CSEVP: A Collaborative, Secure, and Efficient Content Validation Protection Framework for Information Centric Networking

Kaiping Xue, *Senior Member, IEEE*, Jiayu Yang, *Graduate Student Member, IEEE*, Qiudong Xia,
David S. L. Wei, *Senior Member, IEEE*, Jian Li, *Member, IEEE*,
Qibin Sun, *Fellow, IEEE*, and Jun Lu

***Abstract*—As a new architecture of Internet infrastructure, Information-Centric Networking (ICN) is mainly designed to effectively handle the rapidly increasing user demand for content delivery through in-network caching. While facilitating the dissemination of content to users and making better use of the network resources, ICN is also vulnerable in that attackers can inject poisoned content into the network and isolate users from valid content sources. The introduction of signature verification in each router can effectively prevent this attack, but it also introduces great computation overhead. Existing schemes in ICN reduce verification overhead from a single routing perspective but do not consider integrating resources within ICN for collaborative content authentication and cyber self-defense. In this paper, we propose a collaborative, secure, and efficient content validation protection framework, named CSEVP, to implement a multi-router collaborative defense mechanism for ICN. On the one hand, we conduct content verification by probabilistically choosing one router involved in the transmission path to offload the computation overhead of content verification from a single router to multiple ones. On the other hand, we adopt bloom filters for routers to record and share verification results to further facilitate a more efficient content validity verification. The security and efficiency analysis shows that our proposed CSEVP can achieve efficient content validity verification among multiple routers with acceptable low communication and storage overhead.**

***Index Terms*—Information centric networking, content poisoning attack, validity verification, authentication.**

Manuscript received April 22, 2021; revised September 17, 2021 and November 27, 2021; accepted November 30, 2021. Date of publication December 20, 2021; date of current version June 10, 2022. This work is supported in part by the National Natural Science Foundation of China under Grants No. 61972371 and No. U19B2033, and Youth Innovation Promotion Association of Chinese Academy of Sciences (CAS) under Grant No. Y202093. The associate editor coordinating the review of this article and approving it for publication was K. Shiomoto. *(Corresponding authors: Kaiping Xue; Jian Li.)*

Kaiping Xue, Jiayu Yang, Qiudong Xia, Jian Li, and Qibin Sun are with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230027, Anhui, China (e-mail: kpxue@ustc.edu.cn; lijian9@ustc.edu.cn).

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

Jun Lu is with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, Anhui, China.

Digital Object Identifier 10.1109/TNSM.2021.3136547

I. Introduction

INFORMATION-CENTRIC Networking (ICN) is proposed as a next-generation network architecture to cope with contradictions between the current limited bandwidth of IP networks and growing users' demands for content delivery [1]–[4]. In ICN, the network locates and retrieves contents by content names rather than network addresses of contents, which shifts the network attention from *where* contents are to *what* users want. Also, all intermediate routers can cache contents and respond to users' requests directly. Any neighboring intermediate router satisfies users' interests with contents' names, which leverages the network's existing resources, e.g., bandwidth and routers' cache space, and delivers contents to users with lower latency.

However, the above advantages of ICN also pose new challenges to its security [5]. One typical challenge is *Content Poisoning Attack* (CPA). Since intermediate routers can independently select and cache contents when they transmit, unverified and contaminated contents have the opportunity to be cached by routers and remain in the network during transmission, which affects the validity of contents in the ICN network [6]. With this vulnerability, attackers can launch CPA attacks by pretending to be a content provider and injecting the poisoned contents into the cache of the routers. These poisoned contents can be unknowingly spread in the network by ICN itself. The spread of poisoned contents could potentially exhaust a lot of network cache resources and isolate users from valid contents.

To cope with this security problem, the standard ICN leverages digital signature algorithms to protect content validity [7]. When an intermediate router faces unverified content, it conducts signature verification to verify the validity of the content. However, the computation overhead of asymmetric cryptography is too high. If a router receives massive unverified contents in a very short time, it cannot bear such heavy signature verification overhead. Researchers thus further proposed some practical solutions from two aspects: One category considers optimizing the aforementioned signature-based authentication scheme, e.g., [8]–[10], and the other category considers proposing an alternative more feasible mechanism to replace the signature-based authentication scheme, e.g., [11].

To optimize the existing signature-based authentication scheme in [7], Gasti *et al.* [8] proposed to verify content validity probabilistically through using the only HMAC values of content instead of signatures, which can significantly reduce the verification overhead. But the new authentication scheme also brings new security vulnerabilities. In Gasti *et al.*'s scheme [8], all routers use the same key to calculate HMAC values which leads to the situation that the adversary can easily launch a successful attack, as the adversary just needs to learn one key.

Some other non-signature-based authentication schemes handle contaminated content in the cache according to users' authentication feedback. Ghali *et al.* [11] used a lightweight ranking algorithm as a measure to mitigate content poisoning. Their scheme ranks content based on user feedback and selects the highest-ranked content to return to users, thereby avoiding the transmission of contaminated content in the network. However, these schemes are at risk of excluding valid content based upon the forged feedback for users.

A feasible solution to this problem is via the cooperation among routers in the ICN network to verify contents [8]. It's not hard to see that sharing contents' authentication results can effectively reduce unnecessarily repeated verifications. The router that has verified the content can share the authentication results with others. When the same content reaches other routers, they can quickly check the validity of the content based on the shared verification results with no need of frequent cryptographic operations. Besides, since large amounts of duplicates of popular contents are stored in different routers, sharing verification information can avoid unnecessary authentication of these duplicates. Furthermore, redistributing verification tasks among multiple routers can make good use of the network's computing resources. The hit router performs content verification, and other routers responsible for content transmission don't perform this operation [12]. When a single router faces numerous authentication requirements, computing resources of other routers cannot be leveraged even if the responsible router cannot afford all the tasks. A fair verification task allocation scheme is thus needed to allocate verification tasks evenly to intermediate routers to achieve unified utilization of computing resources.

Motivated by the above observation, we present a collaborative, secure, and efficient content validation protection framework, called CSEVP, for ICN. We implement a mechanism for recording and sharing verification results among multiple nodes based on bloom filters to achieve collaborative authentication among multiple routers. It can record and query verified content signature information efficiently with lower storage overhead. Also, because of the merging feature of the bloom filter, routers can easily exchange verification information by sharing and merging their bloom filters with others. To evenly utilize computation resources among routers, we propose a probabilistic disjoint verification method to ensure that the content is verified once before it reaches the user side, and content verification tasks will be fairly assigned to the routers participating in content transmission. In addition, we have established a traceability mechanism for the source of pollution contents based on edge routers. The edge router records the source information of each content entering the network and marks it. When an intermediate router found the pollution content through inspection, it will trace the pollution source and impose a penalty according to the edge marking. Our contributions can be summarized as follows:

- We propose a probabilistic disjoint verification protocol based on multi-router collaboration. All contents will be verified only once before reaching the user, and the verification overhead is evenly distributed to each router along the path.
- We design an efficient and lightweight mechanism for recording and sharing verification results with bloom filters. A single router could record signatures of valid contents in a bloom filter and shares the verification information through the exchange of bloom filters among neighboring routers. It reduces verification overhead remarkably.
- We have established a mechanism for tracing pollution content relying on edge routers. The edge router marks every content in the ICN network, and when the content is deemed to be contaminated, the source can be quickly traced and be published to prevent continued attacks from pollution sources.
- We formally analyze the security strength and conduct experiments by algorithm implementation and network simulation. The experiment results show that our scheme defenses content poison attacks in ICN effectively and efficiently.

The rest of the paper is organized as follows. In Section II we discuss some related work. We present our system model and security assumptions in Section III, and state some preliminaries in Section IV. The details of our schemes are presented in Section V. Then Section VI and Section VII show the security and performance analysis, respectively. Finally, we conclude our work in Section VIII.

II. Related Work

In-Network Caching, as the primary element of ICN, performs well in accelerating content transmission. However, caching capability also triggers security issues that need to be addressed. Access control is one of them, which can be solved through public-key facilities [13], certificateless group signature [14], broadcast encryption [15], or by new technologies such as blockchain [16]. Also, the data privacy issues have to be addressed when caching data at intermediate routers. Li *et al.* [17], and Wang *et al.* [18] proposed flow-based and session-based control mechanisms to prevent consumer privacy leakage, respectively. In addition, contaminated content attacks related to content storage and sharing have become a new security issue [19], [20]. In general, there are two types of security attacks via the network cache, of which one is cache pollution and the other is content poisoning.

Cache Pollution Attack [21] uses numerous unpopular contents to occupy caches, which aims to reduce the cache utilization rate in the network. Xie *et al.* [22] introduced CacheShield to judge this attack by analyzing the popularity distribution of contents in the network cache. Yao *et al.* [23] effectively exploited the regularity of past Interests and popularity to predict the future popularity of each cached content with the help of grey forecast. It detects content pollution and effectively avoids the cache encroaching by non-popular content. Machine learning and statistical methods are also practical methods to thwart this attack. Nguyen *et al.* [24] used clustering and Bayesian analysis modes to identify CFA attacks that are occurring in the network with learning methods. And [25] leveraged the hypothesis testing theory to develop a generalized likelihood ratio test adapted to evolve IFA attacks. Also, [26] used network coding to discriminate between honest and malicious nodes and isolate the malicious ones.

In this paper, we focus our discussion on Content Poisoning Attack (CPA) [5]. In a CPA, the contaminated contents are transferred to the ICN network by the attacker and stored in the routers' content store (CS). These contaminated contents with correct content names can match the user's interest requests, and are distributed throughout the ICN network. When interest requests sent by users with the corresponding content name arrive at these routers, the contaminated content can be matched and delivered to users, and some intermediate routers may store them in their CS. This attack can prevent users from obtaining legitimate source content and may cause secondary harm to users, as the polluted contents occupy a number of cache resources. Since the user needs to request the correct content again after receiving the polluted content, the retransmission process will also consume a lot of network transmission resources.

Currently, many schemes defend the system against this attack through content confirmation at intermediate nodes by adding a digital signature to each transmitted content [27]. However, according to [11], signature verification consumes a lot of computing resources. A lot of work aims to solve the problem of the excessive overhead of signature verification and find a more suitable solution to resist the content poisoning attack. We can roughly divide the existing solutions into three categories.

The first approach, [8], [10], [11], [28], [29] judged the legitimacy of content through users' feedback. In [8] and [11], the authors discussed a self-certifying name whose last component is the hash value of the content. In terms of the hash value embedded in the interest, the valid content is identified without performing signature verification. Since the hash value of dynamically-generated content cannot be created and informed a priori, this approach has a limited application to static content. In [10], [28], [29], feedback is secured by the signature signed by network constituents. However, attackers may issue a massive amount of feedbacks, causing routers' computational resources to be exhausted by verification. The above solutions are based on the user's credible conditions for security analysis. In fact, in most cases, users are unreliable. They are likely to collude with attackers and feedback wrong information to the network, which greatly reduces the system's credibility.

The second method aims at finding a way to replace the digital signature and bind other verification information to the content to verify the content. References [9], [30]–[33] adopted this method. For example, in [9], the authors suggested the Interest-Key Binding (IKB) rule, which adds a bond between the content name and the provider's public key. In the scheme, a user obtains the provider's public key before issuing an interest for the content and embeds its digest (PPKD) in the interest. Since each piece of content also carries the public key, routers match the hash value of the public key with PPKD in the PIT entry. If they do not match, the content is discarded. However, the computational overhead generated by this scheme is still huge. When numerous contents need to be verified, the node still cannot lower the computational overhead caused by the scheme. Yang *et al.* [34] proposed a proactive reputation-based scheme, which selects the next-hop router for interest packets probabilistically based on reputation to isolate attackers. However, nodes with a high reputation still may be malicious, and it can not guarantee that content is secure when reaching users. Li *et al.* [35] proposed a security architecture based on capabilities that specify the access rights of forwarding packets. They also provided a one-time signature scheme that leverages the standard hash function to reduce the verification overhead. However, they still require verification at each router to enable content to be secure when reaching users.

The last scheme uses reasonable screening to reduce unnecessary signature verifications [36]–[38]. Based on the original ICN's content validity verification architecture, these papers achieved optimization of content validity verification by reducing the total number of signature verifications. Reference [37] and its preliminary version of this paper [38] showed an optimization method. The author proposed two core schemes to optimize the efficiency of content verification. The former scheme will only verify the content when it is hit to avoid unnecessary detection of non-popular content. And the latter divides the CS table into two parts. One is a temporary storage table, called non-SLRU, and the other is a long-term storage table, called SLRU. The verified content is stored in SLRU and newly cached content will be stored in non-SLRU, without any verification. In this way, the scheme avoids repeated detection of duplicate contents. However, the first batch of content that enters the network was not verified. Some users requesting unpopular content may still receive contaminated content from the ICN.

Some schemes utilize machine learning technologies. Zhou *et al.* [39] utilized reinforcement learning to avoid content pollution by deciding whether a data packet is to be cached. However, the training process is complex, and the computational overhead of using the model at each router is still too high.

Compared with the work mentioned above, our scheme utilizes a collaborative method, which has been used in caching to increase the overall hit ratio [4]. In our scheme, all of the intermediate nodes participate in the content verification task. They utilize a probabilistic protocol to reduce verification expenses and share the authentication results through bloom filter to speed up the authentication process. Therefore, it can perform content verification efficiently and ensure that the content is secure when it reaches users.

III. System Model, Threat Assumptions and Preliminaries

A. System Model

In our scheme, we consider an ICN network consisting of three parts: routers managed by an Internet Service Provider

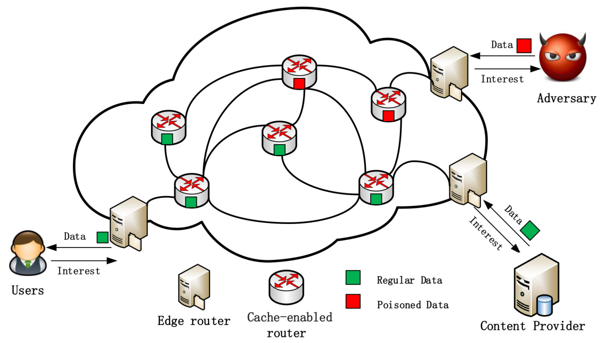


Fig. 1. System Model.

(ISP), some content servers maintained by Content Providers (CPs), and a lot of users, which are elaborated as follows:

- *Routers Managed by an Internet Service Provider:* It provides ICN network service to CPs and users. With the help of an in-network cache, it offers an efficient content distribution. Besides, it is also responsible for verifying contents delivered over the network and eliminating poisoned contents in time to prevent them from being transmitted. The routers in ICN network can be divided into two types: edge routers and cache-enabled routers. Cache-enabled routers forward interest packets and respond to the request if they cache the requested contents. Also, they verify passing content packets at a certain probability and exchange the verification information with each other to speed up the subsequent content verification process. Edge routers simply authenticate contents sent by content providers whether or not contents have all the required information to verify the validity of the content. To indicate the source of the poisoned contents, all contents from CPs will pass through the edge router before being posted to the network. Edge routers label them to indicate from which edge router the content enters the network for retroactive punishment.
- *Content Servers Maintained by Content Providers:* Like YouTube and Netflix, they produce content and deliver them through the ICN network. To facilitate intermediate routers to check the validity of the content, CPs will combine signature and other verification information with contents before publishing. However, some adversaries in the ICN network are assumed to implement content pollution attacks, in which they pretend to be fake content providers and respond to the interest packets transmitted by the edge routers and return polluted content to implement the attack.
- *Users:* They are the consumers of the content and obtain desired contents from CPs or ICN networks.

B. Security Assumptions

In our scheme, routers owned by ISPs are responsible for delivering content, detecting, and eliminating polluted contents in the network. We mainly focus on addressing the content poisoning attack implemented by end-hosts and assume that all the routers trust each other. This assumption is reasonable, as it's challenging to attack intermediate routers by adversaries in real network scenarios, and implementing a content contamination attack directly on the router is also very hard. This security assumption is also given in some other related literatures, e.g., Kim *et al.*'s scheme [37](“Kim Scheme” for short), ABE [40], and LASA [41]. We also consider the scenario that routers don't trust each other, and at the end of Section V, we provide a trust mechanism as an extension of the proposed scheme to address this situation.

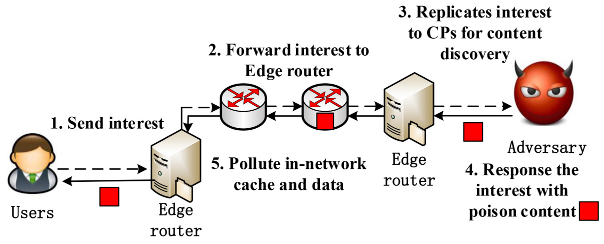

3. Replicates interest to CPs for content discovery
2. Forward interest to Edge router
1. Send interest
5. Pollute in-network cache and data
Edge router
Adversary
4. Response the interest with poison content
Users
Edge router

Fig. 2. Content Poisoning by Interest Replication.

CPs, owners of contents, are assumed to be untrusted. Legitimate CPs respond to the request from the ICN network and send contents through edge routers. However, some adversaries masquerade as content providers and inject poisoned contents into the network when taking the opportunity to respond to requests. It is hard to estimate whether a CP is normal or harmful until all contents received from this CP have been fully inspected. Thus, CPs cannot be trusted by ISPs and routers should verify contents from everywhere.

C. Design Goals

Our scheme is designed to effectively detect *Content Poison Attacks* and securely prevent the ICN from attacks.

As mentioned above, in CPA, poison contents are placed in the in-network cache. Users will obtain contaminated contents when their requests contain the same name as the poisoned contents. Users' experience becomes terrible when ICN cache contains poisoned contents caused by content poison attacks.

Fig. 2 shows how an adversary puts poisoned contents into the ICN network. When a user requests nonexistent content from the ICN network, the edge router forwards the interest packet to CPs who own the requested content. Under certain circumstances, the interest packet will be caught by an adversary masqueraded as a content provider, and it will respond user's request with poisoned content as soon as possible. If the poisoned content arrives at the edge router first without any validity check, it flows into the ICN network and contaminates the in-network caches. The edge router will drop the correct contents from normal content providers for lacing a pending interest table.

For our design goals, our scheme should observe the following rules:

Security: To protect ICN network from the damage caused by CPA, all poisoned contents must be detected and be removed from the network cache before they reach users' sides. Also, the scheme needs to punish the attacker who initiates the CPA and prevents the polluted contents produced by the attackers from entering the network again.

Efficiency: To ensure efficient content validation, the scheme needs to maximize the use of resources in the network for content verification computations. That means it needs to avoid redundant verification calculations as much as possible while fully and evenly utilizing the computing resources of each router.

IV. Preliminaries

A. Digital Signature

The system uses a public-key signature scheme for content integrity verification. Under the assumption of secure distribution of public keys, any router can verify the content integrity. For conciseness of signatures, we use ECDSA [42]:

Syntax SIG for the security parameter $\lambda \in \mathbb{N}$ and any arbitrary message $m \in \{0,1\}^{n(\lambda)}$, where $n(\lambda)$ is a polynomial-bounded function consisting of three PPT algorithms $SIG = (Gen, Sign, Verify)$.

- $(vk_i, sk_i) \leftarrow Gen(1^\lambda)$ outputs a signing key sk_i and the corresponding verifying key vk_i.
- $s \leftarrow Sign(sk_i, m)$ generates a digital signature for the message m.
- $b \in \{0,1\} \leftarrow Verify(vk_i, s, m)$ outputs whether s is a valid signature of message *m*.

The digital signature scheme satisfies the existential unforgeability of signatures.

B. Bloom Filter

Bloom filter is an *m*-bit sequence for membership test with the features of reasonably accurate and space-efficient [43]. The bloom filter *BF* of *m*-bit for strings in $\{0,1\}^{poly(\lambda)}$ is as follows.

- $bf \leftarrow Setup(m, \lambda)$ generates an empty *m*-bit bit array.
- $bf' \leftarrow Insert(bf, e)$ inserts an element *e* by setting the following *l* positions of *bf* to 1: $H(k, 1||e)$, $H(k, 2||e)$,..., $H(k, l||e)$, where $H(k, \cdot)$ is a keyed collision-resistant hash function and *k* is a security parameter.
- $b \leftarrow Test(bf, e)$ checks whether the element *e* has been inserted to the bloom filter by checking whether all of these positions $H(k, 1||e)$, $H(k, 2||e)$,..., $H(k, l||e)$ are 1.

Bloom filter has a certain false positive rate that bloom filter tells that a coexist element is in the set. According to [43], [44], the false positive rate of an *m*-bit bloom filter is:

$$fp \approx \left(1 - \left(1 - \frac{1}{m}\right)^{ln}\right)^{l},$$

where *n* is the number of existing members in a set, and *l* is the number of hash functions used in the bloom filter.

C. Zipf-Mandelbrot Distribution

In our scheme, all arriving contents conform to Zipf-Mandelbrot distribution function with the parameters s and q according to [45]. In the Zipf-Mandelbrot distribution, the probability that the *i*-th popular content among the *N* content appears is denoted by:

$$P_N(i) = \frac{\Omega}{(i+q)^s},$$

where

$$\Omega = \left(\sum_{i=1}^{N} \frac{1}{(i+q)^s}\right)^{-1}.$$

The values of parameters *s* and *q* depend on the type of content being transmitted on the network, such as Web transmission and video data transmission. According to [45] and [46], The value of *s* varies from 0.7 to 1.3 in order to simulate various traffic patterns including Web traffic and video-on-demand services and the value of *q* often equals 0.7.

V. Proposed Scheme

A. Overview

In our scheme, intermediate routers verify contents with signatures included in content packets. In order to avoid the huge overhead caused by asymmetric signature authentication, we have adopted two main approaches to reduce signature verification overhead.

On the one hand, our scheme introduces the bloom filter for recording and sharing verification results. In particular, routers can rapidly determine content's validity by querying the bloom filter with the signature of valid contents. Meanwhile, every router can periodically share their bloom filters with their neighboring routers. By transmitting and merging bloom filters, routers can propagate verification information with each other with a lower communication overhead.

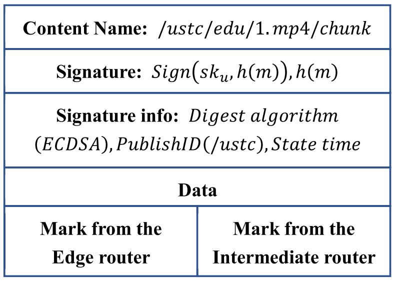
On the other hand, by implementing a probabilistic disjoint verification protocol, the scheme achieves a fair assignment of authentication tasks. In detail, each router on the way back to the user has the same probability to ensure that every content is verified once and is valid before reaching users. When the content's validity is confirmed with an intermediate router, subsequent routers trust the verification results and directly transmit content without any further operations.

We also propose mechanisms for tracing and punishing adversaries. Each content carries the marker of the edge router from which it enters the network. When an intermediate router detects invalid content, it traces the label marked to find content access records at the edge router.

B. Content Punishment

This process includes three steps:

1) *CP Registration:* Before publishing contents into the ICN network, a legitimate CP should register to a trusted third-party central authority (CA) for his/her identity and certificate. We assume that CA chooses a private key *sk* for a legitimate CP with a content name prefixed with '/school' and computes corresponding public key *vk*. Furthermore, CA generates a certificate *Cert* for the CP, which includes CP's public key and its contents name prefix, and sends it to the CP.
2) *Preparation for Contents Chunk:* After receiving the certificate from CA, CP will sign the disseminated content



Content Name: $/ustc/edu/1.mp4/chunk$	
Signature: $Sign(sk_u, h(m)), h(m)$	
Signature info: $Digest\ algorithm\ (ECDSA), PublishID(/ustc), State\ time$	
Data	
Mark from the Edge router	**Mark from the Intermediate router**

Fig. 3. A Legitimate Chunk with Verification Information.

with its signature key. As shown in Fig. 3, a legitimate content chunk should include two parts: necessary verification information and authentication annotation marked by the ICN network. First, legitimate content verification information should include a content name, a signature, and other signature-related information. Given chunk m with the name '/school/edu/1.mp4/chunk1', then the name and the signature $hash(m), s_m$, where $s_m = Sign(sk, h(m))$, should be contained in content chunks. The CP also needs to provide some signature-related information consisting of publishing key vk and the algorithm used for signatures because it is convenient and necessary for intermediate routers to select the appropriate algorithm with the right key to verify the signature. Second, there are two annotations in the chunk that come from the edge router where the content enters the network and the intermediate router where the content is verified. The annotations from the edge router whose content enters the network help networks trace adversaries releasing contaminated content and enforce the punishment mechanism. And the annotation from the intermediate router indicates whether the content is valid or not.

3) *Publishing Contents:* When the requests from ICN network arrive, the CPs will publish content consisting of a series of chunks to the edge router with its certificate. When the edge router receives these content chunks, it verifies the validity of the certificate and confirms that whether the signature of the content packet is valid. If so, the edge router marks its information in the reserved fields of the packet for tracing the content and forwards chunks into the ICN network. Otherwise, it discards them.

C. Contents Authentication Process

When unverified content is published by a CP and enters the ICN network through an edge router, the ICN network first chooses an intermediate router to authenticate the content via a probabilistic verification protocol. The selected intermediate router searches the stored bloom filter containing the verification results verified both by itself and other routers for fast verification. Otherwise, it verifies the content and stores the results of confirmed contents in a bloom filter for sharing and reuse. In the next two subsections, we will describe the two methods in detail.

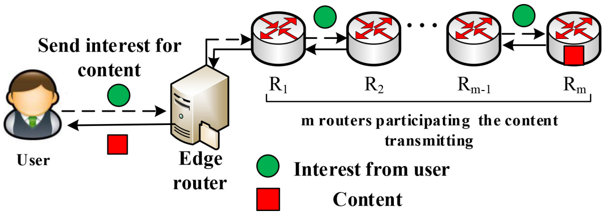

Send interest for content
User
Edge router
R_1
R_2
R_{m-1}
R_m
m routers participating the content transmitting
Interest from user
Content

Fig. 4. Probabilistic Verification Protocol.

Algorithm 1: Probabilistic Verification Protocol

```
Input: The content to be verified, $m$,
The mark from intermediate router of the content, $mark$,
The number of the participating routers, $k$.
if $mark$ is valid then
    Forward the content $m$ without any verification
end
else
    Compute the probability to verify the content:
    $P_k = \frac{1}{n-(k-1)}$,
    Probabilistic choice of whether to verify content,
    if Choice to verify then
        if $Verify(m)$ then
            Mark the content $m$ to be valid,
            Forward the content $m$.
        end
        else
            Drop $m$,
            Tracing the content $m$ with the mark from the
            edge router,
            Resend the interest of the content $m$ with its
            name.
        end
    end
end
```

1) Probabilistic Verification Protocol: In this protocol, intermediate routers on the reply content path will cooperate to perform a probabilistic disjoint verification and guarantee that before the content reaches the user, it must be verified once with the same probability for each router.

To achieve an adequate probabilistic verification, we utilize interest packets to record the number of hops routed to the hit router or CP in the ICN and calculate the probability of each router. We define n as the number of hops included in the interest packet, and this number also represents the number of intermediate routers participating in the probabilistic verification. As shown in Fig. 4, for the reply content from a CP through the edge router, n is counted from the first intermediate router transmitting the interest packet. We don't consider a scenario in which the interest is satisfied at the intermediate router because all contents cached by routers are marked as verified.

We define the set of all participating routers as $R_1, R_2, \ldots, R_n$. Each intermediate router follows the verification protocol shown in Algorithm 1 to authenticate all transmitted contents. When R_1 receives a content, it verifies

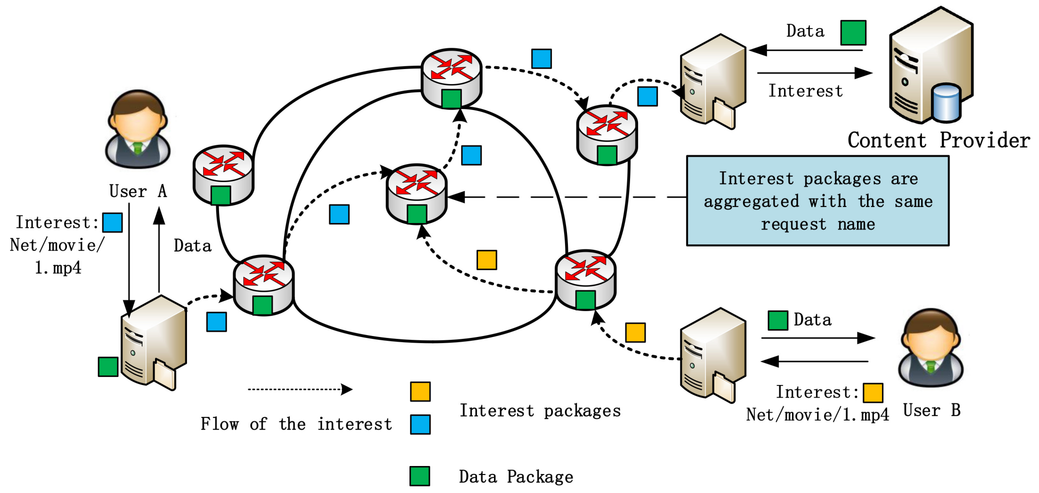

Data
Interest
Content Provider
Interest packages are aggregated with the same request name
User A
Interest: Net/movie/1.mp4
Data
Data
Interest: Net/movie/1.mp4
User B
Flow of the interest
Interest packages
Data Package

Fig. 5. The situation about Interests Aggregating.

the content with the probability of $P_1 = \frac{1}{n}$. If it determines to verify this content and confirms that it is valid, it marks the content with a valid label. Subsequent routers will forward the content with this mark without any check. If R_1 chooses to skip this content and directly transmit it to the next router with PIT, R_2 will face the same decision whether to verify the content or not. Similarly, if none of the previous routers has verified the content, the verification probability of R_k is

$$P_k = \frac{1}{n-(k-1)}. \tag{1}$$

We can prove that the probability of each intermediate router verifying the content is the same: $\frac{1}{n}$, and it must be verified once before reaching the user. The function $Verify(m)$ is the algorithm used by intermediate routers to verify the validity of the content.

In the above scenario, we discuss the case where content is replied from a content provider when it satisfies a single interest packet. In this situation, the data packet returns along the opposite path of its interest packet.

However, when multiple interest packets requesting the same content meet at the same intermediate router, the intermediate router aggregates these interest packets and forwards only one interest packet, making the path of reply content different from the interest packet's forwarding paths. For example, as shown in Fig. 5, two users who request the same content and their interest packets arrive at an intermediate router. The intermediate router forwards the first arriving interest packet and aggregates subsequent interest packets that request the same data into the PIT table. When the content arrives at the router where the aggregation occurred, it is sent to users one by one with corresponding requests according to the PIT. In our scheme, the routers participating in the adequate probabilistic verification are all intermediate routers of the interest packet pathway. As shown in Fig. 5, if the previous router hasn't verified the content, we recalculate the verification probability and select a router from the path between the aggregation router and the corresponding user. Each aggregated interest packet is sent from that interest packet to the aggregated router, and all intermediate routers perform fair probabilistic verification.

2) Verification and Sharing Results Based on Bloom Filter: When an intermediate node authenticates transmitted content, it uses a bloom filter for quick verification. The following parts describe the initialization of the bloom filter, its use, and the sharing of authentication information based on the bloom filter. The process can be divided into the following steps:

a) *Initialization of bloom filter:* Each intermediate router needs to build a bloom filter before they verify any content. The operator $bf \leftarrow BF.setup(m, \lambda)$ is defined as an operation that a router initiates a bloom filter with the size m and λ hash functions. However, considering a certain false positive rate of bloom filter, the ISP needs to set an upper limit of the false positive rate for the router's bloom filter to ensure considerable effectiveness. We define the false positive rate as α. The selection of initial parameters is related to specific ICN network-related parameters, including the size of the network and the traffic of individual routers. In performance analysis, we will combine experiments and inferences to give the best choice of parameters.

b) *Verification based on bloom filter:* For legal content, it will be marked with the corresponding tag to identify that it has been verified, and the content name and corresponding hash value will be stored in the bloom filter. When an intermediate router is chosen to verify the content, it will first query whether the content is in its bloom filters. We present the process of verifying content with the bloom filter in Algorithm 2. As the bloom filter records each verified content's signature (hash value) to indicate that the content has been verified and is legal, the intermediate router can efficiently determine validated content by fast hash lookup with $Test(bf, s_m)$. Otherwise, the content is needed to be verified with signature verification and hash verification if there is no corresponding signature stored in the bloom filter. Specifically, the intermediate router will verify the signature included in the content

Algorithm 2: Verification With Bloom Filter

```
Input: The content to be verified, $m$,
The signature of the content $m$, $s_m$,
The public key of the content, $vk_u$,
The verification bloom filter, bf,
The mark from intermediate router of the content, mark.
Output: Vaild or Invalid
if $Test(bf, s_m)$ then
    Mark the content is Valid,
    Return Valid.
end
else
    if $Verify(vk_i, s_m, hash(m))$ then
        $Insert(bf, s_m)$,
        Return Valid.
    end
    else
        Return Invalid.
    end
end
```

with $Verify(vk_i, s_m, hash(m))$. And then it utilizes the defined operator $Insert(bf, s_m)$ to add the signature of valid content into *bf*.

c) *Share verification information with bloom filter:* Routers share *bf* regularly. When it reaches the sharing period specified by the ISP, each intermediate router will share the *bf* that it is currently using and constantly updating to its neighboring routers. When a router receives a *bf* from another router, it will perform the operation of XOR to manipulate the bloom filter with its own bloom filter bits to merge the bloom filters. Assume that there are two bloom filters waiting to merge: bf_1 and bf_2, and the combined bloom filter $bf = bf_1 \vee bf_2$. It is simple to prove that:

$$\forall \alpha \in bf_1 \rightarrow \alpha \in bf \tag{2}$$

$$\forall \alpha \in bf_2 \rightarrow \alpha \in bf. \tag{3}$$

When the combined *bf* reaches or exceeds the upper bound of the false positive rate, the router needs to update the *bf*.

d) *Update the bloom filter:* As the number of stored signatures grows, the false positive rate of the aggregated bloom filter also gradually increases. To ensure accuracy, a new bloom filter should be generated when the false positive rate of the bloom filter reaches the preset threshold. And the old version bloom filter will continue to be stored in the router. When receiving subsequent contents, the intermediate node will first retrieve the signature from the current bloom filter. If there is no match, it checks all the saved old bloom filters in order from the newest to the oldest. Historical bloom filters stored too long can be removed from the node.

D. Tracing and Punishment Mechanism

Tracking of poison content is based on authentication and marking of the content by the edge router. As we mentioned in the content publishing section, the edge router performs a simple verification of the integrity of the content before it enters the ICN network via the content provider. This step ensures that the content contains complete validation information for intermediate routers to facilitate validation. After qualifying a simple content authentication, the edge router records the content's faceID and the corresponding CP's publishing ID in a table for retroactive accountability.

We can refer to [47] to record different content providers at the edge. Specifically, a legitimate provider must register its credentials with an edge identification and securely obtain an authentication tag. The edge attaches the tag, $T_p =< Pub_e, Pub_p, AP_p, T_e >$, where Pub_e and Pub_p denote the public key locator of edge and provider, respectively, AP_p is the provider's access path, and T_e denotes an expiry time, to data packets. Thus, we can locate the edge node and add the corresponding provider to the blacklist when the intermediate node detects the contaminated content.

When the content is identified as contaminated, the intermediate route is traced to the edge router accessed by the adversary via an edge router mark. The edge router tracks the access face to the adversary based on the table and discovers the specific attacker. Subsequent contents sent by the adversary are denied access to the network using a blacklist to prevent further content poison attacks.

E. Trust Mechanism

As an extension of the previous scheme, we introduce a simple trust mechanism to ensure that our scheme can still verify the authenticity of the content and reduce the verification overhead in the scenario with untrusted routers.

We suppose that each router maintains a trust value for each neighboring node in the network. In the initial state, all routers do not trust each other and verify the content transferred from neighboring nodes with signature. Following the transmission process, each node updates the trust value of their neighboring nodes according to the correctness of the content verification results. They increase the trust value of a neighboring node if the content delivered from this node passes the content's authenticity verification. Otherwise, they decrease the trust value of this node. When a neighboring node's trust value is lower than the preset threshold, the node disconnects from it and will not receive any subsequent content/request from this neighboring node. Consequently, the compromised router can no longer inject any contaminated content into the network.

With this simple trust mechanism, our scheme can still work well in the scenario where routers do not trust each other. To be noted, when implementing the scheme with the supplement mechanism, each node takes additional random re-verification and adjusts the neighboring nodes' trust value.

VI. Security Analysis

In this section, we analyze the security features of our scheme in terms of data validation protection and unforgeability, and analyze the influence of the sufficient detection rate from bloom filter in our scheme.

A. Data Validation Protection:

Lamma 1: All content transmitted in the ICN network is always verified once before it reaches the user, and each router participating in the transmission has the same probability of performing verification.

Proof: In the probabilistic verification protocol, we assume that *n* intermediate routers transfer the content and participate in the protocol. The probability that the *k*th intermediate router validates its content when it is marked as valid is $P_k = \frac{1}{n-(k-1)}$. In other words, we can consider the probability P_k' that the *k*th intermediate route authenticates content being:

$$\begin{aligned} P_k' &= (1-P_1) * (1-P_2) * \cdots * (1-P_{k-1}) * \frac{1}{n-(k-1)} \\ &= \left(1-\frac{1}{n}\right) * \left(1-\frac{1}{n-1}\right) * \cdots * \frac{1}{n-(k-1)} \\ &= \frac{n-1}{n} * \frac{n-2}{n-1} * \cdots * \frac{1}{n-k-1} \\ &= \frac{1}{n}. \end{aligned}$$

Here $P_1' = P_2' = \cdots = P_n'$ which means that the probability of authenticating the content is the same for all participating intermediate routers on the transport path. Indeed, if the *n*th intermediate router needs to authenticate the content, the probabilistic of authentication is:

$$P_n = \frac{1}{n-(n-1)} = 1$$

which guarantees that the content will definitely be authenticated once.

B. Unforgeability

Our scheme ensures that no content provider can forge the signature that makes their content valid in the process of our verification algorithm.

As shown in Section IV-C, for an adversary who plays as a content provider publishes an invalid content into ICN network successfully, it should forge a signature that:

- Adversary generates a valid signed recorded content *m* without knowing the signing key vk_u.
- Adversary generates a valid content recorded in the bloom filter stored in a router.

For the first point, due to the existential unforgeability of the signature scheme ECDSA in Section III-A, no PPT algorithm can forge a signature on content without the signing key.

For a content recorded in a router, noted that since only the routers in the ISP network use a keyed collision-resistant hash function as described in Section III-B, no adversary knows the mapping of bloom filter stored in our routers. Without the loss of generality, we assume that a router R_i validates a content with a bloom filter *bf*. The adversary should forge a signature s_m' which holds the following condition with a significantly higher possibility than a random guess:

$$Test\big(bf, s_m'\big) = 1.$$

The probability of the above hypothesis is

$$\begin{aligned} &Pr[\text{forging an signature to match bf}] \\ &\approx \left(1-\left(1-\frac{1}{m}\right)^{ln}\right)^{l}, \end{aligned}$$

which is set as a false positive rate of a bloom filter α.

C. Sufficient Detection Rate From Bloom Filter

In our proposed scheme, we use bloom filters for storing and sharing verification information. We guarantee that the information of false positive rate is sufficient so that it could defend against covert security. Assume the number of hash function is $l = \frac{m}{n}\dot{l}n2$ and the proportion $\frac{m}{n}$ is a non-negative constant γ, we have:

$$fp \approx \left(0.5^{ln2}\right)^{\gamma} > \frac{1}{2^{\gamma}} > 0.$$

As γ is a non-negative constant, we guarantee that the bloom filter is within the definition of covert security.

D. Cost Analysis

There are two main threats against our mechanism: attackers may transmit contaminated content through a fake tag or a reasonable tag applied for authentication.

The former attack has a low cost, which is implied by using expired tags, fake tags, or unauthorized use of shared or replayed tags. However, we can defend this attack at edge nodes with acceptable cost by performing pre-filtering procedures for detecting the invalid tags. The latter attack is more complex, in which attackers use a legal tag to go through edge nodes and inject contaminated content into intermediate nodes. In this case, our proposed CSEVP framework provides an additional method, named one-time content verification, where several nodes are selected with a small probability. These nodes implement verification and ensure content security with the probabilistic verification method. The cost of one-time content verification is relatively small, which can be further reduced by utilizing the bloom filter that stores the verification results and can be exchanged between neighboring nodes. However, the implementation cost of attackers is very high. As we can locate the malicious provider through the tracing mechanism and add it to the blacklist, attackers need to apply new tags to implement attacks frequently.

To sum up, we can defend against both attacks at a low cost to ensure content security, while the attacker's cost for executing attacks is very high. Therefore, our system is reasonably secure.

VII. Performance Analysis

In this section, we analyze the efficiency of the proposed scheme. First, we analyze the computation resource overhead of the probabilistic disjoint verification protocol and simulate the effect of performing computation resource allocation in a simulated network. Then, we analyze the overheads of the bloom filter-based content verification in our scheme and compare it with the regular ICN scheme and the scheme in [37] to

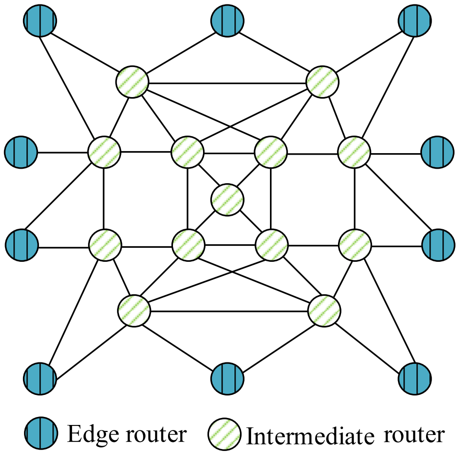


Fig. 6. Topology in the simulation.

demonstrate the superiority of our scheme in the verification efficiency.

We use NS-3 and ndnSIM [48] to simulate our protocol integrated into standard NDN [1]. All the experiments are conducted on a Linux system (Manjaro 18.1 KDE) with a 2.8GHz Intel Core i7 processor and 16G RAM.

A. Efficiency of Probabilistic Verification Protocol

First, we simulated the computational overhead caused by the probabilistic verification protocol. We implemented the probabilistic verification protocol algorithm shown in Algorithm 1 in a simulation environment and tested it 1000 times to count the actual verification overhead. Our measurement indicates that the actual calculation time of the protocol is 0.00447ms, which is an entirely acceptable overhead for the intermediate router.

Then, we need to verify that the probabilistic verification protocol can effectively and evenly distribute the local verification pressure of the network to the entire network and achieve a certain degree of computing resource integration. In order to test our protocol, we perform a stress test on the validity of the content in a simulated network environment. As shown in Fig. 6, it includes 23 routers with 10 edge routers and 13 intermediate routers. To facilitate the verification pressure of the network in the subsequent, we abstract the topology into a grid one as shown in Fig. 7. The grid topology counts the number of verification calculations for each node. The darker the color of a grid, the greater the number of verifications of this router.

In the experiment, any user can randomly initiate a request from the edge, and the responses may reply from any router in the network. The response router can be an intermediate router, or the feedback from the edge that connects a content provider. In the traditional scheme and the scheme in [37], the content will be verified at the hit node, while in ours, it will be collaboratively verified by routers along the path according to the probabilistic verification protocol.

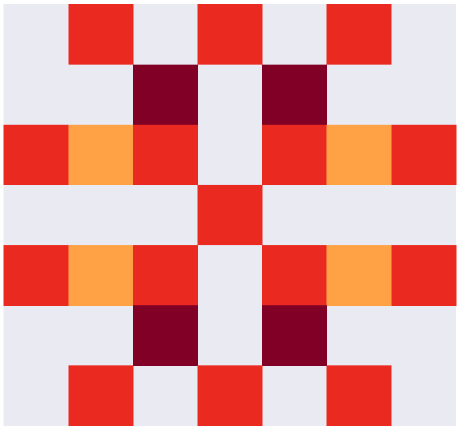
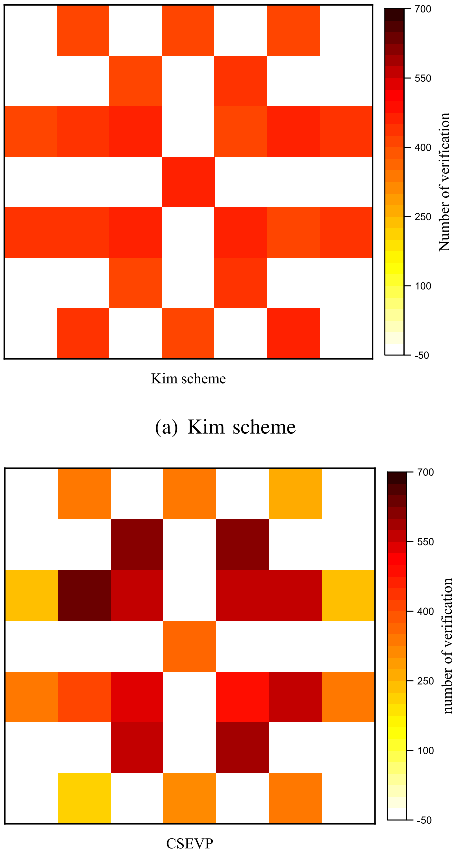


Fig. 7. Topology in the simulation with transform.



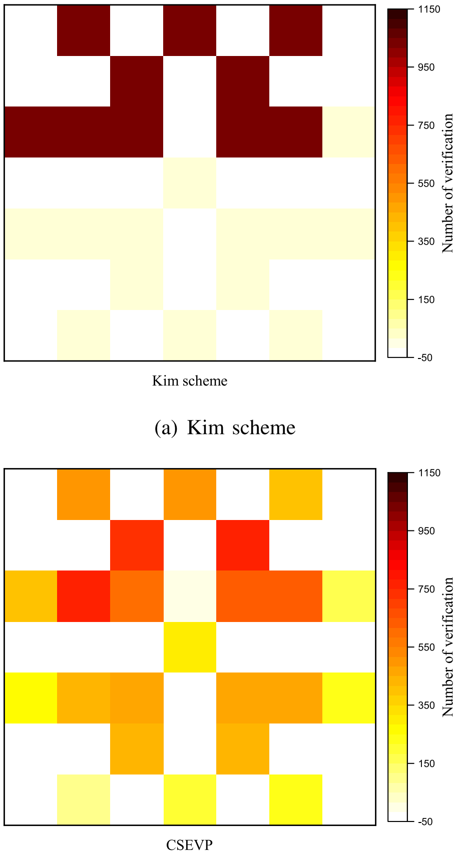
(a) Kim scheme

(b) CSEVP

Fig. 8. The number of the verification times on each router with uniform distribution.

The experiment consists of two parts. In the first part, the user's request from the edge will be randomly sent to any router with equal probability in the network and then the hit router responds to the user. The corresponding router can be an intermediate node in the network or a content provider from outside the edge. The request reaches the hit node along a completely random path and responds in a reverse way. The experiment results are shown in Fig. 8.

Fig. 8(a) represents the number of verifications undertaken by each router in the traditional ICN verification scheme. The hit points of each request are completely random so that the probability distribution of the hit routers is uniform. The number of verifications of each router in the network is the



(a) Kim scheme

(b) CSEVP

Fig. 9. The number of the verification times on each router with non-uniform distribution.

same. Fig. 8(b) shows the number of verifications undertaken by each router that uses the Probabilistic Verification Protocol. We can see that after adopting the Probabilistic Verification Protocol, the number of times each router verifies the content is also relatively balanced.

However, in practice, since most users will request popular content, the routers adjacent to the user will quickly cache the user's popular content and respond to the request. Thus, numerous requests for the same content will be gathered and responded at some specific routers. Therefore, in the second part of the experiment, the request will focus on hits on randomly selected routers in the network. The experiment results are presented in Fig. 9.

In this part of the experiment, we design the most extreme case to verify that our Probabilistic Verification Protocol scheme can evenly distribute the local verification pressure to the entire network. In the experiment, the hit points of the content on the network were concentrated on the 6 routers in the upper left corner. Each router hits 1,000 contents and responds to the users in the network. Traditional schemes verify content on hit routers, which leads to verification focusing on individual routers. In Fig. 9(a), it can be clearly seen that the upper left routers bear the most verification pressure, while other nodes in the network are idle, and their calculation and verification resources are completely wasted.

After using Probabilistic Verification Protocol, as shown in Fig. 9(b), the verification pressure is evenly distributed to each node of the network. Compared to Fig. 9(a), all nodes in the network participate in the verification calculation. And the total number of verification calculations is evenly distributed to each node in the network. Combining the topology diagram and the characteristic diagram, the nodes' verification overhead that is with concentrated hits is gradually dispersed to each node in the network, and the computing resources of the entire network are fully utilized. At the same time, compared with Fig. 9(b) tested under completely uniform distribution conditions, the verification pressure allocated to each router is relatively uniform. This further shows that the Probabilistic Verification Protocol can effectively verify the joint nodes, but the protocol's effect is not affected by the distribution of hits in the network.

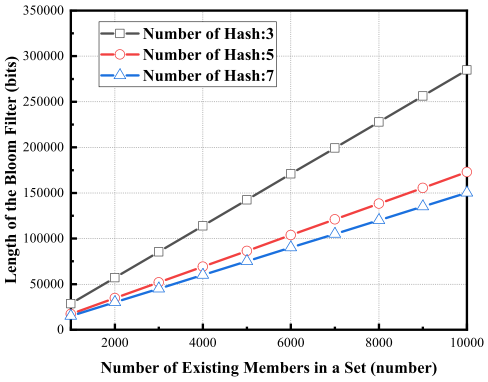


Fig. 10. Space cost about bloom filter.

B. Performance of Verification and Information Sharing Based on Bloom Filter

In this section, we evaluate the improvement of verification efficiency by our collaborative authentication scheme. We analyze the storage cost and verification delay of our scheme. At the same time, through network simulations, we analyze the additional communication overhead caused by information interaction during collaboration.

Analysis of storage: Bloom filter will bring additional storage overhead while speeding up verification. We simulated our scheme in the ICN network environment and evaluated whether the space occupation of the bloom filter is reasonable.

Above all, we estimate the actual size of the bloom filter through simulation. We mentioned earlier that the false positive rate of an m-bit Bloom filter is: $fp \approx (1-(1-\frac{1}{m})^{ln})^{l}$, where n is the number of existing members in a set, and l is the number of hash functions used in the bloom filter. When l and fp are fixed, according to the above formula, m will increase with the change of n. Fig. 10 shows the relationship between the size of the bloom filter m and a number of content stored in the bloom filter n when the false positive rate fp is fixed at 0.001 and the number of hash functions used in the bloom filter l is fixed at 3, 5, 7. We noticed that m and n have a linear relationship under the premise that h and fp are fixed. In fact, the formula can be deduced to:

$$fp \approx \left(1-\left(1-\frac{1}{m}\right)^{ln}\right)^{l} \approx \left(1-e^{-ln/m}\right)^{k}.$$

When m and n are fixed, the value of l that can minimize fp is:

$$l = \frac{m}{n} ln2 \approx \frac{9m}{13n}.$$

According to the above formula, for any given fp, we have:

$$n = m\frac{ln(0.6185)}{ln(fp)}, l = -\frac{ln(fp)}{ln(2)}.$$

According to Fig. 10, in the most extreme case, when n is 10000, a bloom filter with a length of 284737bit is required to ensure the effectiveness of the bloom filter at a false positive rate of 0.001, which makes up approximately 34.75KB of space. This is equivalent to the space resources required by a node to process 10,000 different contents per unit of time. Compared with the storage capacity of nodes in the ICN network, the overhead of this part can be ignored.

In practice, a router may need to process a number of contents within a certain time. Hence, we perform a simulation in the NDN environment by using ndnSIM, and the topology of the network is generated by the two-layer top-down hierarchical model in BRITE [49] that has 1000 routers. Specifically, 10% of routers are users and they send 1000 interests per second, and routers are linked to each other in which the bandwidth is 1Gbps and the delay is 10ms. We randomly select a CP from the intermediate routers and it responds to all the requests through edge routers. In a hundred-second simulation experiment, we count the number of content packets that pass through a node within 20 seconds. According to statistics, the average number of contents passing through a node in 20s is 1591. If these contents are not the same with each other, the occupied space is 22874 bits or 2.79KB. This is still within the acceptance range of a single node.

Evaluation of Performance Improvement at Single Router: In the efficiency evaluation of single router content validity verification, a router processes a certain amount of contents in a unit of time. All arriving contents conform to Zipf-Mandelbrot distribution function with the parameter s and q according to [45]. We test the time required for a router to check multiple contents in sequence under different scenarios. The content verification in a router is performed by a signature algorithm. We choose ECDSA on NISP256p as the algorithm used in the verification test. In the ordinary ICN scheme, the routers will check the arriving contents one by one. In this scheme, the routers will use a bloom filter to accelerate the verification. In the comparison in [37], the router verifies the content of the second arrival, and the verification is stored in the LCS for comparison in the subsequent content verification. For other contents that appear only once, no verification is performed. The verification of this part of the content is generally completed by the user. We generate multiple sets of content lists that conform to the Zipf distribution based on adjusting the total number of content N and corresponding parameters s.

In the experiment, we generate a series of contents that obey the Zipf-Mandelbrot distribution for the router. The content that arrives at the router is from 2000 selected units numbered according to the popularity ranking from 1 to 2000, and 1 represents the most popular one. The content of each unit is 1MB. In our experimental environment, it takes 0.0013s to perform a unit content signature verification operation. The bloom filter used in the experiment occupies 97.6KB, and the set false positive rate is 0.001. The experiment simulates three scenarios under the same environment and performs performance testing on a single router. The result is shown in Fig. 11.

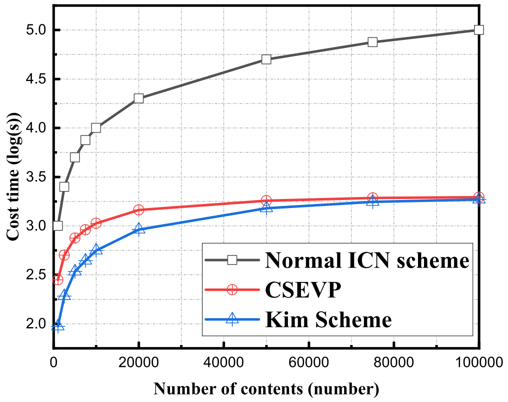


Fig. 11. Verification time cost at a single router.

We notice that in the traditional ICN content verification scheme because all the content needs to be verified once, the overhead is not negligible when a single router needs to process a number of data in a unit of time. It can be clearly seen in Fig. 11 that when the number of data increases, the delay cost of the traditional method at a single router has far exceeded the cost of the verification method proposed by [37] and our solution. The verification scheme mentioned in this scheme can still achieve efficient content verification processing when the router faces a number of data per unit time. The results show that when a single router processes 100,000 units of content in sequence, the router can still complete the validity check of all contents within 2.21706s.

At the same time, we compare with Kim's plan as shown in Fig. 11. In Kim's solution, the content is only verified when the cache is hit again. Therefore, there will be some unpopular contents to be ignored in its plan because the contents only pass through the node once. The omitted detection of contaminated contents in this part will affect the actual experience of the requesting user. Our solution guarantees that each content will be detected at least once while minimizing verification overhead. The experimental results show that in the case where a few contents need to be detected, the advantages of Kim's scheme over our scheme are acceptable. When the number of the requested contents is 1000, Kim's solution is 0.21s higher than ours, and it also brings more missed detections. When the amount of contents requested gradually increases, the gap between Kim's solution and ours is getting smaller and smaller. At the same time, considering that Kim's solution can only avoid the secondary inspection of the content in the cache, while in our solution, the node only needs to store a few of information to achieve efficient verification of the content, and the content does not need to be stored in the CS table. Our solution can be more effective in space.

Performance of Collaborative Verification: In this subsection, some simulations are performed to evaluate the effect of collaborative verification in ICN. Our performance evaluation and analysis are divided into the following parts. First, in the simulation environment, we evaluate the transmission overhead and the efficiency of interaction required for the interaction of the bloom filter between routers in the ICN network. Furthermore, we need to test the efficiency of content verification after the information exchange. Finally, we analyze the impact of the bloom filter's corresponding parameters on the merge of the bloom filter and give the filter parameter selection range according to the specific state of the network.

1) Efficiency of Content Verification With Collaboration: We show the efficiency of multi-router collaboration with theoretical analysis and experiment verification.

Theoretical Analysis: The experiment simulates the improvement of verification efficiency after multiple rounds of verification information exchange and merges in the network nodes, and at the same time estimates the reasonable number of information exchange rounds according to the experimental environment and data scale.

The experimental network is the same as the one used in the previous section and 10,000 different contents are distributed in it. The content requests received by each node are distributed according to Zipf. At the beginning of the experiment, each node receives 100 pieces of contents and processes them. We assume that the node verifies the rationality of these contents and saves the verification results in its own bloom filter. After that, the node began to exchange verification information with each other. We select 50 nodes, recorded their exchange information for each round. And after each round of exchange, we perform verification tests on the subsequent 1,000 requests that arrive according to the Zipf distribution. We count the number of verification contents saved by the node after each round of exchange and the verification efficiency for subsequent requests.

Before the first round of exchange in the experiment, the node's success rate for the newly 1000 arriving contents using the existing verification information is 57.7%. That is, nearly half of the content needs to be verified using traditional methods. This is undoubtedly a huge overhead. After 6 rounds of information exchange, the node's success rate of a quick verification of subsequent requests reaches 90.4%. The theoretical communication cost of 6 rounds of information exchanges is about 280.7ms, which is undoubtedly very low compared to the verification time required by the nodes in the traditional solution and Kim's solution.

Experimental Verification: To illustrate the performance improvement of multi-router collaboration, we show the verification efficiency after multiple rounds of verification information exchanges and merges in the network nodes. Specifically, we present the time cost of Kim's scheme and CVESP in six rounds where CVESP can share bloom filters with neighboring nodes for each round.

From Fig. 12 we can see that CVESP always outperforms Kim's scheme in the overall time cost, and the node verification overhead of CVESP is even better than that of routers in Kim's scheme after five simple rounds of exchange. In the figure, "Kim-user" and "Kim-router" denote the time cost on the user's side and router's side in Kim's scheme in the simulation scenario, respectively. We can observe that CVESP, which stores and shares verification information, can reduce the time cost dramatically. Besides, with the increase of rounds, the authenticated information saved at nodes in CVESP increases. Thus, the time cost of verification can be decreased. However, the verification time of Kim's scheme remains unchanged for lacking shared verification results.

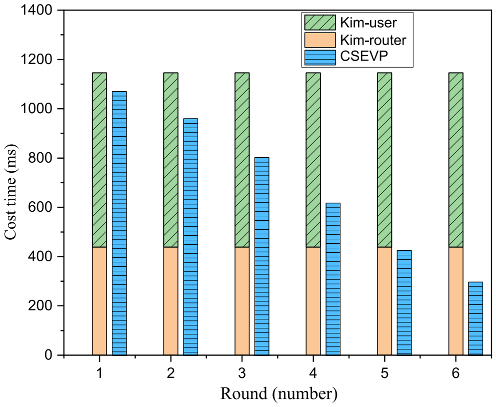


Fig. 12. Verification cost of multi-router collaboration.

VIII. Conclusion

In this paper, we presented a collaborative, secure, and efficient content validation protection scheme, called CSEVP, for ICN. In CSEVP, we implement collaborative authentication for content validity among multiple routers in the ICN network. Furthermore, By leveraging a probabilistic verification protocol, routers participating in transmission can share the pressure of validity verification. Also, the introduction of bloom filter helps routers share verification results and increases the efficiency of validity verification. Via experimental analysis, the results demonstrate that CSEVP is a promising solution for content validation protection in ICN, which meets the security requirements and also guarantees good enough efficiency.

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. Experiments Technol. (CoNEXT)*, 2009, pp. 1–12.

[2] K. Xue *et al.*, "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.

[3] 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.

[4] 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.

[5] R. Tourani, S. Misra, T. Mick, and G. Panwar, "Security, privacy, and access control in information-centric networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 566–600, 1st Quart., 2018.

[6] S. Misra, R. Tourani, and N. E. Majd, "Secure content delivery in information-centric networks: Design, implementation, and analyses," in *Proc. 3rd ACM SIGCOMM Workshop Inf. Centric Netw. (ICN)*, 2013, pp. 73–78.

[7] 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.

[8] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS and DDoS in named data networking," in *Proc. 22nd IEEE Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2013, pp. 1–7.

[9] C. Ghali, G. Tsudik, and E. Uzun, "Network-layer trust in named-data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 12–19, 2014.

[10] S. DiBenedetto and C. Papadopoulos, "Mitigating poisoned content with forwarding strategy," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2016, pp. 164–169.

[11] C. Ghali, G. Tsudik, and E. Uzun, "Needle in a haystack: Mitigating content poisoning in named-data networking," in *Proc. NDSS Workshop Security Emerg. Netw. Technol. (SENT)*, 2014, pp. 1–10.

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

[13] B. Nour, H. Khelifi, R. Hussain, S. Mastorakis, and H. Moungla, "Access control mechanisms in named data networks: A comprehensive survey," *ACM Comput. Surveys*, vol. 54, no. 3, pp. 1–35, 2021.

[14] H. Huang, Y. Wu, F. Xiao, and R. Malekian, "An efficient signature scheme based on mobile edge computing in the NDN-IoT environment," *IEEE Trans. Comput. Social Syst.*, vol. 8, no. 5, pp. 1108–1120, Oct. 2021.

[15] S. Misra, R. Tourani, F. Natividad, T. Mick, N. E. Majd, and H. Huang, "AccConF: An access control framework for leveraging in-network cached data in the ICN-enabled wireless edge," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 1, pp. 5–17, Jan./Feb. 2019.

[16] B. Bera, S. Saha, A. K. Das, and A. V. Vasilakos, "Designing blockchain-based access control protocol in IoT-enabled smart-grid system," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5744–5761, Apr. 2021.

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

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

[19] J. Ni, K. Zhang, and A. V. Vasilakos, "Security and privacy for mobile edge caching: Challenges and solutions," *IEEE Wireless Commun.*, vol. 28, no. 3, pp. 77–83, Jun. 2021.

[20] M. Amadeo *et al.*, "Information-centric networking for the Internet of Things: Challenges and opportunities," *IEEE Netw.*, vol. 30, no. 2, pp. 92–100, Mar./Apr. 2016.

[21] E. G. AbdAllah, H. S. Hassanein, and M. Zulkernine, "A survey of security attacks in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1441–1454, 3rd Quart., 2015.

[22] M. Xie, I. Widjaja, and H. Wang, "Enhancing cache robustness for content-centric networking," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2012, pp. 2426–2434.

[23] L. Yao, Y. Zeng, X. Wang, A. Chen, and G. Wu, "Detection and defense of cache pollution based on popularity prediction in named data networking," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 6, pp. 2848–2860, Nov./Dec. 2021.

[24] T. Nguyen *et al.*, "A security monitoring plane for named data networking deployment," *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 88–94, Nov. 2018.

[25] T. Nguyen *et al.*, "Reliable detection of interest flooding attack in real deployment of named data networking," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 2470–2485, 2019.

[26] A. Fiandrotti, R. Gaeta, and M. Grangetto, "Securing network coding architectures against pollution attacks with band codes," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 730–742, 2018.

[27] Y. Yu, "Public key management in named data networking," Named Data Netw., Univ. California, Los Angeles, CA, USA, Rep. NDN-0029, 2015. [Online]. Available: https://named-data.net/wp-content/uploads/2015/04/ndn-0029-1-public-key-management-ndn.pdf

[28] W. Cui, Y. Li, Y. Xin, and C. Liu, "Feedback-based content poisoning mitigation in named data networking," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2018, pp. 759–765.

[29] A. Compagno, M. Conti, C. Ghali, and G. Tsudik, "To NACK or not to NACK? Negative acknowledgments in information-centric networking," in *Proc. 24th Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2015, pp. 1–10.

[30] 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, pp. 308–320, 2015.

[31] H. Kang, Y. Zhu, Y. Tao, and J. Yang, "An in-network collaborative verification mechanism for defending content poisoning in named data networking," in *Proc. 1st IEEE Int. Conf. Hot Inf. Centric Netw. (HotICN)*, 2018, pp. 46–50.

[32] H. Khelifi, S. Luo, B. Nour, H. Moungla, and S. H. Ahmed, "Reputation-based blockchain for secure NDN caching in vehicular networks," in *Proc. IEEE Conf. Stand. Commun. Netw. (CSCN)*, 2018, pp. 1–6.

[33] W. Cui, Y. Li, Y. Zhang, C. Liu, and M. Zhan, "An ant colony algorithm based content poisoning mitigation in named data networking," in *Proc. 18th IEEE Int. Conf. Trust Security Privacy Comput. Commun. 13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, 2019, pp. 176–183.

[34] N. Yang, K. Chen, and M. Wang, "SmartDetour: Defending blackhole and content poisoning attacks in IoT NDN networks," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12119–12136, Aug. 2021.

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

[36] G. Bianchi, A. Detti, A. Caponi, and N. B. Melazzi, "Check before storing: What is the performance price of content integrity verification in LRU caching?" *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 59–67, 2013.

[37] D. Kim, J. Bi, A. V. Vasilakos, and I. Yeom, "Security of cached content in NDN," *IEEE Trans. Inf. Forensics Security*, vol. 12, pp. 2933–2944, 2017.

[38] D. Kim, S. Nam, J. Bi, and I. Yeom, "Efficient content verification in named data networking," in *Proc. 2nd ACM Conf. Inf. Centric Netw. (ICN)*, 2015, pp. 109–116.

[39] J. Zhou, J. Luo, J. Wang, and L. Deng, "Cache pollution prevention mechanism based on deep reinforcement learning in NDN," *J. Commun. Inf. Netw.*, vol. 6, no. 1, pp. 91–100, 2021.

[40] 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.

[41] P. He, Y. Wan, Q. Xia, S. Li, J. Hong, and K. Xue, "LASA: Lightweight, auditable and secure access control in ICN with limitation of access times," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.

[42] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Security*, vol. 1, no. 1, pp. 36–63, 2001.

[43] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[44] Y. Qiao, T. Li, and S. Chen, "Fast bloom filters and their generalization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 93–103, Jan. 2014.

[45] Z. K. Silagadze, "Citations and the Zipf-Mandelbrot's law," 1999, [Online]. Available: https://arxiv.org/abs/physics/9901035

[46] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM Workshops)*, 2012, pp. 310–315.

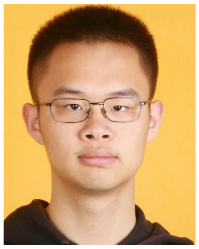
[47] R. Tourani, R. Stubbs, and S. Misra, "TACTIC: Tag-based access control framework for the information-centric wireless edge networks," in *Proc. 38th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 456–466.

[48] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2.0: A new version of the NDN simulator for NS-3," Named Data Netw., Univ. California, Los Angeles, CA, USA, Rep. NDN-0028, 2015. Accessed: Dec. 2021. [Online]. Available: https://named-data.net/wp-content/uploads/2013/07/ndn-0028-1-ndnsim-v2.pdf

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

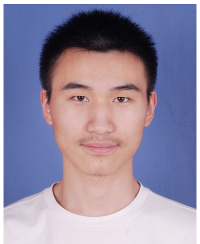
Kaiping Xue (Senior Member, IEEE) received the bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC) in 2003, and the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. From May 2012 to May 2013, he was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, University of Florida. He is currently 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 serves on the Editorial Board of several journals, including the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and 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*. He is an IET Fellow.

Jiayu Yang (Graduate Student Member, IEEE) received the B.S. degree in information security from the School of Cyber Science and Technology, University of Science and Technology of China (USTC) in 2019. She is currently pursuing the Ph.D degree from 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. degree in information security from the School of Cyber Science and Technology, University of Science and Technology of China (USTC), in 2018, and the master's degree in information security from the School of Cyber Security, USTC, in 2021. 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 on the Faculty of Computer Science and Engineering with the University of Aizu, Japan (as an Associate Professor and then a Professor). He has authored and coauthored more than 120 technical papers in various archival journals and conference proceedings. He is currently a Professor of Computer and Information Science Department with Fordham University. His research interests include cloud computing, big data, IoT, and cognitive radio networks. He was a Guest Editor or a Lead Guest Editor for several special issues in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON CLOUD COMPUTING, and IEEE TRANSACTIONS ON BIG DATA. He also served as an Associate Editor of IEEE TRANSACTIONS ON CLOUD COMPUTING from 2014 to 2018, and JOURNAL OF CIRCUITS, SYSTEMS AND COMPUTERS from 2013 to 2018.



Jian Li (Member, IEEE) received the B.S. degree from the Department of Electronics and Information Engineering, Anhui University in 2015, and the Ph.D. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), in 2020. From November 2019 to November 2020, he was a Visiting Scholar with the Department of Electronic and Computer Engineering, University of Florida. He is currently a Postdoctoral Researcher with the School of Cyber Science and Technology, USTC. His research interests include wireless communications, satellite networks, and next-generation Internet.

Qibin Sun (Fellow, IEEE) received the Ph.D. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China in 1997, where he is currently a Professor with the School of Cyber Science and Technology. He has published more than 120 papers in international journals and conferences. His research interests include multimedia security, network intelligence, and security.

Jun Lu received the bachelor's degree from Southeast University in 1985 and the master's degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, in 1988, where he is currently a Professor. His research interests include theoretical research and system development in the field of integrated electronic information systems. He is an Academician of the Chinese Academy of Engineering.