in the CS and forwarded to other routers. Moreover, we will record the number of certain forwarded coded segments through this face until it reaches a sufficient quantity.
- If there are s $(s \ge 1)$ coded segments with the same name prefix present in the CS: $P_1, P_2, \cdots, P_s$, the router will first check whether P_{ii} is linearly independent with the s coded segments by comparing the corresponding encoding coefficient vectors: Q_1, Q_2, $\cdots$, Q_s with Q_{ii}. If Q_{ii} is linearly dependent of the s network coded segments, it will be discarded. Otherwise, the router will randomly combine all these segments and their encoding coefficient vectors to generate new network coded segments: $P'_{ii} = a_0 \cdot P_{ii} + a_1 \cdot P_1 + \cdots + a_s \cdot P_s$, $Q'_{ii} = a_0 \cdot Q_{ii} + a_1 \cdot Q_1 + \cdots + a_s \cdot Q_s$, where integers $a_0, a_1, \cdots, a_s$ are chosen uniformly at random from $\mathbb{F}_q$. In addition, the router will cache the new coded segment P'_{ii} and the corresponding coefficient vector Q'_{ii} in its CS and forward them to the downstream routers. Besides, the number of certain forwarded coded segments through this face is recorded in PIT until the router has forwarded a sufficient number of segments through this face under certain namespace. We will explain it in details as follows.

We design a PIT entry to count the received coded segments through the certain face. Each entry is appended with a value *count*, which represents the number of responded coded segments forwarded through this face under certain name. If a router receives an Interest from some face, the Interest will be recorded in the PIT entry with its name, incoming face and count if there are no suited contents in the router. And the *count* in PIT entry will be set as 0 initially. Otherwise, the router will respond this Interest with linear independent coded segments cached in its CS and refresh the value of *count* in PIT to the number of forwarded coded segments. If the number of responded coded segments *count* is less than the order of encoding matrix m (see IV-B) which is the basic number for users to decode the coded segments, it will be recorded in the CS and wait for other segments' arrival. Once the number of responded linear independent coded segments *count* for certain face reaches m, we can remove that PIT entry. For some upstream routers whose *count* in PIT entry may not reach m, the downstream router or user may already receive enough m coded segments. The upstream routers will keep receiving the coded segments until the CS collects enough segments, which is convenient to satisfy the following interests from other routers. Or the PIT entry in the upstream routers will finally be deleted due to timeout.

D. Decoding Coded Segments

After having received at least m linearly independent coded segments for content C, e.g., P_1, P_2, $\ldots$, P_m, the user can decode them to restore the matrix Z utilizing the corresponding encoding coefficient vectors Q_1, Q_2, $\ldots$, Q_m.

We assume that the received encoding coefficient vectors are $Q_i = (\alpha_{i1}, \beta_{i2}, \cdots \delta_{im})^T$ for $1 \le i \le m$. Then P_i can be depicted as $P_i = \alpha_{i1} \cdot Z_1 + \beta_{i2} \cdot Z_2 + \cdots + \delta_{im} \cdot Z_m$. We have $\begin{cases} P_1 = \alpha_{11} \cdot Z_1 + \beta_{12} \cdot Z_2 + \cdots + \delta_{1m} \cdot Z_m, \\ P_2 = \alpha_{21} \cdot Z_1 + \beta_{22} \cdot Z_2 + \cdots + \delta_{2m} \cdot Z_m, \\ \ldots\ldots \\ P_m = \alpha_{m1} \cdot Z_1 + \beta_{m2} \cdot Z_2 + \cdots + \delta_{mm} \cdot Z_m, \end{cases}$. Finally, these m linearly independent equations can help us to restore m elements in Z by using linear elimination.

After having obtained Z, the user will divide Z into two components, R' and R. The encrypted vector in R can be decrypted using the secret authorization key k. With R, R' can be decoded to obtain M'. Then a reverse operation of AONT can be performed on M' to restore the content C.

The scheme only encrypts one vector of the encoding matrix, significantly reducing computation overhead compared to simply encrypting the entire content [7]–[9]. AONT and encryption of a vector together ensure that an illegitimate user is not able to obtain the original content C.

E. Optimizing ACET

In the aforementioned design, we encrypt the entire vector R_m, but it seems unnecessary. Instead, we can only encrypt a portion of elements in vector R_m, which can also ensure that the adversary, without having access to key k, will not be able to obtain all the AONT-transformed content, and hence will not be able to learn anything about the original content.

We assume that R_m includes m elements, i.e., $R_m = (r_{m,1},\ r_{m,2},\ \ldots,\ r_{m,m-1},\ r_{m,m})$. The finite field of the network coding is $\mathbb{F}_q$, where q is a prime power 2^p. Rather than encrypt all the elements in R_m, we encrypt the first h elements.[2] Assuming the key distribution algorithm is secure (Sec. III), the attacker is not able to obtain the key and can only decrypt the encrypted elements by brute force. Considering that each element is chosen uniformly at random from $\mathbb{F}_q$, the successful probability for brute force is $\frac{1}{2^{hp}}$. For example, when p is 32, the number of encrypted elements can be only 4 to achieve 128-bit security.

V. A Timestamp-Based Revocation Scheme

A comprehensive access control design should allow dynamics of access privilege, e.g., existing privilege can be revoked, and new privilege can be granted. We therefore further extend ACET to support efficient revocation of access. In general, by changing the entire network encoding matrix and re-encoding the original content, a user can be revoked if we keep the new encoding matrix unknown to him/her. This, however, results in significant overhead in both computation and communication. Having observed that the original content is protected by confidentiality-enhanced network coding, by changing a portion of the encoding matrix (i.e., one vector) and keeping this portion secret, the revoked user will no longer be able to have access to "the entire" encoding matrix, and will no longer be able to decode and obtain the entire (i.e., *all*) AONT-transformed segments. Thanks to the all-or-nothing property of AONT, the revoked user will not be able to decode the AONT-transformed segments, and hence can learn *nothing* about the original content. In addition, since only one vector of the encoding matrix needs to be updated, most resulting network coded segments will remain the same, and only one resulting network coded segment will need to be changed. In other words, we only need to replace this out-of-date segment in each router's cache, incurring significant less overhead. Our revocation design mainly consists of three steps: 1) the data publisher changes a portion of the network coding matrix and re-processes the network coded segments; 2) each router eliminates the expired network coded segments from local cache ; and 3) the data publisher distributes the replacing segments to the network.

A. The Data Publisher Re-Processes Network Coded Segments

To revoke a user's access privilege on certain content (i.e., the privilege is expired), the content publisher changes one vector of the corresponding encoding matrix, and uses a new authorization key k' to encrypt this vector (or a portion of it according to the optimization in Sec IV-E).

The publisher will produce the updated network coded segments and distribute them to the network. Legitimate users can request and decrypt the updated segments by requesting the new authorization key k'.

A data publisher produces network coded segments using the confidentiality-enhanced network coding by multiplying the original content with the linear AONT matrix M and the encoding matrix R as follows (Sec. IV-B): $R' = RMC =$

$$\begin{pmatrix} R_1 \\ R_2 \\ \cdots \\ R_{m-1} \\ R_m \end{pmatrix} \begin{pmatrix} 1\,0 \cdots 0\,1 \\ 0\,1 \cdots 0\,1 \\ \vdots\,\vdots\,\ddots\,\vdots\,\vdots \\ 0\,0 \cdots 1\,1 \\ 1\,1 \cdots 1\,\lambda \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ \cdots \\ C_{m-1} \\ C_m \end{pmatrix} = \begin{pmatrix} R'_1 \\ R'_2 \\ \cdots \\ R'_{m-1} \\ R'_m \end{pmatrix},$$

where C_i and R'_i are vectors of n elements, and $1 \leq i \leq m$.

When revoking a user, the publisher changes one vector R_m of the encoding matrix R to a new one R_m^{new} and encrypts R_m^{new} using the new key k'. Based on the updated encoding matrix, the publisher will obtain: $R'' = R_{new}MC =$

$$\begin{pmatrix} R_1 \\ R_2 \\ \cdots \\ R_{m-1} \\ R_m^{new} \end{pmatrix} \begin{pmatrix} 1\,0 \cdots 0\,1 \\ 0\,1 \cdots 0\,1 \\ \vdots\,\vdots\,\ddots\,\vdots\,\vdots \\ 0\,0 \cdots 1\,1 \\ 1\,1 \cdots 1\,\lambda \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ \cdots \\ C_{m-1} \\ C_m \end{pmatrix} = \begin{pmatrix} R'_1 \\ R'_2 \\ \cdots \\ R'_{m-1} \\ R''_m \end{pmatrix}.$$

By comparing R' and R'', we observe that updating vector R_m to R_m^{new} only introduces a slight change to the resulting network coded segments, i.e., only the resulting segment R'_m will be changed, and all the other resulting segments R'_1, R'_2, $\cdots$, R'_{m-1} will all remain the same. Therefore, after R_m has been updated by the producer for revocation purpose, most network coded segments cached in routers can be reused, and only R'_m needs to be updated. Specifically, the out-of-date segment R'_m will be evicted from the cache in each router, and the new segment R''_m will be distributed to the network and incorporated into the cached segments in each router. Let Z_i be a concatenation of both R'_i and the corresponding vector R_i in the encoding matrix (Sec. IV-B), where $1 \leq i \leq m$ and R_m is encrypted with key k. Let Z'_m be a concatenation of R''_m and the corresponding vector R_m^{new} (encrypted with a new key k').

Two issues need to be addressed: 1) how to ensure the out-of-date Z_m can be evicted from the cache of each router, and 2) how to propagate the new Z'_m to the network. To help routers eliminate the out-of-date segments in the

[2] There is no difference in terms of security if we encrypt another subset of h elements in R_m.

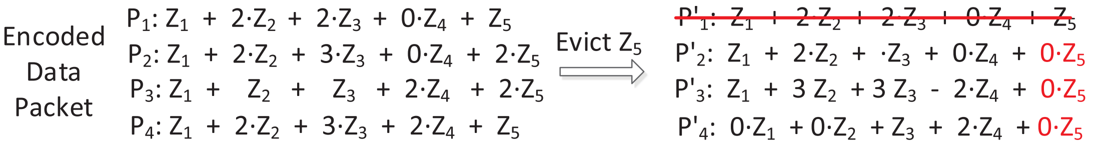

Encoded Data Packet
P_1: $Z_1 + 2\cdot Z_2 + 2\cdot Z_3 + 0\cdot Z_4 + Z_5$
P_2: $Z_1 + 2\cdot Z_2 + 3\cdot Z_3 + 0\cdot Z_4 + 2\cdot Z_5$
P_3: $Z_1 + Z_2 + Z_3 + 2\cdot Z_4 + 2\cdot Z_5$
P_4: $Z_1 + 2\cdot Z_2 + 3\cdot Z_3 + 2\cdot Z_4 + Z_5$
Evict Z_5
~~P'_1: $Z_1 + 2\cdot Z_2 + 2\cdot Z_3 + 0\cdot Z_4 + Z_5$~~
P'_2: $Z_1 + 2\cdot Z_2 + \cdot Z_3 + 0\cdot Z_4 + 0\cdot Z_5$
P'_3: $Z_1 + 3\ Z_2 + 3\ Z_3 - 2\cdot Z_4 + 0\cdot Z_5$
P'_4: $0\cdot Z_1 + 0\cdot Z_2 + Z_3 + 2\cdot Z_4 + 0\cdot Z_5$

Fig. 4. An example of evicting expired segments from cache.

cache, the publisher attaches a timestamp[3] with Z_m which indicates its lifetime. Once the timestamp is expired, Z_m will be evicted from the cache. We will further describe how to eliminate the expired Z_m from the cache in each router (Sec. V-B) by taking advantage of Gaussian elimination and how to propagate the new Z'_m to the network (Sec. V-C).

B. Routers Eliminate Out-of-Date Segments

At each router, network coded segments sent by the data publisher will be randomly combined to generate new coded segments before they are forwarded. To remove the encrypted segment Z_m sent by the publisher from the cache of a router, an immediate solution could be, the data publisher re-publishes a new set of coded segments. Each intermediate router clears its cache, and randomly combines the newly received coded segments to generate a new set of coded segments locally. This straightforward solution, however, is expensive in terms of both computation and communication. Due to linear property of network coding, we can use Gaussian elimination to remove the encrypted segment Z_m from the coded segments, which can significantly avoid unnecessary communication.

Assume that at an intermediate router, there are k coded segments, each is generated by randomly combining the original coded segments (Z_1, Z_2, ..., Z_m in Sec. IV-B) sent by the data publisher, and each time to revoke a user, we will update the segment Z_m, i.e., the old Z_m will be evicted from the router cache, and the new Z'_m will be added to the router cache. The coded segments cached in the routers are shown as follows ($\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m, \ldots, \delta_1, \ldots, \delta_m$ are all coefficients for random network coding):

$$\begin{cases} P_1 = \alpha_1 \cdot Z_1 + \alpha_2 \cdot Z_2 + \cdots + \alpha_m \cdot Z_m, \\ P_2 = \beta_1 \cdot Z_1 + \beta_2 \cdot Z_2 + \cdots + \beta_m \cdot Z_m, \\ \ldots\ldots \\ P_k = \delta_1 \cdot Z_1 + \delta_2 \cdot Z_2 + \cdots + \delta_m \cdot Z_m, \end{cases} \tag{1}$$

To help evict segment Z_m, we build the segment Z_m with an expiration timestamp T_{Thres} which records the time when it is expired. The data publisher will maintain the T_{Thres} value for each encrypted content segment Z_m and also maintain the subscription time for each authorized user. T_{Thres} indicates when the next user will expire and is determined by the most recent expiration time. Since the revocation scheme is based on subscription. Once a user is revoked, Z_m will be revoked, and the new Z'_m encrypted by the new authorization key with the updated T_{Thres} will be distributed to the network.

When the segment's lifetime exceeds the T_{Thres}, by checking the T_{Thres} value in the Z_m segment, the router where the cached segment is located will evict the outdated encrypted segment Z_m out of cache. Since the data publisher maintains the expiration time T_{Thres} for each encrypted segment and the subscription time for each authorized user. When some uses are revoked, the data publisher can respond the latest Interest with the new updated segment Z'_m encrypted by the new authorization key with the updated T_{Thres} to the network. In this way, the other segments except Z_m cached in the routers can be reused and the data publisher only need to resend the updated Z'_m to the network.

A concrete example where $m = 5$ (see Figure 4) is provided to show how we use Gaussian elimination to eliminate Z_m from each router's cache. Upon timeout (i.e., the lifetime of segment Z_5 exceeds the T_{Thres}), Z_5 is expired and should be evicted. By computing $P_4 - P_1$, we can remove Z_5 from P_4, obtaining P'_4; by computing $2 * P_1 - P_2$, we can remove Z_5 from P_2, obtaining P'_2; similarly, we can obtain P'_3. The newly created coded segments P'_2, P'_3, P'_4 only contain the unexpired segments Z_1, Z_2, Z_3, Z_4 which can be reused. All the old segments P_1, P_2, P_3 and P_4 will be removed from the cache.

C. The Data Publisher Propagates the Updated Segments

After the out-of-date segment Z_m has been evicted from the cache of each router, when a user sends an Interest to the network to request the data, the Interest will eventually reach the data publisher since the cached segments in routers are not enough for decoding the data. Once having received the Interest, the publisher will distribute the updated segment Z'_m to the network and each router in the path will apply random network coding to combine Z'_m to the cached segments under the same namespace. An authorized user can obtain the new key, and is hence able to decode and obtain the entire data, but a revoked user is not able to obtain the new key, and is not able to decode the data.

VI. Security Analysis and Discussion

A. Security Analysis

1) Security of ACET: The introduction of AONT ensures that by obtaining a portion of AONT-transformed content, the attacker is not able to derive any information about the original content. In the following, we first prove that the linear AONT used in ACET preserves the security property of AONT (Theorem 2). We then show that by encrypting one vector of the encoding matrix, the attacker cannot decode and obtain the entire AONT-transformed content, and hence

[3] Note that the timestamp here is different from the stale time/freshness period in the original NDN data packet [5]. The stale time/freshness period in the NDN data packet indicates lifetime of regular data packets, while the timestamp here specifically indicates lifetime of Z_m.

cannot obtain anything about the original content (Theorem 3). Finally, we show ACET can satisfy our security definition (Theorem 4).

Theorem 2: The linear $(n, q) - AONT$ used in ACET preserves the security property of AONT.

Proof: Given a linear $(n, q) - AONT$, M is generated using the method introduced in Sec. II-B; the input vector $V = [v_1\ v_2\ \cdots\ v_n]^T$, and the resulting output vector $U = [u_1\ u_2\ \cdots\ u_n]^T$ is computed as: $U = M \cdot V$. We can obtain:

$$\begin{cases} u_i = v_i + v_n, for\ i = 1, 2, \cdots, n-1 \\ u_n = v_1 + v_2 + \cdots + \lambda v_n \end{cases} \tag{2}$$

Therefore, by knowing the output vector $U = [u_1\ u_2\ \cdots\ u_n]^T$, we can compute the input vector $V = [v_1\ v_2\ \cdots\ v_n]^T$ as follows:

$$\begin{cases} v_i = u_i - \frac{1}{n-1-\lambda}\left(u_1 + u_2 + \cdots + u_{n-1} - u_n\right), \\ \quad for\ 1 \le i \le n-1 \\ v_n = \frac{1}{n-1-\lambda}\left(u_1 + u_2 + \cdots + u_{n-1} - u_n\right) \end{cases} \tag{3}$$

We can see that in order to obtain each of the v_i (where $1 \le i \le n$), all the values in the output vector U must be obtained. Otherwise, we cannot compute any of v_i. To conclude, this linear $(n, q) - AONT$ satisfies the all-or-nothing property. □

Theorem 3: By encrypting one vector R_m of the encoding matrix, the attacker cannot decode to obtain all the information about the AONT-transformed content M', and thus cannot obtain the original content C.

Proof: To obtain the original content C, the attacker must obtain all the information about the AONT-transformed content M' due to the all-or-nothing property. Since $R' = RM'$, therefore $M' = R^{-1}R'$. In other words, to obtain the entire M', the attacker must know the entire R^{-1}, which is the inverse matrix of the encoding matrix R. In the following, we show that by encrypting one vector R_m of the encoding matrix R, the attacker will not be able to learn the entire R^{-1} by contradiction.

Let us suppose that the adversary can learn the entire R^{-1}. Therefore, the inverse of matrix R^{-1} is unique and can be computed as: $R = (R^{-1})^{-1} = \frac{(R^{-1})^*}{|R^{-1}|}$, where $(R^{-1})^*$ is the adjugate matrix of R^{-1} and $|R^{-1}|$ is the determinant of R^{-1}. Since the entire R^{-1} is known, both $(R^{-1})^*$ and $|R^{-1}|$ can be uniquely computed, therefore, the entire R can be uniquely computed, and the vector R_m is known to the adversary. This contradicts that R_m is encrypted and unknown to the adversary. Therefore, the adversary cannot learn the entire R^{-1}.

Without being able to learn the entire R^{-1}, the attacker cannot obtain the entire AONT-transformed content M', and hence cannot learn anything about the original content C due to the property of AONT. □

Theorem 4: The ACET scheme can satisfy our security definition for a large enough q.

Proof: We show ACET can satisfy the security conditions in Sec. III: 1) $p_1 \to 0$; and 2) $p_2 \to 1$.

1) $p_1 \to 0$: According to Theorem 3, by encrypting one vector R_m of the encoding matrix, the attacker cannot learn anything about the original content C. In this sense, p_1 is equivalent to the probability that the attacker can successfully decrypt R_m without knowing the key. Assuming the key distribution algorithm is secure (Sec. III), the attacker can only find out R_m by brute-force guessing, and the successful probability is $\frac{1}{q^m}$ considering that each of the m elements in R_m is chosen uniformly at random from $\mathbb{F}_q$. This probability is negligibly small when q and m are sufficiently large. For example, if $m = 10$ [21], $q = 2^{32}$ will result in a probability of $\frac{1}{2^{320}}$, which is approximately 10^{-32}. For the optimization which only encrypts a portion of the elements in R_m, the successful probability of the adversary is also negligibly small (Sec. IV-E) if q is sufficiently large. We conclude that $p_1 \to 0$.

2) $p_2 \to 1$: After having received enough number of network coded segments, a legitimate user is able to decode them, obtaining Z_1, Z_2, ..., Z_m. After having obtained the entire data Z, the user can divide Z into two components, R' and R. The encrypted vector in R can be decrypted using the authorization key. With the entire R in plaintext, all the AONT-transformed segments can be computed, and then, the original content can be decoded from the AONT-transformed segments. Therefore, for the legitimate user which can obtain the authorization key, $p_2 \to 1$. □

2) Security of Revocation: ACET can support revocation of users once they are expired. This is because, the segment Z_m is attached with a timestamp, and once this timestamp is expired, Z_m will be evicted from each router's cache following Gaussian elimination, and the updated segment Z'_m will be propagated to the network afterwards. Note that: 1) A revoked user can always store the original content locally before he/she is revoked, which cannot be addressed by any access control scheme, but once the content is updated by the publisher, the revoked user will no longer decode the updated content. 2) Z_m will expire once T_{Thres} is reached, at which the routers that cache segments containing Z_m will evict Z_m, replacing it with Z'_m. A revoked user cannot have access to the new key k', which is needed to decrypt R_m^{new}. Without having access to R_m^{new}, the revoked user can no longer decode and obtain all the AONT-transformed content, and thanks to the nice property of AONT, he/she will not be able to obtain anything about the original content.

B. Discussion

Thanks for your comments. In our revocation scheme, when some users are revoked, the other legitimate users will need to request the new key. This scheme is feasible when the revocation frequency is not high. Whereas, when the revocation frequency is high, we can propose a key chain-based extension, which pre-assigns keys in case that the present key is revoked. Since our scheme is a subscription-based revocation scheme, the data publisher knows when the subscription of a legitimate user expires. When the legitimate user requests authorization keys from the data publisher, the data publisher will respond with a key chain covering the minimum subscription period of users, for example, one month, so the number of keys in this key chain is limited. In this way, the key requesting frequency can be reduced.

VII. Experimental Evaluation

A. Implementation

We used ndnSIM [22] for our simulations. All of our simulations were performed on a local machine, equipped with an Intel Core i3 3.3G CPU and 5.7G RAM, running Ubuntu 14.04. In our simulations, the size of content store was chosen as 1000 and the number of content items was 10,000. In addition, to simulate the background traffic, each legitimate consumer and each legitimate user sends 10 regular Interests every second. The maximum PIT size was set as 15,000. The duration for Interests was 0-60s. The encryption algorithm is instantiated using AES, which is an asymmetric encryption method.

We deployed Rocketfuel network topology and a tree topology respectively.

(1) Rocketfuel network topology: As part of the European Ebone (an ISP in Europe) [23] network, the topology extracted from Ebone is representative, including typical ones like mesh and tree. We use this topology to evaluate the performance of the scheme under the perspective of the whole network. To simulate the real network application, we used 1 data publisher and 9 users.

(2) Tree topology: We utilize a 4-layer tree topology to test the performance of the scheme under the edge network. Similarly, we deployed 1 data publisher and 5 users, and the users are arranged into different layers of the network.

For comparison, we also implemented a baseline scheme, a group signature-based access control scheme [24] (“Group” for short), and a tag-based access control scheme Tagtic [25].

(1) Baseline: a content-encryption-based access control scheme, in which the user pre-fetches the corresponding authorization key before sending an Interest, and the data publisher encrypts the content which can only be decrypted using the authorization key.

(2) Group [24]: a group signature-based access control mechanism which utilizes the group signature to verify authenticity of Interests sent by users at the edge routers, while protecting the privacy of the users.

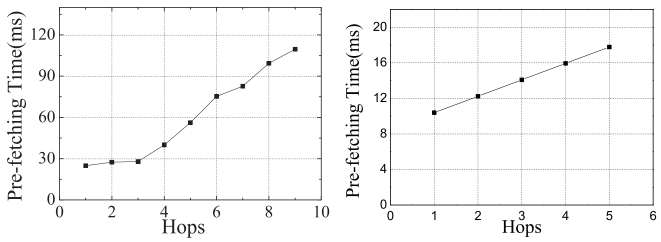
(3) Tagtic [25]: a tag-based access control scheme which authenticates Interest packets at edge and content routers and makes use of bloom filters to distribute the authentication results.

B. Evaluation

We evaluate performance of ACET in both the initial and the transmission process. In the initial process, we assess overhead resulted from obtaining the authorization key. In the transmission process, we assess both transmission delay (i.e., the time between an Interest request is sent and the corresponding content is received) and throughput.

Evaluating the initial process. During initialization, each valid user retrieves the authorization key from the publisher.

(1) Under the Rocketfuel topology, the delay of retrieving the authorization key in ACET is shown in Figure 5(a), in which we calculate the average delay during a period of 100 seconds under different hops away from the data publisher. We observe that: 1) Under the Rocketfuel topology, the average delay varies from 20ms (approximately) to 110 ms (approximately) when the hop count varies from 1 to 9, and such a delay is small. 2) As the number of hops away from the publisher increases, the corresponding delay increases almost linearly. This is because, when a user is more distant away from the data publisher, it takes more time for the user to deliver the request Interest and receive the corresponding authorization key. We can also get that although the distance varies, the transmission delay can always stay in a low level. Since the complexity of key generation is fixed, the initial process can only be impacted by the distance to the data publisher.



(a) Pre-fetching time under Rocketfuel topology

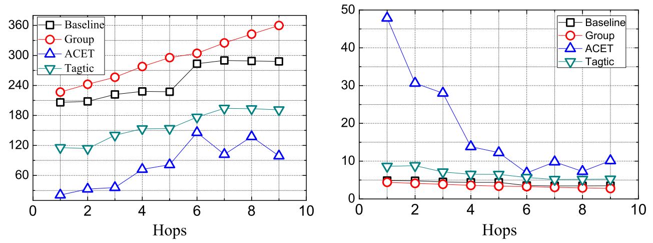
(b) Pre-fetching time under tree topology

Fig. 5. Delay of retrieving the authorization key.

(2) Under a tree topology, the delay of initial retrieving the authorization keys is shown in Figure 5(b). We also calculate the average delay during a period of 100 seconds under different hops away from the data publisher. We observe that: 1) under a tree topology, the average delay varies from 10ms (approximately) to 18 ms (approximately) when the hop count varies from 1 to 5, which is also small. 2) As the hop away from the data publisher increases, the initial process delay also increases. This is similar to that under the Rocketfuel topology. It will cost a more distant user more time to obtain the authorization key. With the distance to producer increases, the initial transmission delay increases linearly.

Evaluating the transmission process. We calculate the average transmission delay for a period of 100 seconds under different hops away from the publisher.

(1) Under the Rocketfuel topology, the transmission delay is shown in Figure 6(a). We also calculate the throughput (Figure 6(b)). We notice that when the hop increases from 6 to 9, the curve of time delay about ACET in the figure has fluctuations. As is shown in Figure 7, the local topologies have different features in the Rocketfuel topology. When there are more paths for a user to connect to the data publisher (indicated as DP in Figure 7), the transmission efficiency will be higher in a network coding-based network, because there are more chances to get coded segments. For example, as illustrated in Figure 7, the users who are 7 and 9 hops away from the data publisher spend less time for transmission than the users who are 6 and 8 hops away, for the reason that there are more paths for the coded segments to be transmitted and encoded to the users who are 7 and 9 hops away. Therefore, the transmission delay may have fluctuations when the hops away from data publisher vary from 6 to 9.



(a) Transmission latency

(b) Throughput

Fig. 6. Performance in Rocketfuel Topology.

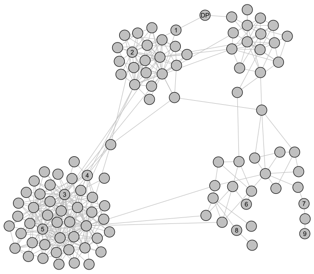
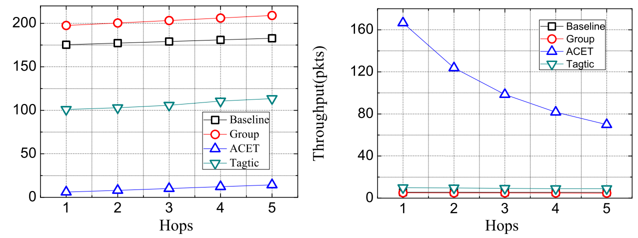


Fig. 7. Rocketfel Topology.

We also observe that our ACET outperforms the other schemes in terms of both the transmission delay and the throughput. This is because: the baseline scheme needs to encrypt the entire content in the publisher and decrypt it in the user, which clearly introduces large overhead; the group signature-based access control scheme (i.e., Group) needs to sign and verify the Interest packet at the edge routers based on the group signature and, since the size of Interest packet is large, the extra overhead is also large; Tagtic needs to validate the Interest packet at the edge routers, distribute the verification result to the remaining network, and the other routers will add the verification result to their local bloom filters for probabilistic verification, which also introduces a lot of extra overhead in the network; our ACET only needs to encrypt a small vector (or a portion of the elements in the vector) in the publisher and decrypt this small vector (or a portion of elements in the vector) in the user which is extremely lightweight.

(2) Under a 4-layer tree topology, the transmission delay and the throughput is shown in Figure 8(a) and Figure8(b). We can observe that under a tree topology, ACET scheme also outperforms the other schemes in transmission delay and throughput. In Figure 8(a), we show the transmission delay from user's sending an Interest request to user's obtaining the corresponding content. It can be observed that with the hops away from the producers increase, the transmission delay rises, but the transmission delay of ACET stays in a low level which is under 50ms. While the transmission delays of other schemes are higher. As is shown in Figure 8(b), the throughput of ACET decreases with the distance increases, falls about 70 pkts when the distance between user and producer is 5, but also higher other schemes. The experimental results of the tree topology are similar to the Rocketfuel topology, but the difference is that under a tree topology, the transmission delay and throughput are quite regular and stable for the reason that the tree topology is more hierarchical and clear.



(a) Transmission latency

(b) Throughput

Fig. 8. Performance in Tree Topology.

VIII. Related Work

Since ICN users more likely obtain desired content from cache of network nodes rather than from original data publishers, access of content can not be enforced by simply relying on traditional access control policies. Instead, a content-based approach is expected. In the following, we summarize the existing access control solutions for ICN in various categories.

Encryption-based access control. As a seminal work, Jacobson [26] proposed an encryption-based access control scheme to protect the content. Some existing encryption-based approaches for access control in NDN can be categorized into symmetric/asymmetric encryption-based [7], attribute and identity-based [8], [9], and session-based. For example, Misra *et al.* [27] use the (n, t)-Shamir's secret sharing based broadcast encryption to enforce Access Control. The framework's strength is that it needs neither additional authorization entity nor extra computational overhead at the routers. Li *et al.* [8] propose an attribute-based encryption access control scheme. In their scheme, a trusted third party defines and manages the subject and object attributes by creating attribute ontology. To publish content, the publisher generates a random symmetric key with which it encrypts the content and generates an AC policy from the attributes defined by the trusted third party. The problem with the attribute-based systems is that the client revocation is not considered and the computation complexity is high. Zhu *et al.* [10] propose an edge re-encryption-based access control scheme, in which producer and edge routers will encrypt the content key, reducing the key retrieving delay to some extend. But it still needs to encrypt the whole content and the user privacy will be exposed to the edge routers.

All of aforementioned designs however, rely on expensive encryption according to their trusted regulation and management methods. Meanwhile, those session-based works need several rounds of sessions to establish the relationship and share the key, which is also expensive.

Authentication-based access control. A class of authentication-based access control mechanisms [24], [25] for NDN delegates the access control enforcement to routers. Xue *et al.* [24] use group signature and hash chain to anonymously authenticate requests at the edge router, which achieves access control at the very beginning. Although Xue *et al.* in their work make use of group signature to anonymously verify the authentication of Interests, this process can introduce redundant validation time. Tourani *et al.* [25] propose an access control mechanism for the ICN wireless edge called TACTIC, in which providers delegate access control enforcement to the edge routers and content routers, allowing cache utilization and promoting content availability with negligible computation and communication overhead. They both utilize the edge routers to perform Interest verification to enforce access control at the initial stage of transmission, but the authentication of Interests will expose the privacy of users to the routers.

Others. Li *et al.* [28] design a lightweight digital signature and access control scheme for ICN. Upon an entity's request for a token, the provider encrypts the token based on the requesters' access level and creates hash-based signatures. And the client verifies signature by generating Merkle hash tree using the retrieved content and the new token. Also, the client revocation has not been considered in this article. Ghali *et al.* [29] tackled the access control problem using an Interest-based model, in contrast to popular encryption-based approaches. They proposed an encryption-based and hash-based name obfuscation to prevent unauthorized clients from obtaining the content name. Although their design does not need to encrypt the data, the name of content is more complex and becomes opaque, which will make the transmission of contents more sophisticated.

IX. Conclusion

In this work, we propose an efficient and secure access control scheme specifically for information-centric edge networking, which utilizes named data networking (NDN) to achieve in-network caching. Deploying in-network caching will facilitate computing and caching abilities in the network, but it will also create new security issues, such as unauthorized access control to the data which are already cached in the network and out of the producer's control. By introducing the confidentiality-enhanced network coding, which combines linear all or nothing transform and encryption with network coding, we successfully come up with such an access control design with efficient revocation support at edge networks. In our design, we can enforce access control on the cached content by encrypting a small portion of the encoding matrix, which significantly improves efficiency. We also realize revocation by only evicting the cached encrypted portion and reusing the unencrypted segments cached in the router. In the last, security analysis and experimental evaluation confirm that our design can ensure access control with an acceptable overhead.

References

[1] *5G Network Architecture—A High-Level Perspective*, Huawei, Shenzhen, China, 2016.

[2] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, Sep. 2016.

[3] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

[4] Y. Jin, Y. Wen, and C. Westphal, "Optimal transcoding and caching for adaptive streaming in media cloud: An analytical approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 12, pp. 1914–1925, Dec. 2015.

[5] L. Zhang *et al.*, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, vol. 157, p. 158, Oct. 2010.

[6] E. Mannes and C. Maziero, "Naming content on the network layer: A security analysis of the information-centric network model," *ACM Comput. Surveys*, vol. 52, no. 3, pp. 1–28, Jul. 2019.

[7] J. Kuriharay, E. Uzun, and C. A. Wood, "An encryption-based access control framework for content-centric networking," in *Proc. IFIP Netw. Conf. (IFIP Netw.)*, May 2015, pp. 1–9.

[8] 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. 2018.

[9] M. Ion, J. Zhang, and E. M. Schooler, "Toward content-centric privacy in ICN: Attribute-based encryption and routing," in *Proc. 3rd ACM SIGCOMM Workshop Inf.-Centric Netw. (ICN)*, 2013, pp. 513–514.

[10] Y. Zhu, R. Huang, Y. Tao, and X. Wang, "An edge re-encryption-based access control mechanism in NDN," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 6, p. e3564, Jun. 2019.

[11] Y. Wang, M. Xu, Z. Feng, Q. Li, and Q. Li, "Session-based access control in information-centric networks: Design and analyses," in *Proc. IEEE 33rd Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2014, pp. 1–8.

[12] R. L. Rivest, "All-or-nothing encryption and the package transform," in *Proc. Int. Workshop Fast Softw. Encryption*. Berlin, Germany: Springer, 1997, pp. 210–218.

[13] D. R. Stinson, "Something about all or nothing (transforms)," in *Designs, Codes and Cryptography*, vol. 22, no. 2. Boston, MA, USA: Kluwer, 2001, pp. 133–138.

[14] L. Zhang *et al.*, "Named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014.

[15] T. Koponen *et al.*, "A data-oriented (and beyond) network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 181–192, 2007.

[16] M. Ain *et al.*, "D2. 3–architecture definition, component descriptions, and requirements," in *Proc. 7th FP EU-Funded Project Deliverable (PSIRP)*, Feb. 2009, p. 11.

[17] D. Wu, Z. Xu, B. Chen, and Y. Zhang, "Towards access control for network coding-based named data networking," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[18] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[19] T. Ho *et al.*, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[20] J. Saltarin, E. Bourtsoulatze, N. Thomos, and T. Braun, "NetCodCCN: A network coding approach for content-centric networks," in *Proc. IEEE 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[21] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote data checking for network coding-based distributed storage systems," in *Proc. ACM Workshop Cloud Comput. Secur. Workshop (CCSW)*, 2010, pp. 31–42.

[22] A. Afanasyev *et al.*, "ndnSIM: NDN simulator for NS-3," Univ. California, Los Angeles, CA, USA, Tech. Rep. NDN-0005, 2012.

[23] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with Rocketfuel," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 133–145, Oct. 2002.

[24] K. Xue, X. Zhang, Q. Xia, D. S. L. Wei, H. Yue, and F. Wu, "SEAF: A secure, efficient and accountable access control framework for information centric networking," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 2213–2221.

[25] R. Tourani, R. Stubbs, and S. Misra, "TACTIC: Tag-based access ConTrol framework for the information-centric wireless edge networks," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 456–466.

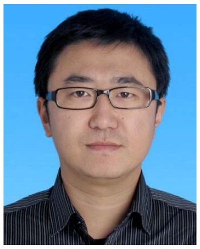
[26] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, 2009, pp. 1–12.

[27] 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 Trans. Dependable Secure Comput.*, vol. 16, no. 1, pp. 5–17, Jan. 2019.

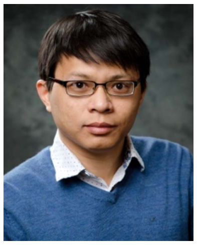
[28] Q. Li, X. Zhang, Q. Zheng, R. Sandhu, and X. Fu, “LIVE: Lightweight integrity verification and content access control for named data networking,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 308–320, Feb. 2015.

[29] C. Ghali, M. A. Schlosberg, G. Tsudik, and C. A. Wood, “Interest-based access control for content centric networks,” in *Proc. ACM Conf. Inf.-Centric Netw.*, 2015, pp. 147–156.

Danye Wu received the B.E. degree in communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2015. She is currently pursuing the Ph.D. degree with the Institute of Computing Technology, CAS, and the University of Chinese Academy of Sciences, Beijing, China. She was also a Visiting Ph.D. Student with the Department of Electrical and Computer Engineering, University of Houston, from 2019 to 2020. Her research interests include future Internet and network security.



Zhiwei Xu (Member, IEEE) received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2002, the M.S. degree from the Inner Mongolia University of Technology, Hohhot, China, in 2008, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2018. He is currently an Associate Professor and a M.S. Supervisor. His research interests include network performance analysis and the related mathematics problems.



Bo Chen (Member, IEEE) received the M.Eng. degree from the University of Chinese Academy of Sciences, China, in 2008, and the Ph.D. degree from the New Jersey Institute of Technology, USA, in 2014. He is currently an Assistant Professor with the Department of Computer Science, Michigan Technological University (MTU), where he also leads the MTU Security and Privacy (SnP) Lab. His current research interests include the areas of applied cryptography and data security. He has been actively solving various security and privacy problems in IoT, cloud computing, information-centric networking, and big data, and has authored/coauthored over 40 publications. He is a member of ACM.

Yujun Zhang (Member, IEEE) received the B.S. degree in computer science from Nankai University in 1999, and the Ph.D. degree in computer architecture from the University of Chinese Academy of Sciences in 2004. He is currently a Professor with the Institute of Computing Technology, CAS, and the University of Chinese Academy of Sciences. His research interests include intelligent networking and systems, network architecture, and network measurement and testing. He received the Technological Invention Award from China Computer Federation in 2013, and the Beijing Young Famous Teacher Award in 2019.

Zhu Han (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer with JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID, USA. He is currently a John and Rebecca Moores Professor with the Department of Electrical and Computer Engineering and with the Department of Computer Science, University of Houston, TX, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He was a Distinguished Lecturer of the IEEE Communications Society from 2015 to 2018, a Fellow of AAAS since 2019, and a Distinguished Member of ACM since 2019. He received the NSF CAREER Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for *Journal on Advances in Signal Processing* in 2015, the IEEE Leonard G. Abraham Prize in the field of Communications Systems (Best Paper Award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Since 2017, he is 1% Highly Cited Researcher according to Web of Science. He is also the Winner of 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation “for contributions to game theory and distributed management of autonomous communication networks.”