SCD2: Secure Content Delivery and Deduplication With Multiple Content Providers in Information Centric Networking

Kaiping Xue, *Senior Member, IEEE*, Peixuan He, Jiayu Yang, *Graduate Student Member, IEEE*, Qiudong Xia, and David S. L. Wei, *Senior Member, IEEE*

***Abstract*— As one of the promising next generation network architectures, information centric networking (ICN) is highly anticipated to improve the bandwidth usage of the Internet and reduce duplicate traffic. Since contents in ICN are disseminated in the whole network, ICN is much more vulnerable and the issue of how to deliver contents securely has been intensively discussed. However, the scalability of the existing schemes is limited. A scalable scheme is expected to be able to achieve fine-grained access control and at the same time also support multiple content providers scenario with efficient key management at user side. Besides, different content providers may publish some identical contents and these contents may be cached in the same intermediate routers, which causes high data redundancy and in turn exerts an adverse impact on the performance of ICN. In this paper, we propose a Secure Content Delivery and Deduplication scheme, called SCD2, to achieve secure and efficient fine-grained access control in ICN with multiple content providers. We first propose a scalable key-policy attribute-based encryption (SKP-ABE) to provide fine-grained access control and allow different attribute authorities to share some public attributes to simplify the key management. Furthermore, based on SKP-ABE, we design a simple but effective mechanism to conduct content deduplication. Finally, we implement a prototype of SCD2 to test its performance and compare it with some existing schemes. The results show that SCD2 has lower storage overhead, a higher degree of deduplication, and better retrieval efficiency.**

***Index Terms*— Information centric networking, access control, content deduplication, multiple content providers.**

I. Introduction

NOWADAYS media traffic has taken up the most Internet traffic and numerous redundant contents are transmitted over the Internet, which leads to low bandwidth utilization ratio and transmission efficiency. To mitigate these problems, information centric networking (ICN) [1]–[4] has been proposed for such content-centric pattern, where contents are cached in the intermediate routers and can be used to directly respond to the requests. In ICN, users can obtain content cached in the routers arbitrarily without the permission of content providers (CPs), which is detrimental to CPs' benefit. Therefore, to facilitate the development of ICN and ensure secure content delivery, many researchers have been working on the access control of ICN and proposed their solutions.

Existing schemes can be simply divided into three categories according to their access control methods: authentication-based schemes [5]–[10], encryption-based schemes [11]–[19], and hybrid schemes [20]–[22]. In authentication-based schemes, only users who pass the authentication conducted by intermediate routers, CPs, or extra security servers are able to obtain the contents for which they desire. But these schemes always cause frequent interactions and extra computation overhead on routers. Encryption-based schemes allow anyone to obtain the encrypted contents, but only legitimate users can decrypt them. Since there is no restriction on access to the contents, the network resources can be easily exhausted by interest flooding attacks [23], [24] launched by malicious users. To overcome the deficiencies of these two kinds of schemes, hybrid schemes are proposed, which combine authentication and encryption and provide two-layer access control. Nevertheless, all these aforementioned schemes have limited scalability, i.e., they can not achieve fine-grained access control nor support multiple content providers scenario with simple key management at the user side.

A feasible method to achieve a scalable access control in ICN is combining authentication with multi-authority attribute-based encryption (MA-ABE), where each CP acts as an attribute authority. There are two kinds of ABE: ciphertext-policy ABE (CP-ABE) [25] and key-policy ABE (KP-ABE) [26]. In CP-ABE, users are allowed to arbitrarily combine their attribute keys to satisfy the access structure embedded in the ciphertext for decryption. Thus, if a user who is the VIP of Netflix and a member of Youtube obtains keys corresponding to attributes "VIP", "Netflix", "member", and "Youtube" from these two CPs, he/she can combine the attributes "VIP" and "Youtube" to get VIP status of Youtube, which is obviously undesired. Therefore, although

Manuscript received November 7, 2019; revised February 5, 2021 and January 7, 2022; accepted February 20, 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor K. Ren. This work was supported in part by the National Natural Science Foundation of China under Grant 61972371 and in part by the Youth Innovation Promotion Association of the Chinese Academy of Sciences (CAS) under Grant Y202093. *(Corresponding author: Kaiping Xue.)*

Kaiping Xue, Jiayu Yang, and Qiudong Xia are with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (e-mail: kpxue@ustc.edu.cn).

Peixuan He is with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, Anhui 230027, China, and also with the Security Research Department, Bytedance Corporation, Beijing 100098, China.

David S. L. Wei is with the Department of Computer and Information Science, Fordham University, Bronx, NY 10458 USA.

Digital Object Identifier 10.1109/TNET.2022.3155110

different CPs may share some attributes like "VIP", different CPs must differentiate these attributes (e.g., assign attribute "Netflix_VIP") for security purposes, which actually increases the overhead to manage attribute keys obtained from these CPs. Though KP-ABE can restrict such arbitrary attribute key combinations by embedding an access structure in users' attribute keys, authorities in existing multi-authority KP-ABE schemes assign attribute keys that implicitly contain access structures entirely independently. Hence, they cannot achieve sharing some attributes among different CPs either and the key management complexity is still high. Thus, a novel ABE algorithm needs to be designed with efficient key management for multiple CPs scenario in ICN.

Besides, since ICN depends on in-network caching to provide better service, the quality of contents cached in the network exerts a direct impact on the quality of its service. Specifically, the more different useful contents it caches, the better service can be provided. However, in the real world, users may request the same contents from different CPs, which causes the cache space of intermediate routers to be occupied by duplicate contents. Considering the fact that the cache space of a router in ICN tends to be limited, the performance of ICN is extremely susceptible to duplicate data. Therefore, it is essential for us to design a content deduplication mechanism while maintaining secure data delivery in ICN.

In this paper, we propose SCD2, a Secure Content Delivery and Deduplication scheme for multiple CPs scenario in ICN with high scalability. The key to this scheme is that we propose Scalable Key-Policy Attribute-Based Encryption (SKP-ABE for short) where different attribute authorities (AAs) can share some public attributes. In order to let the new AA reuse a user's public attribute key, the user needs to submit the public key obtained from other AAs to the user's newly registered AA. Upon receiving the user's existing public attribute key, according to the received public attribute key and access policy, the new AA only needs to generate the key corresponding to the attribute that the user currently does not have. In SCD2, CPs act as both AAs and data owners in SKP-ABE to manage the key assignment and to encrypt the contents they publish for achieving fine-grained access control. Meanwhile, to thwart potential interest flooding attacks, edge routers will send a sample encrypted with the same attribute set as the contents that users request to let users decrypt and see whether they have permission to access the contents. Finally, depending on the feature of sharing attributes, the redundant contents from different CPs in a router can be conveniently eliminated by communicating with any CP.

Our contributions can be summarized as follows:

- We propose a variant scheme of KP-ABE, called SKP-ABE, in which some attributes can be commonly used by different AAs. In such a way, the overhead of key management on the user side can be largely reduced.
- We propose a secure content delivery scheme for multiple CPs scenarios in ICN. Specifically, we utilize SKP-ABE to achieve fine-grained access control and conduct a challenge-response process at edge routers to protect network resources from being exhausted by illegitimate requests.
- We design a novel content deduplication mechanism that is compatible with SKP-ABE, in which a POW process is carried out when CPs publish contents at edge routers to make sure the contents cached in the core network are indeed owned by these CPs and duplicate contents are eliminated by communicating with one of the CPs.
- We formally analyze the security features of our scheme and estimate the performance by comparing our scheme with some existing schemes in terms of storage, communication, and computation overhead. Besides, we implement our scheme in a simulated network to test the effectiveness of content deduplication and retrieval efficiency.

The rest of this paper is organized as follows. Section II first introduces the related work and Section III states the preliminaries. Section IV presents the system model, the threat assumption, and design goals. After stating the structure of SKP-ABE in Section V, the proposed scheme, SCD2, is given in Section VI. Then section VII and section VIII show the security and performance analysis, respectively. Finally we conclude this paper in Section IX.

II. Related Work

A. Access Control in ICN

At the beginning of designing ICN architecture, researchers have put the security framework into ICN [27]. However, the access control is achieved by public key infrastructure and each user needs to communicate with CPs for each content requested, which causes much overhead and is unfriendly to users.

Since then, more schemes have been proposed to provide better access control for secure content delivery. Li *et al.* [5] proposed a capability-based security enforcement scheme for ICN, in which users prove their identities through tokens obtained from CPs. AbdAllah *et al.* [6] proposed a decentralized access control protocol. In this protocol, routers can distinguish between legitimate users and illegitimate users through the random numbers assigned to users and routers. However, since the random numbers in routers are the same, this protocol is not secure enough once the random number is leaked. Fotiou *et al.* [7] proposed to transmit users' requests to CPs for authentication, but it is inefficient due to the frequent authentication on the CP side. Zhu *et al.* [10] proposed a time-based content access control mechanism, which delegates authentication tasks from CPs to intermediate content distribution servers that act as proxies to ensure access management is done securely and effectively. Tourani *et al.* [28] designed a tag-based access control framework that bounds the users' access rights with tags received from content providers. Routers in ICN determine whether they forward contents to users according to the validity of the tags embedded in the requests. Besides, access control in ICN can be achieved by utilizing some advanced encryption algorithm without authentication like attribute-based encryption [13], proxy re-encryption [14], broadcast encryption [12] and so

forth. However, these schemes are vulnerable to interest flooding attacks due to open access. To overcome these defects, Xue *et al.* [21] proposed a two-layer access control scheme where at edge routers an authentication based on group signature is conducted while contents are encrypted by broadcasting encryption to keep confidentiality. Although there have been many proposed schemes, none of them is able to run efficiently in the scenario where there are quantities of CPs and the access policies are diverse. Wu *et al.* [17] proposed a fine-grained privacy protection scheme, called CHTDS. It uses a fixed-length data partitioning idea and a CP-ABE encryption algorithm to encrypt big data blocks and implement access control. FGAC-NDN [18] is also a fine-grained access control scheme, which supports data confidentiality, potential receivers, and mobility. However, it assumes that online shop and consumers do not collude with each other.

B. Attribute-Based Encryption

Since the attribute-based encryption is proposed, it has been regarded as one of the promising techniques to provide fine-grained access control and has been widely used in cloud storage systems [29]–[31]. Goyal *et al.* [26] first put forward KP-ABE and use it to achieve fine-grained access control. After that, Chase [32] extended basic KP-ABE to multi-authority KP-ABE, in which a central authority (CA) is introduced. Then, Chase and Chow [33] further improved the privacy and security of their scheme, and eliminate CA by negotiating secret parameters between each two of the attribute authorities (AAs). Li *et al.* [34] proposed a multiple-authority KP-ABE based on linear secret sharing scheme. Dougherty *et al.* [35] also proposed a multiple-authority KP-ABE access control framework for pervasive edge computing. Their scheme eliminates the need for the "always-on" authentication servers in the cloud by delegating the authentication tasks to semi-trusted edges. However, since the key assignment of different AAs is completely independent, users will get different keys corresponding to the same attributes, which largely increases the overhead of managing these keys.

Closely following the proposal of multi-authority KP-ABE, in 2008, Muller *et al.* [36] proposed the first multi-authority CP-ABE with many attribute authorities and a master authority responsible for key management. Then, Yang *et al.* [37] proposed a multi-authority CP-ABE scheme, where a central authority is only involved in system initialization. After that, a number of researchers propose their multi-authority CP-ABE schemes to address policy update [38], privacy-preserving [39], single-point bottleneck [40], [41], etc.

Nevertheless, since users are allowed to arbitrarily combine the attributes they have in CP-ABE, users can combine the attribute sets that are beyond their access privileges, once multiple content providers share some attributes. Sultan *et al.* [42] proposed a novel role-based encryption (RBE) access control scheme, which takes into account the hierarchies in roles. The scheme allows the consumers who pay a higher level of subscription to access contents both in the higher and lower part of the hierarchy using their decryption keys. Based on RBE, Sultan *et al.* [43] further proposed secure access and accountability framework for provisioning services. It can enable authenticated consumers to obtain the keys that decrypt required content and mitigate DoS attacks by an anonymous signature-based authentication scheme. However, it is unable to solve the problem of content duplication.

C. Content Deduplication

Content deduplication is a widely used technique to reduce storage space and upload bandwidth in the cloud storage system. To make it possible to conduct deduplication over encrypted contents, Bellare *et al.* [44] proposed message-locked encryption (MLE), in which message is encrypted by a message-derived key and identical message plaintexts have the same tags generated from the ciphertexts. After that, a lot of schemes were proposed to enhance the security and availability of MLE. Keelveedhi *et al.* [45] then introduced a key server to generate the file tag to protect data confidentiality. Li *et al.* [46] proposed a secure deduplication scheme with efficient and reliable key management, where users can securely distribute the convergent key shared across multiple servers. Jiang *et al.* [47] provided a deduplication solution with a stronger security guarantee, where the tag is fully randomized instead of being derived deterministically from the plaintext. But MLE cannot flexibly control data access and achieve fine-grained access control. Therefore, some data deduplication schemes based on attribute-based encryption [48], [49] are proposed. However, [48] doesn't consider the Proof of Ownership process, which is essential for data deduplication and directly related to the security of a scheme. Also, [49] needs to execute pairing operation for each tag every time when cloud conducts data deduplication, of which the overhead is unacceptable.

III. Preliminaries

A. Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups with the same prime order p, and g be the generator of $\mathbb{G}$. A bilinear map is defined as a mapping $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ that has the following three properties:

1) *Bilinearity*: For all $a, b \in \mathbb{Z}_p$ and $u, v \in \mathbb{G}$, we have $e(u^a, b^b) = e(u, v)^{ab}$;
2) *Non-degeneracy*: $e(g, g) \neq 1$;
3) *Computability*: For all $g_1, g_2 \in \mathbb{G}$, there is an efficient algorithm to compute the mapping $e(g_1, g_2)$.

Our scheme works on the bilinear pairings with Decisional Bilinear Diffie-Hellman (DBDH) assumption [50] as follows:

Definition 1: DBDH Assumption: Given a 4-tuple $(g^a, g^b, g^c, e(g, g)^z)$ *as input, it is hard to distinguish whether* $z = abc$ *or* z *is randomly chosen from* $\mathbb{Z}_p$*. Specifically, an adversary* $\mathcal{A}$ *has a non-negligible advantage* ϵ *to solve DBDH if*

$$\left| Pr\Big[\mathcal{A}(g^a, g^b, g^c, e(g, g)^{abc}) = (g^a, g^b, g^c, e(g, g)^{abc})\Big] - Pr\Big[\mathcal{A}(g^a, g^b, g^c, \eta) = (g^a, g^b, g^c, e(g, g)^{abc})\Big] \right| \geq \epsilon,$$

where η *is random in* $\mathbb{G}_T$.

B. Key-Policy Attribute-Based Encryption (KP-ABE)

KP-ABE [26] is a cryptographic prototype that is widely used for secure access control in distributed systems. In a KP-ABE-based framework, users' secret keys are restricted by an access policy defined by content providers, while the contents published from content providers are tagged by some attributes. Only when attributes associated with the contents satisfy the access policy of users' secret keys, the users are capable of decrypting the contents. A normal KP-ABE scheme consists of four algorithms as follows:

Setup$(\mathcal{U}) \rightarrow (PK, MK)$: The setup algorithm takes a universal set of attributes $\mathcal{U}$ and the implicit security parameter as input. It outputs public parameters PK and a master key MK.

Encrypt$(M, S, PK) \rightarrow CT$: The encryption algorithm takes a message plaintext M, a set of attributes S used to represent the message, and public parameters PK as input. Then it outputs the corresponding ciphertext CT of the message.

KeyGen$(\mathbb{A}, MK, PK) \rightarrow SK$: The key generation algorithm takes an access structure $\mathbb{A}$, master key MK, and public parameters PK as input. It outputs the decryption key SK that implicitly contains $\mathbb{A}$.

Decrypt$(CT, SK, PK) \rightarrow M$ *or* $\perp$: The decryption algorithm takes ciphertext CT, secret key SK, and public parameters PK as input. If the attribute set S of ciphertext CT satisfy the access structure $\mathbb{A}$ of SK, it outputs the message M. Otherwise, it outputs $\perp$.

C. Access Structure

Definition 2: Access Structure: Let $\{P_1, P_2, \ldots, P_n\}$ *be a set of parties. A collection* $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$ *is monotone if* $\forall B, C$*: if* $B \in \mathbb{A}$ *and* $B \subseteq C$*, then* $C \in \mathbb{A}$*. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection)* $\mathbb{A}$ *of non-empty subsets of* $\{P_1, P_2, \ldots, P_n\}$*, i.e.,* $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} \setminus \{\varnothing\}$*. The sets in* $\mathbb{A}$ *are called the authorized sets, and the sets not in* $\mathbb{A}$ *are called the unauthorized sets.*

In our paper, parties are presented by all kinds of attributes. The access structure $\mathcal{T}$ is expressed by a tree structure where each non-leaf node of the tree represents a threshold ("AND", "OR"), described by its children and a threshold value, and each leaf node represents an attribute. Denote n_x and k_x as the child node number and threshold value of a non-leaf node x. Particularly, x is an OR gate when $k_x = 1$, and x is an AND gate when $k_x = n_x$.

IV. System Model, Security Assumption and Design Goals

A. System Model

Our system in ICN is composed of four entities: *Internet Service Provider*, multiple *Content Providers*, *Central Authority*, *Users*, which is shown in Fig. 1.

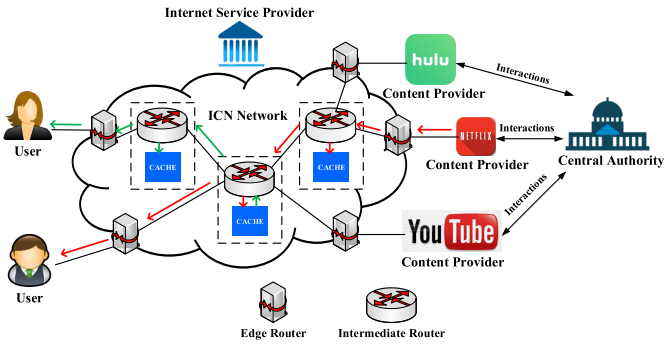


Fig. 1. System model.

- *Internet Service Provider (ISP)*: It owns an ICN network in our system and provides ICN access service to users and content providers. The routers in the ICN network are able to cache contents that are transmitted through them. According to their locations, we further divide them into two categories: edge routers and intermediate routers. Specifically, the edge routers directly connect with users and content providers, whereas intermediate routers do not. Intermediate routers perform caching and forwarding tasks, while edge routers are also responsible for authentication and content ownership verification in addition to the tasks of caching and forwarding.
- *Content Providers (CPs)*: Like hulu, they are data owners as well as the helpers of content deduplication. For one thing, they need to deal with user registration and key management. To issue secret keys to users successfully, CPs also play the role of AAs in the general KP-ABE scheme. For another, they may help ISP deduplicate cached contents when there are enough economic incentives.
- *Central Authority (CA)*: It is the administrator of the entire system, and is in charge of system initialization by generating system parameters and public keys of attributes. Also, it is the agent that helps CPs negotiate to decide which attribute is public, and it is responsible for public attribute announcement. In the real-world scenario, its task can be taken by a content provider league that is established by numerous content providers.
- *Users*: They are content consumers and possess secret keys bound with access structures. They can decrypt contents they request if and only if the attributes embedded in the contents satisfy the access structures of their secret keys.

B. Security Assumption

The ISP in our system is assumed to be semi-trusted. To be more precise, it is honest but curious. It will honestly follow the proclaimed protocols, but in the meantime, it also attempts to learn something from the contents it caches in the ICN network. We assume that CPs are also semi-trusted. They provide content subscription services to users and offer necessary assistance to ISP for content deduplication. However, they always intend to obtain the secret keys of private attributes of other content providers. Besides, we assume that different CPs adopt the same standard of encoding and media format, which means that different CPs will generate data packets with the same content fields for the same contents.

CA is a trusted entity in our system with the mission of initializing the entire system. Users in our system are assumed to be untrusted. They may try any means to obtain access permission beyond their privileges including colluding with each other.

C. Design Goals

In this paper, we intend to devise a secure content delivery and deduplication scheme for ICN, where multiple content providers are in the unified cryptographic system. Specifically, our scheme has the following design goals:

- **Data confidentiality.** The proposed scheme needs to ensure that unauthorized users and curious routers are unable to learn anything from the encrypted contents.
- **Scalable access control.** Our scheme should achieve fine-grained access control to meet the increasingly diverse access policies of content providers. In the meantime, it should be able to efficiently support the scenario of multiple content providers, which is specifically manifested as the key management overhead incurred by users is low.
- **Convenience in content deduplication.** The proposed content deduplication solution needs to be compatible with the entire security framework without complicated operations.
- **High efficiency.** The key management at the user side in our scheme should be simple and efficient. Specifically, the space of key management has low complexity. Also, our scheme should be efficient for users to fetch their desired contents.

V. Structure of SKP-ABE

In this section, we first illustrate how to manage the attributes and corresponding keys. Then we describe the main construction of SKP-ABE.

A. Overview

1) Attribute Management.

In SKP-ABE, to make some users' secret keys able to be commonly utilized by different AAs, we further categorize corresponding attributes as public attributes and private attributes. To be more specific, public attributes are generally expressed as common attributes that can be shared across content providers. There are usually two kinds of public attributes. One is the ones that represent users' identities like age, occupation, and so forth. The other is the ones that most content providers commonly have, such as the type of users and categories of videos. On the contrary, private attributes are those that content providers don't want to share with other CPs, and they generally are content providers' identities, unique user types and categories, and so on.

In our scheme, content providers determine which attributes are public and which attributes are private through CA by negotiating. Content providers can get the public attribute information from CA, and if they want to share an attribute, they can register this attribute to CA. Meanwhile, if a content provider, like Netflix, wants to set an attribute, which has been announced as a public attribute (e.g., "Member"), to be private, it can easily create a new attribute "Member_Netflix" instead of using the public attribute "Member". This process is exactly the same as the traditional CP-ABE, which shows that our scheme is compatible with the traditional ones in terms of attribute management.

In order to let users have as many public attribute keys as possible, in our system, only the attributes representing the identity of data owners (e.g., attribute "Netflix" for data owner Netflix) are the private attributes and others are all public attributes. Normally, to claim the ownership of content, when a data owner publishes content, the attribute set of the content always includes the data owner's private attribute. For example, the contents published by Netflix always have the "Netflix" attribute in their attribute sets. Meanwhile, the implicit access structures in users' secret keys are always the format: data owner's private attribute AND specific access policy (like the access structure in Fig. 2) to distinguish different data owners.

2) Key Management.

To make public attribute keys shared among different AAs, all the AAs can obtain the master keys of public attributes, and use these keys to generate corresponding public keys and users' attribute keys. It is noted that these attribute keys can be directly used to decrypt contents encrypted with the corresponding public keys of public attributes from any AA without any distinctions. However, since the master keys of private attributes can be solely obtained by the AAs that maintain these corresponding attributes, the public keys and users' attribute keys corresponding to private attributes can only be issued by the AAs that maintain them. Unlike public attributes, users' attribute keys of private attributes are only able to be used to decrypt contents encrypted with the corresponding public keys of private attributes from the issuer of these keys.

B. Main Construction

Our proposed SKP-ABE consists of four procedures: setup, key generation, encryption, and decryption as normal KP-ABE. The details in each procedure are as follows:

1) Setup: CA first generates a bilinear mapping group system $S = (p, \mathbb{G}, \mathbb{G}_T, g, e(\cdot,\cdot))$, where $\mathbb{G}, \mathbb{G}_T$ are cyclic multiplicative groups with the prime order p, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a bilinear mapping, and g is a randomly selected generator of $\mathbb{G}$.

Then CA randomly chooses a number $t_i \in \mathbb{Z}_p$ for each attribute $Att_i \in \mathcal{U}$ as a part of master key, where $\mathcal{U}$ is the universal set of public attributes. The public parameter of CA is published as:

$$PK_{CA} = \left(S, T_1 = g^{t_1}, \cdots, T_{|\mathcal{U}|} = g^{t_{|\mathcal{U}|}}\right).$$

If AA_i wants to join the system, it is required to register to CA first. Upon CA successfully verifies its identity and admits it to be the one part of the system, AA_i is able to obtain the master key corresponding to public attributes $(t_1, \cdots, t_{|\mathcal{U}|})$. Subsequently, it selects two random numbers $\alpha_i, \beta_i \in \mathbb{Z}_p$, where β_i is used as a part of master key corresponding to the private attribute. Then it can obtain its master key

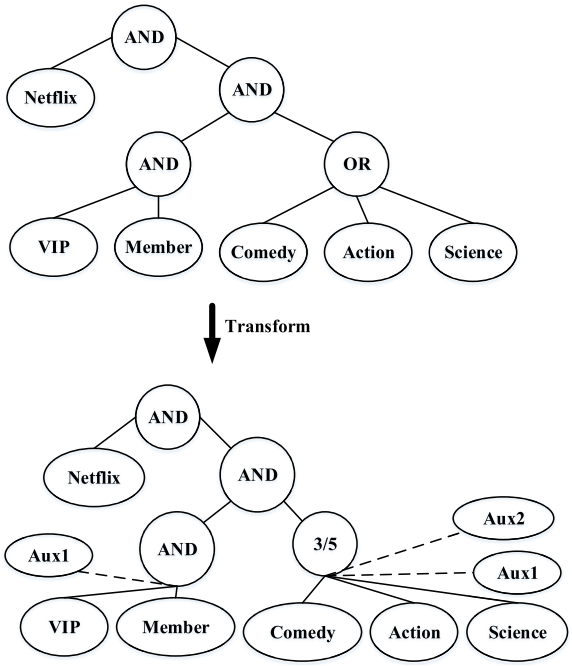


Fig. 2. Access structure transformation.

$MK_i = \{(t_1, \cdots, t_{|\mathcal{U}|}), \alpha_i, \beta_i\}$ and publishes its public parameters as:

$$PK_i = \Big(h_i = g^{\beta_i}, Y_i = e(g,g)^{\alpha_i}\Big).$$

Besides, we define the Lagrange coefficient $\Delta_{i,S}$ for $i \in \mathbb{Z}_p$ and a set S of elements in $\mathbb{Z}_p$: $\Delta_{i,S}(x) = \prod_{y \in S, y \neq i} \frac{x-y}{i-y}$.

2) Key Generation: When the user U_j registers to an AA, the AA needs to run the key generation algorithm, which includes the following steps:

- **Access structure transformation:** In order to maintain data confidentiality when reusing the public attribute keys, AA should transform U_j's access structure before generating secret keys. At first, AA adds $n_x - 1$ auxiliary nodes to each "OR" node x as its child nodes. Afterward, it adds one auxiliary child node to each "AND" node of which child nodes are all attribute nodes. Particularly, the child auxiliary node of each gate node can be the same. An example of access structure transformation is illustrated in Fig. 2.
- **Secret key generation and distribution:** In this step, there is a polynomial q_x which is associated with each node x in the transformed access structure $\mathcal{T}$. The polynomial q_x is generated in a bottom-up manner from leaf nodes to the root node R. For each non-leaf node x in $\mathcal{T}$, the degree d_x of polynomial q_x is set as $k_x - 1$. And for the leaf node x, d_x is set as 0. Besides, the polynomial q_x of non-root node x satisfies $q_x(0) = q_{parent(x)}(index(x))$. The procedure of key generation is a little different in two different situations: first registration and subsequent registrations to different CPs.

  i) For U_j's registration to first attribute authority AA_i, to generate U_j's secret keys, AA_i needs to choose a random number s_x for each non-auxiliary node x in $S_{\mathcal{T}}$ as its polynomial secret value (i.e., $q_x(0) = s_x$), where $S_{\mathcal{T}}$ is the set of all the leaf nodes in $\mathcal{T}$. Then AA_i can obtain the polynomial value $q_x(0)$ of each non-leaf node x which has a threshold from bottom to top by computing:

$$q_x(0) = \sum_{z \in S_x} \Big(q_z(0)\Delta_{z', S'_x}(0)\Big),$$

where $z' = index(z), S'_x = \{index(z) : z \in S_x\}$, and S_x is the d_x-size set of node x's child nodes which have their polynomial values. Finally, AA_i gets the polynomial value $s_R = q_R(0)$ of root node R. Then AA_i obtains the polynomial value $q_x(0)$ of auxiliary node x under an "OR" gate by computing $q_x(0) = q_{parent(x)}(index(x))$. After that, AA_i computes U_j's secret keys as follows:

$$SK = \Big(\mathcal{T}, D_i = g^{\alpha_i - s_R}, D'_i = g^{\frac{q_{x'}(0)}{\beta_i}}, \forall x \in S_{\mathcal{T},1} : D_x = g^{\frac{q_x(0)}{t_x}}, \forall x \in S_{\mathcal{T},2} : D_x = g^{q_x(0)}\Big),$$

where x', $S_{\mathcal{T},1}$, and $S_{\mathcal{T},2}$ represent private attribute, the set of the leaf nodes with the public attributes, and the set of auxiliary leaf nodes, respectively.

ii) After the first registration, when U_j wants to register to other different AAs, he/she needs to send all of his/her secret keys corresponding to public attributes to these AAs firstly. Assume that U_j registers to a new content provider $AA_{i'}$ and the transformed access structure in $AA_{i'}$ is denoted as $\mathcal{T}'$.

We also denote $S_{\widetilde{\mathcal{T}}',1}$ as the set of the leaf node x with public attributes that satisfies $x \in S_{\mathcal{T}',1} \wedge x \notin S_{\mathcal{T},1}$. Then, $AA_{i'}$ chooses random number $s_x = q_x(0) \in \mathbb{Z}_p$ for leaf node $x \in S_{\widetilde{\mathcal{T}}',1}$ and the leaf nodes with private attributes. $AA_{i'}$ computes $E_x = g^{q_x(0)}$. For the received secret keys corresponding to the leaf node x with public attributes, $AA_{i'}$ computes $E_x = D_x^{t_x} = g^{q_x(0)}$. Then, $AA_{i'}$ can obtain E_x of each non-leaf node x by computing:

$$E_x = \prod_{z \in S_x} E_x^{\Delta_{z', S'_x}(0)} = g^{q_x(0)},$$

where $z' = index(z), S'_x = \{index(z) : z \in S_x\}$ and S_x is the d_x-size set of node x's child nodes which have their polynomial values. After that, $AA_{i'}$ obtains the polynomial value $g^{q_x(0)}$ of auxiliary node x under an "OR" gate by computing $g^{q_x(0)} = g^{q_{parent(x)}(index(x))}$. Finally, it generates U_j's secret keys as follows:

$$SK = \Big(\mathcal{T}', D_{i'} = g^{\alpha_{i'}}/E_{R'}, D'_{i'} = g^{\frac{q_{x'}(0)}{\beta_{i'}}}, \forall x \in S_{\widetilde{\mathcal{T}}',1} : D_x = g^{\frac{q_x(0)}{t_x}}, \forall x \in S_{\mathcal{T}',2} : D_x = g^{q_x(0)}\Big),$$

where R' is the root node of $\mathcal{T}'$.

3) Encryption: To encrypt message M, the data owner needs to select a random number $r \in \mathbb{Z}_p$. Then, it assigns a public attribute set S_M and the private attribute corresponding to its identity to the message M. Suppose AA_i maintains the private

attribute, the data owner finally generates the ciphertext as follows:

$$CT = (S_M, L_i = MY_i^r, C_i = g^r, C_i' = h_i^r, \forall x \in S_M : C_x = T_x^r).$$

4) Decryption: After receiving the encrypted contents CT, U_j first checks whether attribute set S_M satisfies the access structure $\mathcal{T}$ implicitly included in the secret keys obtained from AA_i. If S_M doesn't satisfy $\mathcal{T}$, the decryption algorithm returns $\perp$; otherwise U_j decrypt CT in a bottom-up manner as follows:

- For the leaf node with private attribute, U_j computes
$$F_x = e(C_i', D_i') = e(g,g)^{q_x(0)r}.$$
- For a leaf node x with public attribute Att_i, if $Att_i \in S_M$, then U_j computes
$$F_x = e(C_x, D_x) = e(g,g)^{q_x(0)r}.$$
Otherwise, $F_x = \perp$.
- For an auxiliary leaf node x, U_j computes
$$F_x = e(C_i, D_x) = e(g,g)^{q_x(0)r}.$$
- For a non-leaf node x, let S_x be an arbitrary k_x-size set of its child nodes. If set S_x exists, each element z of which satisfies $F_z \neq \perp$, then U_j computes
$$F_x = \prod_{z \in S_x} F_z^{\Delta_{z',S_x'}(0)} = e(g,g)^{q_x(0)r}.$$
Otherwise, set F_x as $\perp$.
- Finally, U_j gets $F_R = e(g,g)^{s_R r}$ or $F_R = \perp$ for root node R. If $F_R \neq \perp$, then U_j can recover the message by computing
$$M = \frac{L_i}{e(C_i, D_i)e(g,g)^{s_R r}} = \frac{Me(g,g)^{\alpha_i r}}{e(g,g)^{(\alpha_i - s_R)r}e(g,g)^{s_R r}}.$$

VI. Our Proposed Scheme: SCD2

A. Overview

SCD2 can be mainly divided into four phases: user registration, content publication, content retrieval, and content deduplication. Also, CPs play the role of both AAs and data owners in SKP-ABE for the tasks of secret key distribution and content publication, respectively.

When users register with CPs, CPs act as the AAs in SKP-ABE to assign secret keys to the users. With the introduction of public attributes in SKP-ABE, users can manage their keys far more efficiently. When CPs want to publish some contents, they act as the data owners in SKP-ABE. In order to achieve convenient content deduplication, before CPs publish contents to the core network, they are required to conduct a challenge-response process to prove their ownership of the contents that they claim to publish. This way, it can assure that the contents cached in the core network are all the contents that CPs indeed own. This is to prevent attackers from publishing some faked contents with correct tags as real ones to expect routers to delete their faked contents during deduplication and finally obtain the ownership of those real contents. Similarly,

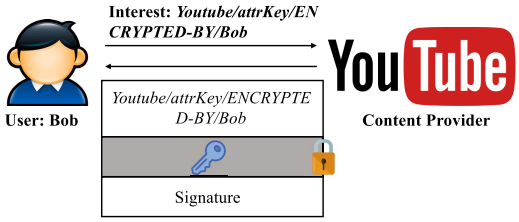


Fig. 3. Interactions in user registration phase.

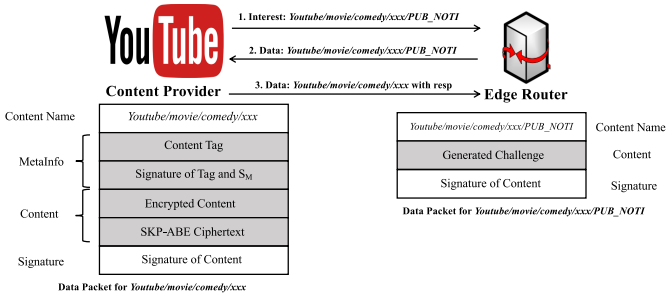


Fig. 4. Interactions in content publication phase.

a challenge-response verification mechanism is also introduced in content retrieval phase and only legitimate requests are injected into the core network by edge routers for protecting the network resources. Finally, routers can randomly be in contact with one of the CPs which publish the identical popular contents and achieve efficient data deduplication on these popular contents.

In the following subsections, we take the NDN as an example of ICN architecture to illustrate the construction of our scheme.

B. User Registration

To acquire the capability of decrypting the contents from different CPs, users need to register with these CPs to get their secret keys firstly. For example, Fig. 3 shows the process of the registration of a user named Bob. After receiving the interest from Bob, the content provider generates the corresponding attribute keys for Bob by following the method proposed in SKP-ABE. Then, it encrypts the attribute keys by Bob's public key and returns the corresponding data packet.

C. Content Publication

As shown in Fig. 4, when a CP wants to publish contents, it sends an interest which has the suffix *"PUB_NOTI"* to the edge router to start the Proof of Ownership (POW) process. Here, we assume that CP_i gets an extra secret key pair: $sk_i = x_i, pk_i = g^{x_i}$ when CP_i registers with CA. Then, the edge router randomly chooses a number $chal \in \mathbb{Z}_p$ as the challenge and replies to the interest with the corresponding data packet which contains the generated challenge.

Upon receiving the data packet that contains the challenge from an edge router, CP firstly computes a tag, $Tag = H(g^{H(M)})$, to reveal the correspondence between the tag and content M, which helps routers determine whether two pieces of data are potential duplicates. Here $H(\cdot)$ is a hash function: $\{0,1\}^* \rightarrow \mathbb{Z}_p$. Afterwards, CP generates the corresponding

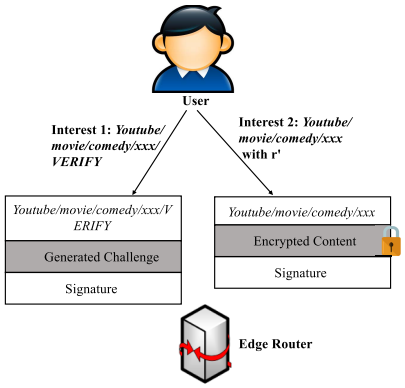


Fig. 5. Interactions in content retrieval phase.

response by computing $resp = H(M) - x_i \times chal$. To preserve data confidentiality, CP needs to encrypt the content to be published. CP chooses a random number k as a secret key to encrypt content M by using a symmetric encryption algorithm such as AES and encrypts the secret key k by using the SKP-ABE algorithm with attribute set S_M. Thus, it gets the encrypted content $\widehat{C} = E_k(M)$ and the SKP-ABE ciphertext $\widetilde{C} = ABE.Enc(k, S_M)$. The CP generates the data packet of content to be published. Here content tag and signature $Sig(S_M, Tag)$ of Tag and S_M form the MetaInfo field, and $\widehat{C}$ and $\widetilde{C}$ form the Content field. Finally, CP publishes the content by sending the data packet with $resp$ to the edge router.

Instead of transmitting the encrypted contents to the core network directly, the edge router is required to verify the ownership of the contents first by checking whether $H(g^{resp} \times pk_i^{chal}) = Tag$ and then verify the validity of $Sig(S_M, Tag)$. If so, the edge router believes CP indeed owns these contents and injects them into the core network; otherwise, it abandons these contents. Finally the content M is stored as $(\widehat{C}, \widetilde{C}, Tag)$ in the routers in ICN.

D. Content Retrieval

Here we suppose U_j requests content M published by CP_i. To speed up the authentication process, edge routers can pre-compute some auxiliary information:

$$\forall CP_i \in S_{cp} : AI_i = (\delta Y_i^{\lambda}, g^{\lambda}, h_i^{\lambda}), \forall x \in \mathcal{U} : AI_x = T_x^{\lambda},$$

where S_{cp} represents the universal set of different CPs, $\mathcal{U}$ is the universal set of public attributes, and δ, λ are the random numbers in $\mathbb{Z}_p$.

As shown in Fig. 5, U_j first needs to send an interest which has the suffix *"VERIFY"*. Upon receiving U_j's interest for verification, the edge router finds out the corresponding attribute set S_M and sends the data packet to U_j which contains the challenge $chal = \{(AI_i, AI_x), E_{H(\delta)}(r)\}$, where (AI_i, AI_x) is the encryption result $ABE.Enc(\delta, S_M)$ using SKP-ABE, $x \in S_M$ and r is randomly chosen number. If the user is an authorized user, he/she is able to decrypt $ABE.Enc(\delta, S_M)$ and get the result δ' using secret keys obtained from CP_i and other CPs. Finally, the user can recover the plaintext r' from $E_{H(\delta)}(r)$ and sends the interest for the content with r'. The edge router believes the request is legitimate and forwards it to the ICN network if and only if the received response r' is equal to r.

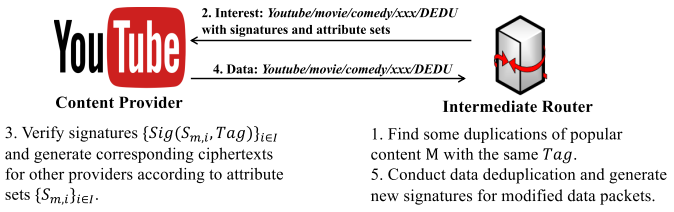


Fig. 6. Interactions in content deduplication phase.

Besides, we can further accelerate request authentication by considering the continuity of users' requests (i.e., users always continuously request a serial of different chunks of the same content). For the request of the first chunk of content, the authentication process is the same as that aforementioned. But for the subsequent requests of the chunks of the same content, we can use the method proposed in [20] to replace the complicated challenge-response authentication with simple hash authentication.

Specifically, when U_j requests the first chunk of content, he/she generates a hash chain with proper length and sends the response with hash chain tail H_{tail} piggybacked. After the first successful authentication, the edge router maintains an authentication state table (AST), which has two fields: the prefix of the filename f and the latest received hash value H_{lt}. Here H_{lt} is initialed as H_{tail} at first. So for the subsequent requests, U_j only needs to use the hash chain in reverse order and sends the request with the new element H_{new}. Due to the one-way property of the hash chain, the edge router can easily authenticate these requests by finding the item in AST which satisfies $H_{lt} = H(H_{new})$ with the same filename prefix. If such an item is found, it means that the authentication passes and the edge router update the latest received hash value as H_{new}.

E. Content Deduplication

TABLE I shows the content store (CS for short) table when routers in ICN cache the same content M published by different CPs without deduplication. We can see that although these content are identical with the same tag, routers must store complete ciphertext from different CPs separately because of the different symmetric keys and randomly chosen numbers in SKP-ABE.

Considering the limited storage resources of routers in ICN, it is necessary for routers to deduplicate cached contents. Since unpopular contents cached in the routers always be replaced frequently, we only consider the popular contents which always take up cache space for a long time. As shown in Fig. 6, to conduct content deduplication, intermediate routers first need to check whether there are popular cached contents with the same tag Tag regularly. If so, intermediate routers send one of the providers of these identical contents an interest with the signatures $\{Sig(S_{M_i}, Tag)\}_{i \in I}$ and attribute sets $\{S_{M_i}\}_{i \in I}$ of other CPs. Here I is the set of the identifiers of other CPs, and S_{M_i} represents the public attribute set of content M published by CP_i. Then, the contacted CP

TABLE I
CACHED CONTENTS IN CONTENT STORE TABLE

(a) Content Store Table without Deduplication

Content Name	Cached Contents		
	MetaInfo	Content	Signature
youtube/movie/comdey/xx	$Tag, Sig(S_{M_i}, Tag)$	$E_k(M), L_i, C_i, C'_i, \{C_x\}_{x \in S_{M_i}}$	Sig_i
hulu/movie/comdey/xx	$Tag, Sig(S_{M_{i'}}, Tag)$	$E_{k'}(M), L_{i'}, C_{i'}, C'_{i'}, \{C_x\}_{x \in S_{M_{i'}}}$	$Sig_{i'}$
netflix/movie/comdey/xx	$Tag, Sig(S_{M_{i''}}, Tag)$	$E_{k'}(M), L_{i''}, C_{i''}, C'_{i''}, \{C_x\}_{x \in S_{M_{i''}}}$	$Sig_{i''}$

(b) Content Store Table with Deduplication

Content Name	Cached Contents			
	MetaInfo	Content		Signature
youtube/movie/comdey/xx	$Tag, Sig(S_{M_i}, Tag)$	L_i, C'_i	$E_k(M), C_i, \{C_x\}_{x \in S_{M \|\| \mathcal{U}}}$	$Sig_{ISP}(M_i)$
hulu/movie/comdey/xx	$Tag, Sig(S_{M_{i'}}, Tag)$	$L_i, C'_{i'}$		$Sig_{ISP}(M_{i'})$
netflix/movie/comdey/xx	$Tag, Sig(S_{M_{i''}}, Tag)$	$L_i, C'_{i''}$		$Sig_{ISP}(M_{i''})$

verifies received signatures and encrypts the symmetric key used in its own content with the same random number and different attribute sets $\{S_{M,i}\}_{i \in I}$ of other CPs. Besides, in order to enhance CPs' positivity in helping ISP conduct content deduplication, ISP needs to offer CPs some economic incentives. Finally, the intermediate routers maintain only one copy of M, which are symmetrically encrypted, and delete the duplicate ones. Besides, they need to generate new signatures $Sig_{ISP}(M_i)$ for modified contents to make the built-in security mechanism of ICN unaffected. Here we assume all the intermediate routers own the private key of ISP.

Let us take TABLE I(a) as an example. When an intermediate router finds that there are some duplications of popular content M in its cache space, it randomly chooses one of the providers. Here it chooses to send an interest packet with the signatures $Sig(S_{M_i}, Tag)$ and S_{M_i} of Hulu and Netflix to Youtube. After verifying these signatures successfully, Youtube generates the corresponding ciphertext as follows:

$$L_{i'} = kY_{i'}^r, \quad C'_{i'} = h_{i'}^r, \; L_{i''} = kY_{i''}^r, \; C'_{i''} = h_{i''}^r,$$
$$\forall x \in S_{M_\mathcal{U}} - S_{M_i} : C_x = T_x^r,$$

where i, i', i'' are the identifiers of Youtube, Hulu, and Netflix, respectively, k is the symmetric key and r is the random number used by Youtube to encrypt its own M and k, respectively, and $S_{M,\mathcal{U}}$ represents the set of public attributes used by all the three CPs. Then, Youtube returns these results to the router which asks for content deduplication. Finally, a lot of redundant data can be deleted and new signatures are generated. TABLE I(b) shows details in the CS table with deduplication, where many data can be shared in different CPs.

VII. SECURITY ANALYSIS

A. Data Confidentiality

Lemma 1: If DBDH assumption holds on group $\mathbb{G}$, there is no polynomial-time adversary that can break our SKP-ABE in the Attribute-based Selective-Set model (proposed in [26]) with non-negligible advantage.

Proof: Suppose we have a polynomial-time adversary $\mathcal{A}$ that can break our scheme in the Selective-Set model with a non-negligible advantage $Adv_\mathcal{A}$. Now we show how to construct an algorithm $\mathcal{B}$ that is able to break the DBDH assumption with non-negligible advantage.

Initialize. Challenger $\mathcal{C}$ initializes the system with a bilinear mapping e, groups $\mathbb{G}, \mathbb{G}_T$ and generator $g \in \mathbb{G}$. Then, $\mathcal{C}$ tosses a secure random coin $\mu \in \{0, 1\}$ and generates a tuple $(A, B, \bar{C}, Z)$. If $\mu = 0$, $\mathcal{C}$ sets the tuple as $(g^a, g^b, g^c, e(g, g)^{abc})$; otherwise, the tuple is set as $(g^a, g^b, g^c, e(g, g)^z)$, where a, b, c, z are random numbers in $\mathbb{Z}_p$. Then, algorithm $\mathcal{B}$ runs $\mathcal{A}$. $\mathcal{A}$ chooses a public attribute set S_i for each chosen content provider CP_i, which will be challenged later.

Setup. Algorithm $\mathcal{B}$ generates public parameters for each chosen CP_i. Set $Y_i = e(g, g)^{ab}$ (α_i is implicitly set as ab) and $h_i = g^{\gamma_i}$ (i.e., $\beta_i = \gamma_i$), where γ_i is a random number in $\mathbb{Z}_p$. For any attribute $x \in S_i$, it chooses a random number $\gamma_x \in \mathbb{Z}_p$ and sets $T_x = g^{\gamma_x}$ (i.e., $t_x = \gamma_x$). For every other attribute $x \in \mathcal{U} \backslash S_i$, it sets $T_x = g^{b\eta_x} = B^{\eta_x}$ (i.e., $t_x = b\eta_x$), where random number $\eta_x \in \mathbb{Z}_p$. Then, $\mathcal{B}$ sends the public parameters to $\mathcal{A}$.

Secret Key Queries. $\mathcal{A}$ adaptively makes requests to each chosen content provider CP_i for secret keys corresponding to any access structures $\mathcal{T}_i$, which is not satisfied by attribute set S_i. Let CP_j be the first content provider queried.

- The CP queried by $\mathcal{A}$ is CP_j. To generate the secret keys, $\mathcal{B}$ needs to assign a polynomial Q_x of degree d_x for every node in the access tree $\mathcal{T}_j$. We first define the following two procedures: PolySat and PolyUnsat.
  $PolySat(\mathcal{T}_x, S)$: This procedure sets up the polynomial for the root node x of a 2-layer access subtree $\mathcal{T}_x$ with satisfied leaf nodes x'. If x is an "AND" node, for all leaf nodes x' in subtree $\mathcal{T}_x$ without a polynomial, it randomly chooses number $s_{x'} \in \mathbb{Z}_p$ and sets $q_{x'}(0) = s_{x'}$. If x is an "OR" node, for the attribute leaf nodes x' in subtree $\mathcal{T}_x$ without a polynomial, do the same thing. For the attribute leaf nodes x' in subtree $\mathcal{T}_x$ with a polynomial in both situations, it directly uses these polynomials. By interpolation, it can easily get the polynomial of root node $q_x(0)$. For the "OR" node x, it finally needs to generate the auxiliary child nodes' polynomial $q_{x'} = q_x(index(x'))$.

$PolyUnsat(\mathcal{T}_x, S)$: This procedure sets up the polynomials for the root node x of a 2-layer access subtree $\mathcal{T}_x$ whose leaf nodes x' are not all satisfied nodes. If x is an "AND" node, for all leaf nodes in subtree $\mathcal{T}_x$ without a polynomial, if x' is a satisfied node, it randomly chooses a number $s_{x'} \in \mathbb{Z}_p$ and sets $q_{x'}(0) = s_{x'}$; otherwise, it defines the polynomial of x' as $g^{q_{x'}(0)} = g^{s_{x'}}$. If x is an "OR" node, it only generates polynomials for attribute leaf nodes x' without a polynomial like "AND" node. For the leaf nodes x' in subtree $\mathcal{T}_x$ with a polynomial, it directly uses these polynomials. By interpolation, it gets the polynomial of their parents' node $g^{q_x(0)}$. For "OR" node x, it finally needs to generate the auxiliary child nodes' polynomial $g^{q_{x'}} = g^{q_x(index(x'))}$.

To give the secret keys for access structure $\mathcal{T}_j$, $\mathcal{B}$ runs algorithm $PolySat(\mathcal{T}_x, S)$ or $PolyUnsat(\mathcal{T}_x, S)$ for each node from the penultimate layer to the first layer in order according to the situation of subtree $\mathcal{T}_x$. Finally, $\mathcal{B}$ gets the polynomial $g^{q_R(0)}$ of the root node in $\mathcal{T}_j$. We let $g^{q_R(0)} = g^{a-\bar{x}}$, where $\bar{x} = a - q_R(0)$. Notice that for each attribute leaf node x of $\mathcal{T}_j$, we know q_x completely if x is satisfied; if x is not satisfied, then at least $g^{q_x(0)}$ is known (in some cases q_x might be known completely).

Now $\mathcal{B}$ defines the final polynomial $Q_x(\cdot) = bq_x(\cdot)$ for each node x in $\mathcal{T}_j$. Then the secret keys for access structure $\mathcal{T}_j$ is constructed as:

$$\begin{aligned} SK = \Big(&\mathcal{T}_j, D_j = g^{\alpha_j - q_R(0)} = g^{b\bar{x}} = B^{\bar{x}}, \\ &D'_j = g^{\frac{Q_x(0)}{\beta_j}} = g^{\frac{bq_x(0)}{\gamma_j}} = B^{\frac{q_x(0)}{\gamma_j}}, \\ &\forall x \in S_{\mathcal{T}_j,2} : D_x = g^{Q_x(0)} = g^{bq_x(0)} = B^{q_x(0)}, \\ &\forall x \in S_{\mathcal{T}_j,1} \cap S_j : \\ &D_x = g^{\frac{Q_x(0)}{t_x}} = g^{\frac{bq_x(0)}{\gamma_x}} = B^{\frac{q_x(0)}{\gamma_x}}, \\ &\forall x \in S_{\mathcal{T}_j,1} - S_j : \\ &D_x = g^{\frac{Q_x(0)}{t_x}} = g^{\frac{bq_x(0)}{b\eta_x}} = g^{\frac{q_x(0)}{\eta_x}}\Big). \end{aligned}$$

- The CP queried by $\mathcal{A}$ is not CP_j denoted as $CP_{j'}$. For the leaf nodes $x \in \mathcal{T}_j \cap \mathcal{T}_{j'}$, $\mathcal{B}$ will use the same polynomial q_x. For other nodes, $\mathcal{B}$ generates polynomials and the final secret keys as what it does when CP_j is queried.

Challenge. $\mathcal{A}$ submits two challenge messages m_0 and m_1 to $\mathcal{B}$ for CP_i. Then, $\mathcal{B}$ tosses a secure coin $\nu \in \{0, 1\}$ and returns an encryption result of m_ν to $\mathcal{A}$. The ciphertext is constructed as:

$$CT = \Big(L_i = m_\nu Z, C_i = \bar{C}, C'_i = \bar{C}^{\gamma_i}, \{C_x = \bar{C}^{\gamma_x}\}_{x \in S_i}\Big).$$

If $\mu = 0$, then $Z = e(g, g)^{abc}$. We let $r = c$ and we can have $Y_i^r = (e(g, g)^{ab})^c = e(g, g)^{abc} = Z$, $C_i = g^c = \bar{C}$, $C'_i = (g_i^\gamma)^c = \bar{C}^{\gamma_i}$, and $C_x = (g_x^\gamma)^c = \bar{C}_x^\gamma$. Therefore, ciphertext CT is a valid random encryption of message m_ν.

If $\mu = 1$, then $Z = e(g, g)^z$. Since z is a random number in $\mathbb{Z}_p$, $L_i = m_\nu e(g, g)^z$ is a random element of $\mathbb{G}_T$ which contains no information of m_ν from the view of $\mathcal{A}$.

More Secret Key Queries. $\mathcal{A}$ repeats secret key queries phase to request secret keys for other access structures that can not be satisfied by set S_i from each chosen CP_i.

Guess. $\mathcal{A}$ outputs its guess ν' of ν. If $\nu' = \nu$, $\mathcal{B}$ outputs its guess $\mu' = 0$ to indicate that $Z = e(g, g, g)^{abc}$; otherwise, it outputs guess $\mu' = 1$ to indicate that Z is a random element in $\mathbb{G}_T$.

In the case where $\mu = 1$, i.e., Z is random in $\mathbb{G}_T$, $\mathcal{A}$ can not get any information on it. So we have $Pr[\nu' = \nu | \mu = 1] = \frac{1}{2}$. Otherwise when $\mu = 0$, the ciphertext CT is a valid ranadom encryption of m_ν. Since $\mathcal{A}$ has an advantage $Adv_\mathcal{A}$ to break our scheme in Selective-Set model, we have $Pr[\nu' = \nu | \mu = 0] = \frac{1}{2} + Adv_\mathcal{A}$. Then, we have $Pr[\mu' \neq \mu | \mu = 1] = \frac{1}{2}$ and $Pr[\mu' = \mu | \mu = 0] = \frac{1}{2} + Adv_\mathcal{A}$ because when $\nu' = \nu$, $\mathcal{B}$ guesses $\mu' = 0$.

Finally we can get the overall advantage of algorithm $\mathcal{B}$ in DBDH game:

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

We can conclude that if there is a polynomial-time adversary who can break our scheme in Selective-Set model with non-negligible advantage $Adv_\mathcal{A}$, we can find an algorithm to solve the $DBDH$ problem with a non-negligible advantage $Adv_\mathcal{A}$. So *Lemma 1* is proved. □

Lemma 2: If the security of Shamir's secret sharing scheme holds [51], any malicious user cannot decrypt the contents beyond their privileges through shared attribute secret keys.

Proof: The access structure transformation implemented before issuing the secret keys can be divided into two parts: adding the corresponding number of auxiliary nodes to each "OR" gate, and adding one auxiliary node to "AND" gate, of which child nodes are all leaf nodes. We show how these two parts prevent malicious users from utilizing shared attribute secret keys to break confidentiality.

After implementing the access transformation, according to Lagrangian Interpolation, we can find that the polynomials of nodes $x_1, x_2, \cdots, x_n$ are not the same, i.e., $q_{x_1}(0) \neq q_{x_2}(0) \neq \cdots \neq q_{x_n}(0)$, where $x_1, x_2, \cdots, x_n$ are the child nodes of an "OR" gate in the access structure assigned by CP_i. Thus, unlike traditional KP-ABE, in which the polynomials of child nodes under "OR" gates are the same, users cannot get extra information through "OR" gates in our scheme besides the polynomials of auxiliary nodes.

If malicious user U_j successfully decrypt the contents of which the attribute set dissatisfies the "AND" gate with the same n attribute child nodes and an extra auxiliary child node in his/her access structure assigned by $CP_{i'}$, it means he/she can reconstruct the polynomial of the "AND" gate, through limited polynomials of child nodes and auxiliary nodes x_a under aforementioned "OR" gate $\{q_{x_i}(0)\}_{i=0}^{t}, \{q_{x_a}(0)\}_{a=0}^{n-1}$, where $t < n$. It is obviously breaks the security of Shamir's secret sharing scheme. Thus *Lemma 2* is proved. □

Based on *Lemma 1* and *Lemma 2*, we can conclude that data confidentiality can be guaranteed in our system.

TABLE II
COMPARISON WITH OTHER ACCESS CONTROL SCHEMES

Scheme	Data Confidentiality	IFA* Resistance	Offline CP	Multiple CPs	Fine-grained	Additional Features
SEAF [20]	Yes	Yes	Yes	Yes	No	Service Accountability
Capability [5]	No	Yes	Yes	Yes	No	-
CHTDS [17]	Yes	No	Yes	No	Yes	-
FGAC-NDN [18]	Yes	No	No	No	Yes	-
AccConf [12]	Yes	No	Yes	No	No	-
TACTIC [28]	Yes	Yes	Yes	Yes	No	-
SCD2	Yes	Yes	Yes	Yes	Yes	Content Deduplication

* Interest flooding attack

B. Impersonation Attack Resistance

Lemma 3: If the security of Shamir's secret sharing scheme holds, any content provider cannot impersonate a user through the public attribute keys that the user sent to it during registration process.

Proof: Through the key generation process of SKP-ABE, we can see that a user's secret keys consist of five parts: access structure $\mathcal{T}$, the key corresponding to root node D_i, private attribute key D_i', public attribute keys D_x and keys corresponding to auxiliary leaf nodes D_x. To impersonate a user, the content provider needs to obtain all of these keys, which means that given public parameters PK_{CA}, PK_i of CA and the target CP_i and public attribute keys D_x, the content provider can get other four parts of the user's secret keys.

First, due to the limited possibilities of the access structure $\mathcal{T}$, it is likely for the dishonest content provider to obtain it by brute force. Then, since the polynomial values in D_i and auxiliary node keys D_x are selected by random, if this content provider wants to impersonate a user, it can only guess the polynomial value by guessing without any useful information. Because normally the prime order p is a large number, the probability of the dishonest content provider guessing correctly is negligible. Finally, for the key corresponding to root node D_i, if this content provider can obtain it with public attribute keys only, it means that it can break the security of Shamir's secret sharing scheme, which obviously contradicts our assumptions.

This completes the proof. □

C. Interest Flooding Resistance

Lemma 4: Any adversary cannot launch interest flooding attacks by sending a mass of illegitimate requests to exhaust resources in ICN.

Proof: The edge routers in our system are the guards for inspecting the illegitimate requests. To access certain content, users must pass the authentication at the edge router side by decrypting the ciphertext $ABE.Enc(\delta, S_f)$ received from edge routers. Since if and only if the access structures $\mathcal{T}$ implicitly contained in users' secret keys are satisfied by attribute set S_f, the ciphertext can be decrypted, any adversary who is not granted access to the content (i.e., S_f cannot satisfy his/her access structure) is unable to pass the authentication and his/her requests will be abandoned by edge routers at the very beginning. Therefore our system can guarantee the resistance of interest flooding. □

D. Unforgeability

Lemma 5: The correct responses in POW process for publishing some contents can not be forged by any adversary who has no ownership of these contents to deceive edge routers and get ownership during content deduplication.

Proof: In our system, CPs must finish the POW process to prove that they have ownership of the contents to be published. If some adversaries break the POW process, they can publish some faked contents with the correct tags of the contents they expect to own. There is going to be a high probability that their faked contents are deleted during deduplication and they successfully obtain the ownership of real contents from other CPs.

However, the adversaries must get the correct $H(M)$ of the content they want to disguise if they try to forge correct responses. Suppose adversaries can get all the messages during the challenge-response process. If an adversary can compute $H(M)$ without the knowledge of M, i.e., given $resp = H(M) - x_i \times chal, Tag = H(g^{H(M)}), chal$, it can compute $H(M) = resp + x_i \times chal$ or $H(M) = log(H^{-1}(Tag))$. The former attempt is impossible unless the adversary can get the private key x_i, but leaking private key is out of our consideration. The latter one is obviously contradicting with the one-way property of hash function and intractability of discrete logarithm problem. Thus, the correctness of *Lemma 5* can be ensured. □

E. Feature Comparison

We compare our proposed SCD2 with several other schemes in terms of data confidentiality, interest flooding resistance, offline CP, multiple CPs, fine-grained access control, and additional features. As shown in TABLE II, scheme Capability doesn't consider data confidentiality, and CHTDS, FGAC-NDN, and AccConf are vulnerable to IFA. Among these schemes, only FGAC-NDN needs an online CP to respond to requests. Besides, although some schemes have the features of supporting multiple CPs and some schemes can support fine-grained access control, none of them can have both features and achieve high scalability except SCD2. As for additional features, SEAF and SCD2 support service accountability and content deduplication respectively. Above all, only our proposed SCD2 possesses all of the desired security features.

VIII. PERFORMANCE EVALUATION

In this section, we first analyze the storage, communication, and computation overhead of our proposed scheme. Then,

TABLE III
STORAGE OVERHEAD

Scheme	Content Provider	Router	User
SEAF [20]	$N_u\|p\|$	$N_{m,2}(\|C\|+2\|p\|)$	$(1+2^{N_x-1})N_A\|p\|$
Capability [5]	N/A	$N_{m,2}\|C\|+(\|\mathcal{T}_P\|+\|\mathcal{T}_Q\|)\|H\|$	$2^{N_x-1}N_A\|H\|$
TACTIC [28]	N/A	$N_{m,2}\|C\|+\|BF\|$	$N_A\|Tag\|$
SCD2	$(4+N_i)\|p\|$	$(N_{m,2}-\Delta_m)\|C\|+N_{m,2}\|Sig\|$ $+N_{m,2}(3+N_{a,2}-2\Delta_m-\Delta_x)\|p\|$	$(N_A(2+N_{au}+N_x)-\Delta_x)\|p\|$

TABLE IV
COMMUNICATION OVERHEAD

Scheme	System Initialization		Key Generation and Distribution			Encryption/Decryption
	CA	AA	CA	AA	User	Owner
Li [34]	Not exist	N/A	Not exist	$3N_{i,j}\|p\|$	$3N_{i,j}\|p\|$	$(3+2N_{a,2})\|p\|$
Chase [32]	$N_A\|p\|+N_{a,1}\|p\|$	$(N_i+1)\|p\|$	$\|p\|$	$N_{i,j}\|p\|$	$(N_{i,j}+1)\|p\|$	$(2+N_{a,2})\|p\|$
Chase [33]	Not exist	$(N_A-1)\|p\|$	Not exist	$(N_{i,j}+5)\|p\|$	$5(N_A-1)\|p\|+N_{i,j}\|p\|$	$(2+N_{a,2})\|p\|$
SKP-ABE	$N_{a,1}\|p\|$	$N_{a,1}\|p\|$	N/A	$(N_{i,j}+N_{au})\|p\|$	$(N_{i,j}+N_{au})\|p\|$	$(2+N_{a,2})\|p\|$

we implement SCD2 in a simulated NDN network to test the performance of content retrieval and deduplication.

A. Overhead Analysis

For clarity of description, we have the following definitions: Let p be the size of elements in the groups with prime order p, $|C|$ be the size of symmetrically encrypted ciphertexts, $|H|$ be the size of hash values and $|Sig|$ be the size of signatures. Besides, we denote $N_u, N_{m,1}, N_{m,2}, N_{m,3}, N_A, N_{a,1}, N_{a,2}, N_i,\ N_{i,j}, \Delta_m, \Delta_x$ as the number of users, contents published by CPs, contents cached in routers, contents requested by users, AAs (i.e., CPs in this paper), all attributes, the attributes corresponding to certain content, attributes maintained by CP_i, attributes obtained in U_j's access structure, contents deleted during deduplication, and duplicated attributes in different access structures, respectively. N_{au} represents the average number of the auxiliary nodes in users' access structure from different CPs and N_x represents the average number of attributes in users' access structure from different CPs.

1) Storage Overhead We compare SCD2 with other schemes that can support multiple CPs in terms of storage overhead with the same number of access policies used. Specifically, when there are N_x attributes used to generate access structures and suppose only "AND" gates exist in these structures, CPs can define 2^{N_x} different access policies.

TABLE III shows the details of storage overhead on each entity. Except for Capability [5] and TACTIC [28], all of the other schemes mentioned here, i.e., SEAF [20] and our scheme, have storage overhead on CPs. In SEAF [20], CP needs to store every user's information to trace them. Similarly, in SCD2, CP is required to store secret information for each attribute and its secret key pair $\{sk_i, pk_i\}$.

From the perspective of routers, SEAF, Capability, and TACTIC all need to store N_m ciphertexts and some other information to help with decryption or authentication. In TABLE III, $\mathcal{T}_P, \mathcal{T}_Q$ are the sets used to authenticate requests in Capability, and BF is the bloom filter in TACTIC, and the size of BF increases as the number of users and content providers grows. While with content deduplication, SCD2 can store less Δ_m ciphertexts. Considering the fact that the size of ciphertexts $|C|$ is much larger than other terms, SCD2 is able to store more useful contents with the same size of cache space, which will also be confirmed in Section VIII-B.

The storage overhead on users in TACTIC is proportional to N_A. In SEAF and Capability, it is related to N_A and 2^{N_x}. So as the number of CPs, contents, and access policies grow, the storage overhead on users will increase rapidly. Due to the huge number of contents and CPs in the real world, the size of secret keys each user stores in these schemes will be extremely large.

Our proposed scheme has the least storage overhead on user because users just need to store the secret keys of different attributes and additional auxiliary nodes. Since with more CPs, the probability of duplicated attributes being utilized to construct access structures by different CPs will be larger. In the most extreme cases where a user already has the secret keys of all the public attributes, the storage overhead will turn to be $(2N_A+N_{a,1}+N_AN_{au})|p|$. Even if compared with the scheme using standard KP-ABE whose storage overhead at user side is $N_A(N_x+2)|p|$, our scheme also has big advantages when the number of CPs is large because N_{au} is much less than the number of duplicated attributes. Fig. 7 illustrates the storage cost at user with the fixed probability of duplicated attributes: 10% and $|p|$ equals 160 bits. We can see that even with 43 CPs and 10 attributes on average, a user only needs to store 12.9 KB secret keys, which is efficient enough in key management.

2) Communication Overhead The communication overhead comparison among SKP-ABE, Chase's schemes [32], [33] (denoted as scheme-1 and scheme-2) and Li's scheme [34] is shown in TABLE IV. During the system initialization phase, since AAs in Li's scheme are entirely independent, there is no need for them to communicate. However, each AA in Chase's scheme-1 and scheme-2 and our scheme needs

TABLE V
COMPUTATION OVERHEAD IN KEY GENERATION

Scheme	$\mathbb{Z}_p$			$\mathbb{G}$	
	LCC	Multiplication	Addition	Exponentiation	Multiplication
Li's scheme [34]	N/A	$n(2l+1)$	$2n(l-1)$	$6n$	$3n$
Chase's scheme-1 [32]	N/A	$d(d-1)/2+n$	d	n	N/A
Chase's scheme-2 [33]	N/A	$N_A+n+d(d-1)/2$	$d+N_A(N_A-1)$	$2(N_A-1)+n$	$(N_A+1)(N_A-1)$
SKP-ABE (1_{st} CP)	d	$n+d-n_2$	1	$3+n_{au}$	N/A
SKP-ABE (n_{th} CP)	d	$n+d-n_2-\Delta n_p$	N/A	$3+n_{au}+d+n$	$d-n_2+1$

* n, n_2, n_{au}, d represent the number of attribute nodes, non-leaf nodes, auxiliary nodes and the sum of threshold in the access policy, respectively. And LCC is short for Lagrange Coefficient Computation.

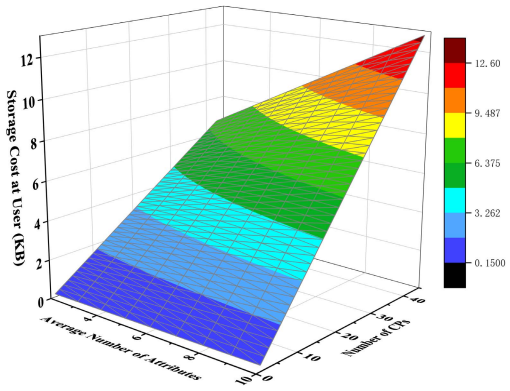


Fig. 7. Storage cost at user.

to communicate with CA or other AAs to determine the parameters, but the overhead of our scheme is not larger than Chase's schemes. In the key generation and distribution phase, our scheme brings a little extra communication overhead on AA and users compared with Chase's scheme-1 because of the extra auxiliary nodes generated in access structure transformation. But it is much less than Li's scheme and Chase's scheme-2. Besides, our scheme outperforms Li's scheme on communication overhead in the encryption/decryption phase on account of the smaller size of our ciphertext. In conclusion, the communication overhead in SKP-ABE is acceptable.

3) Computation Overhead Our proposed SKP-ABE introduces some auxiliary nodes in users' access structures, so it is inevitable to bring some extra computation overhead in key generation and decryption phases. The detailed comparison on computation overhead of key generation is illustrated in Table V.

SKP-ABE and Chase's scheme-2 have the same computation overhead of encryption. Compared with these two schemes, Chase's scheme-1 needs one less exponentiation operation, but Li's scheme performs worst, with extra $3n$ exponentiation and $2(N_A-1)$ multiplication on $\mathbb{G}$ and $2n$ multiplication on $\mathbb{Z}_p$.

Chase's scheme-1 has the least computation overhead in the decryption phase. The decryption in Chase's scheme-2 involves $n+1$ more Multiplication and one more exponentiation in $\mathbb{G}$. However, Li's scheme has a much higher computation overhead, since it is required to conduct 2 more pairing operations for each attribute.

TABLE VI
COMPUTATION COST IN CONTENT DEDUPLICATION

Entity	Operation	Time (ms)
Content Provider	Response Generation	≈ 0
	Tag Generation	1.344
	AES-256	0.004 (1KB)
	SKP-ABE (10 attributes)	16.134
Edge Router	Response Verification	2.694

In SKP-ABE, users are able to precompute some values associated with auxiliary nodes to decrease the computation overhead of decryption. They can compute

$$F'_x = \prod_{z \in \bar{S}_x} g^{q_z(0)\cdot \Delta_{z',S'_x}}$$

in advance for non-leaf node x, where $\bar{S}_x$ is the set of auxiliary child nodes of x. Upon receiving the ciphertext, they can get the correct F_x by computing

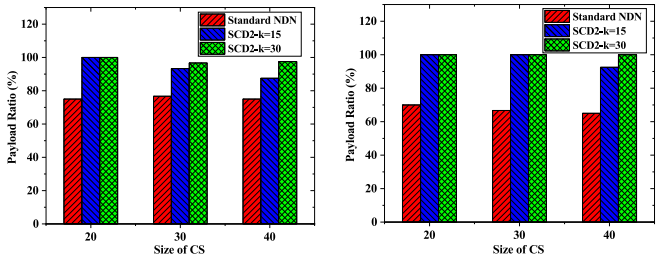
$$F_x = e(C_i, F'_x) \prod_{z \in S_x - \bar{S}_x} F_z^{\Delta_{z',S'_x}}.$$

Finally, users only need to conduct n_2 more times of multiplication in $\mathbb{Z}_p$ and exponentiation in $\mathbb{G}$ and one more time of pairing than Chase's scheme-1, where n_2 is the number of non-leaf nodes in users' access structures.

Then, we implement POW in content deduplication phase by using GNU Multiple Precision Arithmetic (GMP), Pairing-Based Cryptography (PBC), and OpenSSL libraries. The experiment is conducted on a Linux system (Ubuntu 16.04 LTS) with a 3.6GHz Intel Core i7 processor and 20G RAM. TABLE VI demonstrates the computation cost of each step in the challenge-response process. Note that the largest cost in the process is the encryption using SKP-ABE, which costs 16.134 ms with 10 attributes contained in the contents. Since encryption using AES-256 and SKP-ABE can be pre-computed before the POW, the content deduplication is efficient enough.

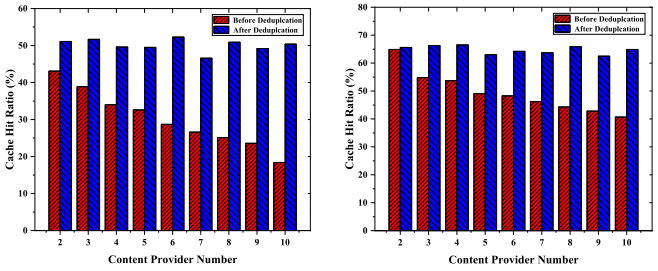
B. Performance in Simulations

Here, we implement SCD2 in a simulated named data network (NDN) environment using ndnSIM 2.3 [52], which is a typical type of ICN architecture. The network topologies are generated by using BRITE [53], where the number of user nodes is 20% of routers. The links between any two routers



(a) 2 CPs (b) 3 CPs

Fig. 8. Deduplication efficiency vs. CS size.



(a) (b)

Fig. 9. (a) Cache hit ratio vs. content provider number under 1000 MB cache size, (b) cache hit ratio vs. content provider number under 5000 MB cache size.

have randomly selected bandwidth and delay from 1 to 5 Gbps and 1 to 5 ms respectively. Users connect with edge routers through a link whose bandwidth is 100 Mbps and delay is 1 ms.

1) Content Deduplication Performance We first test the deduplication efficiency of our scheme, which is represented by payload ratio (i.e., the ratio of nonredundant content size to the full CS size). In this simulation, users can request contents from different randomly chosen CPs and their requests of each CP follow the Zipf popularity distribution. In other word, the probability of a user requesting the i-th popular content of a CP is $p_i = \frac{F(\alpha)}{i^\alpha}$, where $F(\alpha)$ is the normalization factor and α is the Zipf exponent. We consider that the i-th popular content, where $i\%2 = 0$, from each CP, is the same content and we only conduct content deduplication on k most popular contents. The cache policy in this simulation is LFU, α is set as 0.8, and each CP has 100 contents.

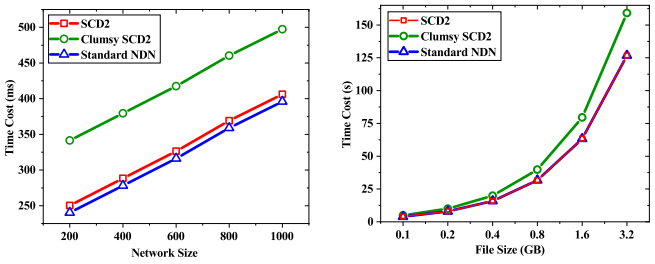
Fig. 8 shows the results of payload ratio in an intermediate router after users sent 60000 requests. As shown in Fig. 8(a), when there are 2 CPs, the payload ratio of standard NDN which has no content deduplication mechanism maintains around 75% with growing CS size. Although the payload ratio of SCD2 witnessed a downward trend as the CS size increases due to the increasing unpopular duplicated contents, SCD2 far outperforms standard NDN. And a higher chosen k value leads to a higher payload ratio because routers can delete more duplicated contents, but in the meantime, higher k also incurs more communication overhead. Thus, ISP should consider the tradeoff between deduplication efficiency and communication overhead and choose the proper k value. Similar results of the simulation where there are 3 CPs can be seen in Fig. 8(b). The payload ratio of standard NDN stays at about 65%, which is much worse than that of SCD2. SCD2's payload ratio decreases slower than that in the simulation which has 2 CPs because of the lower probability of duplicated unpopular contents. Therefore, we can conclude that when the number of CPs becomes huge, SCD2 can have high efficiency in content deduplication.

To show how content deduplication influences the network performance, we also test the cache hit ratios before and after content deduplication with different numbers of CPs. We let user nodes send 1000 interests in a second by following the Zipf popularity distribution. Each router node has the same fixed cache space. From Fig. 9, we can see that as the number of content providers increases, the cache hit ratio before deduplication decreases gradually, whereas the cache hit ratio of a network with duplicate content de-duplication keeps steady. When the cache size is 1000 MB, it improves about 32% for 10 CPs. At the same time, we note that when the node's cache space becomes larger, the gap between the performances before and after deduplication will be narrower. However, when the cache size becomes 5000 MB, the scenario with deduplication still performs better than the one without deduplication by 24%. It thus verifies that it is necessary to conduct content de-deduplication in ICN, and SCD2 can effectively improve the cache utilization in the network through content de-duplication.

2) Access Control Performance Then, we make comparisons between standard NDN, SCD2, and clumsy SCD2 in terms of content retrieval delay. Here CPs only use symmetric encryption to preserve confidentiality in standard NDN and in clumsy SCD2, edge routers authenticate all of the requests by challenge-response mechanism. In the simulations, users request continuous chunks of a file and they do not send the next requests until they receive the contents of current requests. Besides, users' access structures in different CPs are like Fig. 2 with 3 "AND" gates and one "OR" gate.

Specifically, we first evaluate the relation between content retrieval delay and network size. The chunk size is set to 1 MB and every file comprises 10 chunks in this simulation. As illustrated in Fig. 10(a), the content retrieval delay increases when network size grows in all three schemes. But compared with standard NDN, SCD2 only introduces a little extra overhead, which is about 10 ms, because of hash chain authentication, while clumsy SCD2 incurs 90 ms extra delay since it has to conduct complicated challenge-response authentication.

We also test the content retrieval delay with different file sizes ranging from 0.1 to 3.2 GB. The simulation is conducted on a network topology of 1000 nodes and the chunk size is also set to 1 MB. As we can see in Fig. 10(b), with growing file size, since there are more chunks to be transmitted, content retrieval delay also increases. SCD2 shows great performance that is nearly the same as the performance of standard NDN. Nevertheless, due to more requests that need to be authenticated, the disadvantage of using the challenge-response mechanism for every request is magnified with increasing file size. So the extra delay introduced in clumsy SCD2 increases from 1.013 to 32.416 s when file size varies from 0.1 to 3.2 GB.



(a) (b)

Fig. 10. (a) Content retrieval delay vs. network size, (b) content retrieval delay vs. file size.

According to aforementioned simulations, we conclude that SCD2 is highly efficient and effective in both content delivery and content deduplication.

IX. Conclusion

In this paper, we proposed a new content delivery scheme, named SCD2, to ensure secure content delivery and deduplication in ICN with multiple content providers. By adopting the SKP-ABE in which different CPs can share public attributes, our proposed scheme is able to achieve fine-grained access control with efficient key management. In SCD2, in addition to using SKP-ABE, we also designed an attribute-based challenge-response authentication mechanism at edge routers to thwart the potential interest flooding attacks. Furthermore, routers in ICN can conduct content deduplication for those popular contents by asking one of the CPs for some security parameters. Finally, through analysis on security and performance, we proved that SCD2 is secure enough and highly efficient with acceptable overhead.

References

[1] 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.

[2] M. F. Al-Naday, N. Thomos, and M. J. Reed, "Information-centric multilayer networking: Improving performance through an ICN/WDM architecture," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 83–97, Feb. 2017.

[3] S. Wang, J. Bi, J. Wu, and A. V. Vasilakos, "CPHR: In-network caching for information-centric networking with partitioning and hash-routing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2742–2755, Oct. 2016.

[4] J. Qin, K. Xue, J. Li, Q. Sun, and J. Lu, "Service prioritization in information centric networking with heterogeneous content providers," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 4, pp. 4476–4488, Dec. 2021.

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

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

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

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

[9] E. G. Abdallah, M. Zulkernine, and H. S. Hassanein, "Preventing unauthorized access in information centric networking," *Secur. Privacy*, vol. 1, no. 4, p. e33, Jul. 2018.

[10] L. Zhu *et al.*, "T-CAM: Time-based content access control mechanism for ICN subscription systems," *Future Gener. Comput. Syst.*, vol. 106, pp. 607–621, May 2020.

[11] 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.

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

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

[14] C. Bernardini, S. Marchal, M. R. Asghar, and B. Crispo, "PrivICN: Privacy-preserving content retrieval in information-centric networking," *Comput. Netw.*, vol. 149, pp. 13–28, Feb. 2019.

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

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

[17] Z. Wu, E. Xu, L. Liu, and M. Yue, "CHTDS: A CP-ABE access control scheme based on hash table and data segmentation in NDN," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 843–848.

[18] Y.-F. Tseng, C.-I. Fan, and C.-Y. Wu, "FGAC-NDN: Fine-grained access control for named data networks," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 143–152, Mar. 2019.

[19] S. Jiang, J. Liu, L. Wang, Y. Zhou, and Y. Fang, "ESAC: An efficient and secure access control scheme in vehicular named data networking," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10252–10263, Sep. 2020.

[20] 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.

[21] K. Xue, P. He, X. Zhang, Q. Xia, D. S. L. Wei, H. Yue, and F. Wu, "A secure, efficient, and accountable edge-based access control framework for information centric networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1220–1233, Jun. 2019.

[22] P. He, K. Xue, J. Yang, Q. Xia, J. Liu, and D. S. L. Wei, "FASE: Fine-grained accountable and space-efficient access control for multimedia content with in-network caching," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 4, pp. 4462–4475, Dec. 2021.

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

[24] T. Nguyen *et al.*, "Reliable detection of interest flooding attack in real deployment of named data networking," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 9, pp. 2470–2485, Sep. 2019.

[25] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 321–334.

[26] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, 2006, pp. 89–98.

[27] Z. Zhang *et al.*, "An overview of security support in named data networking," *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 62–68, Nov. 2018.

[28] 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.

[29] J. Li, K. Ren, and K. Kim, "A2BE: Accountable attribute-based encryption for abuse free access control," IACR Cryptol. ePrint Arch., Tech. Rep., 2019. [Online]. Available: http://eprint.iacr.org/2009/118

[30] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proc. 5th ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS)*, 2010, pp. 261–270.

[31] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 9, pp. 2546–2559, Sep. 2016.

[32] M. Chase, "Multi-authority attribute based encryption," in *Proc. Theory Cryptogr. Conf. (TCC)*. Amsterdam, The Netherlands: Springer, 2007, pp. 515–534.

[33] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS)*, 2009, pp. 121–130.

[34] Q. Li, J. Ma, R. Li, J. Xiong, and X. Liu, "Large universe decentralized key-policy attribute-based encryption," *Secur. Commun. Netw.*, vol. 8, no. 3, pp. 501–509, Feb. 2015.

[35] S. Dougherty, R. Tourani, G. Panwar, R. Vishwanathan, S. Misra, and S. Srikanteswara, "APECS: A distributed access control framework for pervasive edge computing services," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2021, pp. 1405–1420.

[36] S. Müller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," in *Proc. Int. Conf. Inf. Secur. Cryptol. (ICISC)*. Seoul, South Korea: Springer, 2008, pp. 20–36.

[37] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "DAC-MACS: Effective data access control for multiauthority cloud storage systems," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1790–1801, Aug. 2013.

[38] K. Yang, X. Jia, and K. Ren, "Secure and verifiable policy update outsourcing for big data access control in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 12, pp. 3461–3470, Dec. 2015.

[39] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," in *Proc. Int. Conf. Inf. Secur. (ISC)*. Pisa, Italy: Springer, 2009, pp. 347–362.

[40] 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 Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1484–1496, May 2016.

[41] K. Xue *et al.*, "RAAC: Robust and auditable access control with multiple attribute authorities for public cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 953–967, Apr. 2017.

[42] N. H. Sultan, V. Varadharajan, S. Camtepe, and S. Nepal, "An accountable access control scheme for hierarchical content in named data networks with revocation," in *Proc. Eur. Symp. Res. Comput. Secur. (ESORICS)*. Guildford, U.K.: Springer, 2020, pp. 569–590.

[43] N. H. Sultan, V. Varadharajan, C. Kumar, S. Camtepe, and S. Nepal, "A secure access and accountability framework for provisioning services in named data networks," in *Proc. 40th Int. Symp. Reliable Distrib. Syst. (SRDS)*, Sep. 2021, pp. 164–175.

[44] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (Eurocrypt)*. Athens, Greece: Springer, 2013, pp. 296–312.

[45] S. Keelveedhi, M. Bellare, and T. Ristenpart, "DupLESS: Server-aided encryption for deduplicated storage," in *Proc. 22nd USENIX Secur. Symp. (USENIX Security)*, 2013, pp. 179–194.

[46] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 6, pp. 1615–1625, Jun. 2014.

[47] T. Jiang, X. Chen, Q. Wu, J. Ma, W. Susilo, and W. Lou, "Secure and efficient cloud data deduplication with randomized tag," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 532–543, Mar. 2017.

[48] Z. Yan, M. Wang, Y. Li, and A. V. Vasilakos, "Encrypted data management with deduplication in cloud computing," *IEEE Cloud Comput.*, vol. 3, no. 2, pp. 28–35, Mar. 2016.

[49] H. Cui, R. H. Deng, Y. Li, and G. Wu, "Attribute-based storage supporting secure deduplication of encrypted data in cloud," *IEEE Trans. Big Data*, vol. 5, no. 3, pp. 330–342, Sep. 2019.

[50] D. Boneh and X. Boyen, "Efficient selective-ID secure identity-based encryption without random oracles," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. (Eurocrypt)*. Interlaken, Switzerland: Springer, 2004, pp. 223–238.

[51] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

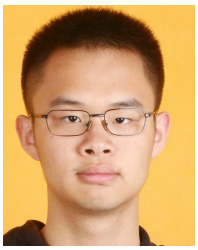
[52] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2.0: A new version of the NDN simulator for NS-3," UCLA, Los Angeles, CA, USA, Tech. Rep. NDN-0028, 2015. [Online]. Available: https://named-data.net/publications/techreports/ndn-0028-1-ndnsim-v2/

[53] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *Proc. 9th Int. Symp. Modeling, Anal. Simulation Comput. Telecommun. Syst. (MASCOTS)*, 2001, pp. 346–353.

Kaiping Xue (Senior Member, IEEE) received the bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003, and the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. From May 2012 to May 2013, he was a Post-Doctoral Researcher with the Department of Electrical and Computer Engineering, University of Florida. Currently, he is a Professor with the School of Cyber Science and Technology, USTC. His research interests include next-generation internet architecture design, transmission optimization, and network security. He is an IET Fellow. He serves on the Editorial Board of several journals, including the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT. He has also served as a (Lead) Guest Editor for many reputed journals/magazines, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, *IEEE Communications Magazine*, and *IEEE Network Magazine*.

Peixuan He received the bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2017, and the master's degree from the Department of Electronic Engineering and Information Science (EEIS), USTC. Currently, he is an Engineer with the Security Research Department, Bytedance Corporation, China. His research interests include network security protocol design and analysis.

Jiayu Yang (Graduate Student Member, IEEE) received the bachelor's degree in information security from the School of Cyber Science and Technology, University of Science and Technology of China (USTC), in 2019, where she is currently pursuing the Ph.D. degree with the School of Cyber Science and Technology. Her research interests include future internet architecture design, transmission optimization, and network security.



Qiudong Xia received the B.S. and master's degrees in information security from the School of Cyber Science and Technology, University of Science and Technology of China (USTC), in 2018 and 2021, respectively. His research interests include architecture design and security protection in ICN.

David S. L. Wei (Senior Member, IEEE) received the Ph.D. degree in computer and information science from the University of Pennsylvania in 1991. From May 1993 to August 1997, he was an Associate Professor and then a Professor with the Faculty of Computer Science and Engineering, The University of Aizu, Japan. He is currently a Professor with the Computer and Information Science Department, Fordham University. He has authored and coauthored more than 130 technical papers in various archival journals and conference proceedings. His current research interests include cloud and mobile edge computing, cyber security, quantum engineering, and machine learning and its applications. He was a Lead Guest Editor or a Guest Editor for several special issues in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, and the IEEE TRANSACTIONS ON BIG DATA. He also served as an Associate Editor for *Journal of Circuits, Systems and Computers* (2013–2018), IEEE TRANSACTIONS ON CLOUD COMPUTING (2014–2018), and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS for the Series on Network Softwarization and Enablers (2018–2020).