RPBV: Reputation-Based Probabilistic Batch Verification Scheme for Named Data Networking

Kunpeng Ding*, Jiayu Yang*, Jiangping Han*, Bobo Wang*, Ruidong Li†, Kaiping Xue*‡
*School of Cyber Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027 China
† Institute of Science and Engineering, Kanazawa University, Kanazawa, Ishikawa 920-1192 Japan
‡Corresponding author: kpxue@ustc.edu.cn

2023 IEEE/ACM 31st International Symposium on Quality of Service (IWQoS) | 979-8-3503-9973-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/IWQOS57198.2023.10188736

***Abstract*—As a promising implementation of Information Centric Networking, Named Data Networking (NDN) can facilitate content distribution with in-network caching and location-independent data access. However, the reliance on caches makes NDN vulnerable to content poisoning attacks, which waste network resources and decrease transmission efficiency. Most mitigating schemes follow the pattern that each content is repeatedly verified individually in each router and all producers have the same status, which wastes computation resources and degrades network performance. In this paper, we propose a Reputation-based Probabilistic Batch Verification (RPBV) scheme to address the issue, in which producers' reputation is estimated according to verification results to distinguish different producers. We provide an adaptive probabilistic verification method based on reputation to avoid a lot of unnecessary verification operations. At the same time, we adopt an efficient batch verification algorithm to simultaneously verify multiple content, which reduces the overhead greatly. With the above mechanisms implemented only on the edge router to avoid repeated verification, we provide an optional probabilistic verification method on intermediate routers to strengthen the security. The extensive simulations show that RPBV achieves much lower computation overhead and shorter content retrieval time than the traditional schemes.**

***Index Terms*—Named data networking, content poisoning attack, batch verification, probabilistic verification, reputation.**

I. Introduction

The Internet's dominating service has shifted from remote data access to large-scale content distribution. To accommodate this requirement, Named Data Networking (NDN) [1] has been proposed as a new network paradigm, which changes the Internet architecture from the host-centric model to the content-oriented model. in NDN, each content is assigned with a globally unique name, based on which the network performs routing, requesting, and publishing. Besides, NDN uses in-network caching to optimize bandwidth usage and enables location-independent content access for mobility management and multi-path forwarding, which facilitates content distribution and lower content retrieval latency.

In spite of these advantages, NDN poses new security challenges [2]–[4]. The content poisoning attack is a typical one intensified by the cache, where an adversary injects fake content into the network. Since routers independently cache content, the fake content will be stored in caches opportunistically if not verified, and unknowingly spread in the network by NDN itself. As a result, a wide range of contaminated caches degrades the caching performance and isolates users from valid content.

To cope with this attack, NDN enables each data packet to carry a digital signature that is verified by routers and users. However, due to the huge signature verification overhead, routers cannot afford to verify signatures of each data packet, which can arrive at a rate above hundreds of Gbps [5], [6]. Researchers have thus proposed some practical solutions from two aspects. One category aims to optimize the way of verifying packets. For example, Li *et al.* [7] proposed a one-time signature algorithm, which verifies packets through lightweight hash operations. However, every time a signature is generated, related keys must be updated to guarantee the security. Additionally, many tokens must be carried by data packets and stored in routers, leading to a lot of communication and storage overhead. Another category considers leveraging feedback information instead of verifying packets themselves. For example, Ghali *et al.* [8] designed a ranking algorithm based on users' feedback information, which ranks content and returns the highest-ranked content to users, thus avoiding the transmission of fake content in the network. However, the scheme is at risk if attackers forge the feedback to interfere with the ranking result.

The aforementioned schemes are either insufficiently secure or entail additional costs to achieve security. There are many redundant verification operations in these schemes, which can be eliminated to ease the burden on routers without compromising security. First, when packets arrive at line speed, they cannot be verified individually at the same speed and thus accumulate in the router. Simultaneously verifying packets can help shorten verification time and reduce verification costs. Secondly, since the majority of producers in the network are honest and only publish valid content, it is unnecessary to verify every packet. Instead, verification can be done probabilistically, with the verification probability related to the producer's reputation to balance cost and security. Finally, it is no need to verify the same packet in every router. It can block fake content entirely to verify packets solely in the edge router, or mainly in the edge router and partly in intermediate routers.

In this paper, we propose a reputation-based probabilistic batch verification scheme, named RPBV, which estimates

979-8-3503-9973-8/23/$31.00 ©2023 IEEE

each producer's reputation according to verification results of packets published by the producer. Based on the reputation, a dynamic and adaptive probabilistic verification method is designed to ensure minimal verification of packets from each producer while providing the preset security level. Besides, RPBV adopts a batch verification algorithm to reduce the average verification time, which works on an identity-based signature algorithm that avoids verifying certificates. Furthermore, we implement the above mechanisms only on the edge router to avoid repeated verification in each router. The edge router has advantages in three aspects: first, the edge can block fake content at the beginning of the packet's journey into the network; second, the edge is a necessary node for packets to enter the network, so it can receive a large number of packets in a short time, thus suitable for batch verification; third, it is connected directly with end-hosts, thus suitable for estimating producers' reputation. Optionally, intermediate routers can verify packets with small probability to reinforce security. Our contributions can be summarized as follows:

- We propose an adaptive batch verification scheme to reduce the verification overhead. Meanwhile we adjust the batch size periodically and combine it with individual verification to mitigate inherent defects of batch verification, e.g., waiting latency and coarse-grained results.
- We extend the scenario of content poisoning attacks and measure producers' reputation to accommodate semi-malicious attackers. Based on reputation, we further design a dynamic probabilistic verification method to minimize the number of verification operations.
- We leverage the location and network traffic characteristics of edge routers and implement our scheme on them. Security analysis shows that signature unforgeability is guaranteed and Denial of Service (DoS) attacks and content poisoning attacks are resisted. Simulation results show that the content retrieval time of our scheme is shortened by about 44%.

The rest of this paper is organized as follows. Section II introduces the related work. In Section III, we present the system model, threat assumption, and preliminaries. The proposed scheme is given in Section IV. Then Section V shows the security analysis. Section VI shows the performance analysis. Finally we conclude this paper in Section VII.

II. Related Work

Content poisoning attacks have been studied in the literature in recent years. Proposed schemes can be divided into three categories. In some schemes, routers rely on feedback information from users or neighbor nodes to remove fake content and detect malicious producers. In [8], the authors provided a ranking function based on users' feedback to make valid content rank higher than fake content. Routers always select the highest-ranked content in response to a request, but attackers may forge the feedback to interfere with ranking results. Qu *et al.* [9] proposed a token-based policy, which evaluates producers' activities based on feedback and signature verification results and accordingly assign tokens to control producers' activities. However, it costs much to verify every data packet. Saha *et al.* [10] proposed a centralized method, where every node records the number of genuine and fake content received from neighbor nodes, and the system collects all nodes' feedback information to blacklist malicious nodes. However, the method cannot accurately differentiate between honest and malicious nodes. Dibenedetto *et al.* [11] made the next hop that returned bogus data packets the least preferred option for future requests. However, this approach may also penalize honest producers who are adjacent to malicious producers. Cui *et al.* [12] utilize the feedback to clean caches, trace evil source, and adjust forwarding path. Although forged feedback is blocked after verification, it leads to DoS attacks that drain resources during the verification of the feedback.

Some schemes use more lightweight verification algorithms or optimize the verification process to reduce verification costs. The self-certifying naming is adopted in some schemes [5], [13]. The content's hash or public key's hash is appended to the name for validation purposes, but it is difficult for the user to know the hash beforehand. Li *et al.* [7] proposed a one-time signature algorithm to reduce verification costs, but this requires data packets to carry extra tokens, and each node has to store many tokens, leading to increased communication and storage overhead. Huang *et al.* [14] adopted a certificateless group signature algorithm and a batch verification scheme to improve verification efficiency. However, every data packet has to be repeatedly verified several times and many valid packets are discarded if they are verified at the same time with invalid packets. Li *et al.* [15] designed a trust chain that merges authority-based trust and neighbor-based trust, which collects certificates hop by hop and updates the trust chain adaptively, leading to extra computation and storage overhead. Kim *et al.* [16] developed the "check on hit" idea, where content is stored in the cache without verification until it is requested. The scheme avoids useless operations, but users still receive fake content the first time the content enters the network. Xue *et al.* [17] proposed a collaborative verification protocol, which makes verification results shared with neighbor routers and makes content verified only once. Although repeated verification is avoided, the false positive rate of the bloom filter leads to inaccurate results.

Other schemes provide general models for content poisoning attacks. Anisetti *et al.* [18] presented a certification methodology to certify nonfunctional properties of network nodes. The methodology results in a certificate according to a set of policies aimed at proving a specific behavior, but it is complex and difficult to implement. Kar *et al.* [19] used a game-theoretical model to simulate the attacker-defender situation and found the Nash equilibrium as the strategy to take. However, the Nash equilibrium in dynamic games is not the optimal solution. Nguyen *et al.* [20] proposed a monitoring plane composed of 18 metrics, where detectors are designed to raise alarms when a metric strays from its expected behavior and a Bayesian network is leveraged to infer anomalies. the effect is doubtful in large topologies.

III. System Model, Threat Assumption and Preliminaries

A. System Model

We consider an NDN architecture consisting of three parts: content producers, an Internet Service Provider (ISP), and users, as shown in Fig. 1.

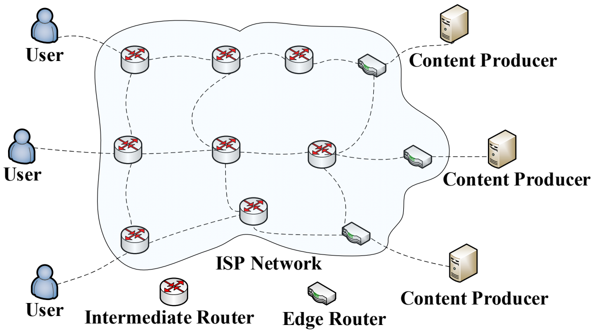

Fig. 1. System Model.

Content producers publish data packets with signatures. Routers in the ISP network are responsible for forwarding packets, caching content, and so on. In addition, they abide by our scheme to check the content's authenticity and integrity. In this paper, if a router is connected to a producer, it is called an edge router, or it is called an intermediate router. Users request the content they want by sending interest packets.

B. Threat Assumption

This paper studies content poisoning attacks, where the adversary injects fake content into the network. In theory, content poisoning attacks can be launched by routers or end-hosts, but in real deployment, routers are in a domain, e.g., an ISP [7]. Taking the control of a network element is hardly feasible while leveraging end-hosts to perform the attack is easier [21]. So content poisoning via router compromise is not a common threat. The risk of content poisoning attacks is related to the fact that it can be implemented by end-hosts only [16]. In this paper, we focus on addressing the content poisoning attack implemented by end-hosts and assume that all routers trust each other, which is a common security assumption found in related literature such as [17], [22] and [23]. Additionally, we consider the situation where routers do not trust each other, e.g., in a distributed environment, and offer an extended reputation mechanism as a supplement to the proposed scheme, presented at the end of Section IV.

This paper extend the scenario of content poisoning attacks to accommodate different kinds of producers: honest, malicious, and semi-malicious. Honest producers always publish authentic content with valid signatures. In contrast, malicious producers always publish fake content with invalid signatures. Semi-malicious producers may decide whether to publish authentic content in their namespace or fake content in others' namespace and the probability of injecting fake content is uncertain. Fake content issued by the latter two kinds of producers will pollute caches and isolate users from valid data.

C. Preliminaries

1) Bilinear Map: Let G and G_T be multiplicative cyclic groups with the same prime order q. The generator of G is g. A bilinear map can be described as $e : G \times G \rightarrow G_T$ that has the following properties: (a) *Computability*: there exists an algorithm to efficiently compute the map e; (b) *Non-degeneracy*: $e(g,g) \neq 1$; (c) *Bilinearity*: $e(u^a, v^b) = e(u,v)^{ab}$ for any $a, b \in \mathbb{Z}_q^*$ and $u, v \in G$.

2) Complexity Assumption: The identity-based signature algorithm used in this paper relies on Diffie-Hellman (DH) assumption:

Definition 1. ***DH Assumption***. For unknown $a, b \in \mathbb{Z}_q^*$, given a turple (g, g^a, g^b), where g is the generator of a cyclic group G of order q, it is infeasible to compute g^{ab}.

IV. Proposed Scheme

A. Overview

As shown in Fig. 2, the proposed scheme implements the verification process in edge routers and intermediate routers.

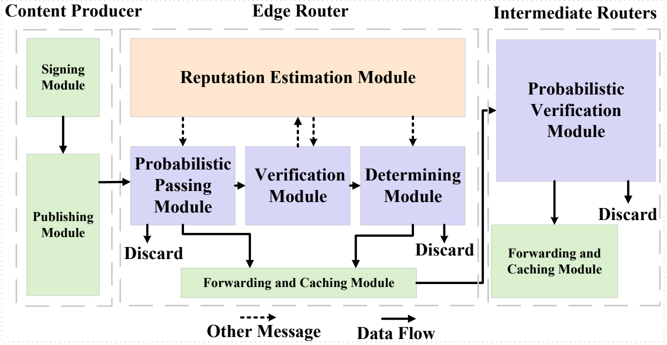

Fig. 2. System Overview.

The content producer signs the data packet with the identity-based signature algorithm [24] and publishes it to the network. An edge router maintains separate processing modules for each producer connected to it. After the edge router is connected to a producer and starts processing its packets, time is divided into consecutive intervals called Monitor Intervals (MIs). Within each MI, several parameters in these modules are updated. In the reputation estimation module, the producer's reputation is estimated based on the verification results in the edge router, which shows the producer's probability of publishing valid content. Initially, a packet goes into the probabilistic passing module, where we make a decision based on the reputation whether it should go into the verification module, or be discarded, or be forwarded directly. Subsequently, packets that enters the verification module are verified using the individual verification algorithm or the batch verification algorithm [25], in which the batch size is determined by the reputation. Since there may be still some valid packets in an invalid batch, packets in the determining module are discarded with the probability decided by the reputation. Then, in each intermediate router, packets are verified with small probability. Our scheme enables the network to verify content at low cost, with a negligible amount of fake content received by users.

B. Initialization and Signing

In the initialization step, a legitimate content producer registers to get its private key and public parameters as follows: (a) The trusted authority generates a tuple $(G, G_T, q, g, e, H_1, H_2)$, where G and G_T are multiplicative cyclic groups with the same prime order q, g is the generator of G, e is a bilinear map $G \times G \rightarrow G_T$, H_1 and H_2 are hash algorithms. Then the master secret key msk is selected randomly from $\mathbb{Z}_q^*$, and the master public key mpk is calculated as g^{msk}. (b) The producer defines its public key $pk = H_1(ID)$, where ID is its identity. Then the trusted authority calculates the corresponding secret key $sk = pk^{msk}$. While msk is only known by the trusted authority, sk is known by the trusted authority and the content producer, and other parameters are public to all entities.

In the signing step, a content producer generates the signature for content D with sk as follows. The producer randomly chooses h from G, s from $\mathbb{Z}_q^*$, and computes $S_1 = e(h, g)^s$, $a = H_2(D||S1)$, $S_2 = sk^a \times h^s$. Finally the signature is (S_1, S_2).

C. Probabilistic Passing Module in the Edge Router

It is a waste to verify every packet since multiple honest producers only publish genuine packets. Given producers' reputation, we can choose different verification probabilities for them to minimize verification times.

In the first Monitor Interval (MI), the edge router verifies all packets because reputation has not been measured. After the first MI, it verifies content probabilistically. We develop the idea of [26] to determine the verification probability p_v. To achieve fewer verification times without damaging security, we lower the verification probability of content from producers with higher reputation. However, even producers that never send fake content are subject to packet verification in case they become compromised in the future. Besides, we discard all data packets from producers that are much more likely to inject fake content than valid content, because they can harm the network more than they benefit it. Based on these principles, we design the verification probability p_v as:

$$p_v = \begin{cases} \alpha, & rep \geq 1 - \frac{\eta}{1-\alpha}, \\ 1 - \frac{\eta}{1-rep}, & \beta \leq rep < 1 - \frac{\eta}{1-\alpha}, \\ 0, & rep < \beta, \end{cases} \tag{1}$$

where $\alpha \in (0, 1)$, $\beta \in (0, 1)$, $\eta \in [0, 1]$, and rep is the producer's reputation. Here, α is the minimum verification probability for producers whose reputation exceeds a big threshold. β is a preset small threshold for identifying content producers as malicious. Packets from a producer whose reputation is less than β are discarded without verification. η is the maximum error probability that the system can tolerate, balancing efficiency and security. Typically, η is very small. Function (1) enables the edge router to verify content the least times, while ensuring that the ratio of unverified fake content does no exceed η, as presented in Theorem 1.

D. Verification Module in the Edge Router

The edge router undertakes the main verification work to reduce the error rate to a certain low extent with the help of the probabilistic passing module and determining module.

In the first Monitor Interval (MI), the edge router performs individual verification to acquire the accurate initial reputation, since batch verification cannot tell the verification result of each packet. As shown in **Algorithm 1**, to verify a packet's signature (S_1, S_2), we first compute a and then check the equation in line 2 to output the result.

```
Algorithm 1: Individual Verification
  Input: Content $D$, public key $pk$, and signature
         $(S_1, S_2)$, public parameters $g$ and $mpk$.
  Output: Valid or Invalid.
1 $a \leftarrow H_2(D||S_1)$;
2 if $e(S_2, g) = e(pk^a, mpk) \times S_1$ then
3     return Valid;
4 else
5     return Invalid;
6 end
```

During each subsequent MI, the edge router performs batch verification to save verification cost, which means we verify multiple packets simultaneously. During batch verification, a group of packets that are verified simultaneously is referred to as a batch, and the number of packets in a batch is the batch size. Batch verification is advantageous in terms of low overhead, especially for large batches. However, when the data flow is not as fast as expected, earlier arriving packets have to wait a non-negligible amount of time for later packets to form a batch. We find that the edge router usually receives a number of packets within a short period, so we implement batch verification on the edge router with a suitable batch size.

As shown in **Algorithm 2**, to verify a batch with batch size n, we compute a_j and choose δ_j for $j \in [1, n]$, and check the equation in line 5 to output the result. A valid result indicates that all packets in the batch are valid, whereas an invalid result indicates that at least one packet in the batch is invalid, and it is unclear which and how many packets are invalid.

The choice of batch size is crucial. While a larger batch size reduces the average verification costs, the verification result is less accurate as it is uncertain which packet(s) in an invalid batch are invalid. Therefore, a suitable batch size must be selected to achieve a balance. Generally, a batch with a smaller size or from a higher-reputation producer is more likely to be valid, so we choose the batch size based on reputation. Note that having too many options of distinct batch sizes leads to inaccurate reputation estimation since the estimation process presented later depends on the batch size. So we only choose two typical batch sizes M and N based on the observation of the verification cost curve that falls with the increasing batch

```
Algorithm 2: Batch Verification
Input: Content $D_j$, public key $pk_j$, and signature
       $(S_{j,1}, S_{j,2})$ for $j = 1, 2, ..., n$, public
       parameters $g$ and $mpk$.
Output: Valid or Invalid.
for $j = 1, 2, ..., n$ do
    $a_j \leftarrow H_2(D_j || S_{j,1})$;
    $\delta_j \overset{\$}{\leftarrow} \mathbb{Z}_q^*$;
end
if
  $\mathrm{e}(\prod_{j=1}^{n} S_{j,2}^{\delta_j}, g) = \mathrm{e}(\prod_{j=1}^{n} pk_j^{a_j \times \delta_j}, mpk) \times \prod_{j=1}^{n} S_{j,1}^{\delta_j}$
  then
    return Valid;
else
    return Invalid;
end
```

size. N is the batch size where the cost is significantly lower than that of individual verification, while M is the threshold beyond which increasing the batch size does not substantially reduce the cost. We exploit the reputation threshold $1 - \frac{\eta}{1-\alpha}$ used in the probabilistic passing module to classify producers into two categories and assign them two different batch sizes. We design the batch size bs as:

$$bs = \begin{cases} M, & rep \geq 1 - \frac{\eta}{1-\alpha} \\ N, & rep < 1 - \frac{\eta}{1-\alpha} \end{cases}. \tag{2}$$

E. Determining Module in the Edge Router

This module serves to determine whether to discard or forward packets after the verification result comes out. When the edge router performs individual verification, it marks and forwards valid packets while discarding invalid ones. Instead, when the edge router performs batch verification, the decision is complicated. If the batch is valid, meaning all packets in it are valid, we mark and forward all the packets. However, if the batch is invalid, a decision needs to be made on whether to discard it based on the following consideration. Firstly, it is not ideal to forward all packets in an invalid batch as it would render the previous verification process useless. Secondly, discarding all packets in an invalid batch is not ideal as there could be a significant number of valid packets among them. Here we can make use of reputation again to decide the discarding probability. Since the rate of invalid packets in an invalid batch, namely $\frac{1-rep}{1-rep^n}$, decreases with the increasing reputation, the batch shouled be discarded with a lower probability. We continue to exploit the reputation threshold $1 - \frac{\eta}{1-\alpha}$ used in the probabilistic passing module to divide producers into two types and assign two different computation formula to them. We design the discarding probability p_d as:

$$p_d = \begin{cases} \max\left\{0, 1 - \frac{\eta}{\alpha(1-rep)}\right\}, & rep \geq 1 - \frac{\eta}{1-\alpha}, \\ \max\left\{0, 1 - \frac{\eta}{1-rep-\eta}\right\}, & rep < 1 - \frac{\eta}{1-\alpha}, \end{cases} \tag{3}$$

where $\alpha \in (0, 1)$, $\eta \in [0, 1]$, and rep is the producer's reputation, which have the same meanings as those used in the probabilistic passing module. Function (3) guarantees that the ratio of bogus packets that are verified but not discarded does not exceed η, as presented in Theorem 2.

F. Reputation Estimation in the Edge Router

Verification results reflect content producers' reputation, namely the probability of sending valid content. Reputation is estimated for each content producer individually to facilitate customized choices for verification probability, batch size, and discarding probability in other modules, since the edge router can easily distinguish content from different producers.

First we consider a simple situation where every producer injects content with a constant probability of being valid. To estimate the reputation accurately, we apply Bayesian estimation that combines the analyzed data with prior information. In the Bayesian estimation model, we aim to estimate a packet's probability of being valid, referred to as ω. Since the validity of each packet is independent, each batch's validity is also independent. Thus the probability of a batch being valid is $\theta = \omega^n$, where n is the batch size. We use $X = (x_1, x_2, ..., x_m)$ to represent each of the m batches is valid or not. The Bayesian formula about θ is:

$$P(\theta \mid X) = \frac{P(X \mid \theta) P(\theta)}{\int_\Theta P(X \mid \theta) P(\theta) d\theta}, \tag{4}$$

where $P(X \mid \theta)$ is the likelihood function, and $P(\theta)$ and $P(\theta \mid X)$ are the prior distribution and posterior distribution of θ, respectively.

Now we have to calculate $P(X \mid \theta)$ and $P(\theta)$ to derive $P(\theta \mid X)$. We count the number of valid batches as t and the number of invalid batches as f to calculate the likelihood function as $P(X \mid \theta) = \theta^t \times (1-\theta)^f$ since X follows the binomial distribution. It is mathematically complex to calculate $P(\theta \mid X)$ given an arbitrary $P(\theta)$. However, if we specify the prior distribution of θ as $\mathrm{Beta}(\theta \mid a, b)$, where a and b are associated with the number of valid batches and invalid batches respectively, then the posterior distribution of θ is also a Beta distribution given by

$$P(\theta \mid X) = \mathrm{Beta}(\theta \mid t+a, f+b) = \frac{\theta^{t+a-1}(1-\theta)^{f+b-1}}{\mathrm{B}(t+a, f+b)}. \tag{5}$$

We can obtain a and b in the prior distribution by applying maximum likelihood estimation in the first monitor interval (MI), during which we perform individual verification because batch verification algorithm can only output inaccurate results about each packet. We use $Y = (y_1, y_2, ..., y_k)$ to represent whether each of the k packets is valid or not. We count the number of valid packets as t_0 and invalid packets as f_0 to obtain the maximum likelihood estimator of the initial ω as ω_0 by maximizing $P(X \mid \omega) = \omega^{t_0} \times (1-\omega)^{f_0}$:

$$\omega_0 = \frac{t_0}{t_0 + f_0}. \tag{6}$$

With the known ω_0, we set $a = 1$ and $b = \frac{1}{\omega_0^n} - 1$. From equation (5), we can get the posterior distribution of θ:

$$P(\theta \mid X) = \text{Beta}\left(\theta \mid 1 + t, \frac{1}{\omega_0^n} - 1 + f\right).$$

Therefore, the Bayesian estimator of θ is $\frac{t+1}{t+f+\frac{1}{\omega_0^n}}$. Finally, we know that the Bayesian estimator of ω, namely the reputation, is $rep = \left(\frac{t+1}{t+f+\frac{1}{\omega_0^n}}\right)^{\frac{1}{n}}$.

Next we consider a more realistic and complicated situation where some producers inject content with varying probabilities of being valid. For example, a producer may inject valid content in the first time interval and inject invalid content in the second equal time interval. If the producer repeats this operation, its reputation will not be low enough to produce a high verification probability when it injects invalid content, leading to a considerable amount of unverified invalid content entering the network. So it is necessary to synthesize the historical and current reputation estimation.

We modify the calculating process from the second MI on. We record the number of valid and invalid batches in history as t and f, respectively, and record the number of valid and invalid batches in the current MI as t_{cur} and f_{cur}, respectively. Compute the Bayesian estimation of rep in the whole time as:

$$rep_{whl} = \left(\frac{t+1}{t+f+\frac{1}{\omega_0^n}}\right)^{\frac{1}{n}}. \tag{7}$$

Compute the maximum likelihood estimation of rep in the current MI as:

$$rep_{cur} = \left(\frac{t_{cur}}{t_{cur}+f_{cur}}\right)^{\frac{1}{n}}. \tag{8}$$

To show the degree to which historical reputation differs from current reputation, we compute the fluctuation factor ff as:

$$ff = \frac{max(rep_{whl}, rep_{cur})}{min(rep_{whl}, rep_{cur})}. \tag{9}$$

Define the weight of current reputation as λ. Since the higher ff is, the greater λ should be, we let their relation be:

$$\lambda = max\left(1, log_\rho ff\right), \tag{10}$$

where ρ is the smallest value of ff that makes λ equal to 1. We can get the ultimate estimation of reputation as:

$$rep = (1-\lambda)rep_{whl} + \lambda rep_{cur}. \tag{11}$$

This formula enables us to estimate the reputation of a producer, regardless of its probability of injecting valid content being constant or variable.

Algorithm 3 shows how the edge router estimates the producer's reputation. Individual verification is conducted during the first monitor interval (MI), and t_0 and f_0, representing the count of valid and invalid packets, respectively, are recorded. Then equation (6) is used to compute ω_0 to get the initial reputation and prior distribution (line 4). Batch verification is carried out during each subsequent MI, and the number of valid and invalid batches both in history and in the current MI are recorded. Due to the potential variation in batch sizes across MIs, the current batch size needs standardization, and equation (7) needs modification. We use the modified equation (7) and equation (8) to compute rep_{whl} and rep_{cur} (lines 7-10). Finally equation (11) is used to get the ultimate reputation (lines 11-13).

```
Algorithm 3: Reputation Estimation
  Input: The number of valid packets $t_0$ and invalid
         packets $f_0$ in the first MI, the number of valid
         batches $t_{cur}$, invalid batches $f_{cur}$ and the batch
         size $n$ in the current MI, the standardized batch
         size $N$, the maximum acceptable fluctuation
         factor $\rho$, the current time $now$.
  Output: Reputation $rep$.
1  t=0;
2  f=0;
3  if $now$ in the first MI then
4      $\omega_0 = \frac{t_0}{t_0+f_0}$;
5      return $\omega_0$;
6  else
7      $rep_{cur} = \left(\frac{t_{cur}}{t_{cur}+f_{cur}}\right)^{\frac{1}{n}}$;
8      $t = t + t_{cur} \times \frac{n}{N}$;
9      $f = f + f_{cur} \times \frac{n}{N}$;
10     $rep_{whl} = \left(\frac{t+1}{t+f+\frac{1}{\omega_0^N}}\right)^{\frac{1}{N}}$;
11     $ff = \frac{max(rep_{whl}, rep_{cur})}{min(rep_{whl}, rep_{cur})}$;
12     $\lambda = max\left(1, log_\rho ff\right)$;
13     $rep = (1-\lambda)rep_{whl} + \lambda rep_{cur}$;
14     return $rep$;
15 end
```

G. Probabilistic Verification Module in Intermediate Routers

The intermediate router performs probabilistic verification of data packets to block fake content further. Upon receiving an interest packet, the edge router calculates the verification probability $p_{ir} = 1 - \gamma^{\frac{1}{n_{ir}}}$ for the corresponding data packet, where n_{ir} is the number of intermediate routers that the interest packet has passed through, and γ is a security parameter. Each intermediate router on the path verifies the data packet with this same probability to balance the load.

The intermediate router performs individual verification rather than batch verification for two reasons: the lower traffic density in intermediate routers makes forming batches slow, and individual verification can accurately identify fake content.

H. Extended Reputation Mechanism

We introduce a simple extended reputation mechanism as a supplement to the previous scheme, to ensure that our scheme can effectively mitigate content poisoning attacks with low overhead even when compromised routers generate fake content. The key idea is to make intermediate routers act similarly to edge routers, since their neighbor nodes are no longer trusted. As the previous scheme shows, the intermediate router also performs the probabilistic passing process, individual/batch verification process, determining process and reputation estimation process to detect fake content at low cost. An intermediate router will disconnect from the neighbor node whose reputation falls below the preset threshold, and choose the neighbor node with the highest reputation as the top priority option for future requests.

V. Security analysis

A. Signature Unforgeability

An adversary $\mathcal{A}$ cannot forge a signature to pass individual verification or batch verification used in our scheme.

1) Individual Verification: Given message M and public parameters, $\mathcal{A}$ must obtain the secret key sk, namely pk^{msk} to pass the individual verification. Because $pk \in G$, pk can be written as g^b, and sk can be written as $g^{msk \times b}$. Besides, mpk is known to be g^{msk}. This means $\mathcal{A}$ must compute $g^{msk \times b}$ from $\{g, g^b, g^{msk}\}$, which contradicts ***DH Assumption***. Thus, $\mathcal{A}$ cannot forge a signature that can pass the individual verification.

2) Batch Verification: We prove that if a signature passes the batch verification, it can pass the individual verification.

Lemma 1. Choose exponents δ_i of l bits. G_T is a multiplicative cyclic group with the prime order q, and g is the generator of G_T. If $\prod_{j=1}^{n} A_j^{\delta_j} = \prod_{j=1}^{n} Y_j^{\delta_j}$, where $A_j \in G_T$ and $Y_j \in G_T$, $A_j = Y_j$ doesn't hold for all $j \in [1, n]$ with probability at most 2^{-l}.

Proof. Since $A_j \in G_T$ and $Y_j \in G_T$, they can be written as $e(g, g)^{a_j}$ and $e(g, g)^{y_j}$, where $a_j, y_j \in \mathbb{Z}_q^*$. We have:

$$\begin{aligned} &\prod_{j=1}^{n} A_j^{\delta_j} = \prod_{j=1}^{n} Y_j^{\delta_j} \\ \Longrightarrow &\prod_{j=1}^{n} e(g, g)^{a_j \times \delta_j} = \prod_{j=1}^{n} e(g, g)^{y_j \times \delta_j} \\ \Longrightarrow &e(g, g)^{\sum_{j=1}^{n} a_j \times \delta_j} = e(g, g)^{\sum_{j=1}^{n} y_j \times \delta_j}. \end{aligned}$$

Define $\beta_j = a_j - y_j$. So $\sum_{j=1}^{n} \beta_j \delta_j = 0 \ (mod\ q)$.

Now assume that at least one of the individual equations does not hold. Without loss of generality we assume that $A_1 \neq Y_1$. That is to say, $\beta_1 \neq 0$. Since q is a prime, we can find the inverse of β_1, denoted γ_1, such that $\beta_1 \gamma_1 = 1 (mod\ q)$. So $\delta_1 = -\gamma_1 \sum_{j=2}^{n} \delta_j \beta_j \ (mod\ q)$.

Let event E occurs if $A_1 \neq Y_1$ but $\prod_{j=1}^{n} A_j^{\delta_j} = \prod_{j=1}^{n} Y_j^{\delta_j}$. Let $\Delta = (\delta_2, ..., \delta_n)$, and let $|\Delta|$ be the number of possible values for this vector. We can get that given a fixed vector Δ, there is exactly one value of δ_1 that will make event E happen. In other words, given a random δ_1, the probability of E is $Pr[E|\Delta] = 2^{-l}$. If we pick δ_1 randomly and sum over all possible choices of Δ, we have:

$$\begin{aligned} Pr[E] &\leq \sum_{i=1}^{|\Delta|} Pr[E|\Delta] \times Pr[\Delta] \\ &= \sum_{i=1}^{2^{l(n-1)}} 2^{-l} \times 2^{-l \times (n-1)} = 2^{-l}. \end{aligned}$$

Therefore, $A_j = Y_j$ doesn't hold for all $j \in [1, n]$ with probability at most 2^{-l}, which is negligible since the value of l is set as 80 in our algorithm. ■

If an adversary $\mathcal{A}$ can forge a signature that can pass the batch verification, it holds that

$$e(\prod_{j=1}^{n} S_{j,2}^{\delta_j}, g) = e(\prod_{j=1}^{n} pk_j^{a_j \times \delta_j}, mpk) \times \prod_{j=1}^{n} S_{j,1}^{\delta_j}.$$

Using the properties of the bilinear map, we have:

$$\prod_{j=1}^{n} e(S_{j,2}, g)^{\delta_j} = \prod_{j=1}^{n} e(pk_j^{a_j}, mpk)^{\delta_j} \times \prod_{j=1}^{n} S_{j,1}^{\delta_j}.$$

From Lemma 1, we can further get: for $j \in [1, n]$, $e(S_{j,2}, g) = e(pk_j^{a_j}, mpk) \times S_{j,1}$, which means every signature can pass the individual verification. It indicates that $\mathcal{A}$ can forge a signature that can pass the individual verification, which contradicts our previous proof. Therefore, $\mathcal{A}$ cannot forge a signature that can pass the batch verification.

B. Defense of DoS Attacks

If a malicious producer injects invalid packets at a high speed, routers may be overwhelmed by the verification process, causing a Denial of Service (DoS) attack.

Our scheme can effectively throttle this attack by detecting producer's reputation. First we consider a simple situation where every producer injects content with a constant probability of being valid. After continuously injecting invalid packets for a while, the malicious producer's reputation will drop below β. Then, the edge router will discard all packets from a producer whose reputation is less than β based on the verification probability function (1). As a result, our scheme can quickly detect and defend against DoS attacks launched by malicious producers.

Next we consider a more realistic and complicated situation where some producers inject content with varying probabilities of being valid to evade the defense, such as alternating between valid and invalid content to obtain a higher reputation. However, this effort is useless since our reputation estimation module considers the fluctuation factor. When the difference between historical and current reputation is high , the ultimate reputation we compute is closer to the current reputation. As a result, the producer will get a low reputation and its invalid packets can never be forwarded.

So our scheme can defend against the DoS attack no matter whether the producer injects invalid content with constant probability or varying probability.

C. Negligible ratio of admitted bogus packets

We prove that although there are a few bogus packets admitted into the network and received by users, the ratio is negligible. A bogus packet is admitted if it is forwarded by the edge router and intermediate routers on the path. We analyze the two stages as follows.

The first stage is in the edge router's probabilistic passing module, batch verification module and determining module. Some bogus packets may be forwarded from the edge router to the intermediate router due to the verification probability and the discarding probability. These packets can be divided into two categories: the first category skips the verification process in the probabilistic passing module, and the second category is verified invalid in the batch verification module but not discarded in the determining module. We prove that the ratio of each of these categories does not exceed η.

Theorem 1. Assume an edge router receives data packets from a content producer who publishes genuine packets with probability rep. In the probabilistic passing module, when the edge router verifies packets with probability p_v in function (1), the ratio of bogus packets that skip the verification process and then are forwarded to the next node will not exceed η.

Proof. Suppose the edge router receives Q packets, in expectation there are $Q(1-p_v)$ packets that are not verified and $Q(1-rep)$ bogus packets. The probability of missing exactly k invalid packets is:

$$P(k) = \frac{\binom{(1-rep)Q}{k}\binom{rep\times Q}{(1-p_v)Q-k}}{\binom{Q}{(1-p_v)Q}},$$

where $k \in [0,K]$ and $K = \min\{(1-rep)Q,(1-p_v)Q\}$. Therefore, the mean of k, referred to as n_1, is

$$n_1 = E(k) = \sum_{k=0}^{K} k \times P(k) = Q(1-p_v)(1-rep).$$

If $rep \geq 1-\frac{\eta}{1-\alpha}$, we have:

$$n_1 \leq Q(1-\alpha)\left(\frac{\eta}{1-\alpha}\right) = \eta Q.$$

If $rep < 1-\frac{\eta}{1-\alpha}$, we have:

$$n_1 \leq Q\left(\frac{\eta}{1-rep}\right)(1-rep) = \eta Q.$$

Therefore, $n_1 \leq \eta Q$ it always holds all the time. The ratio of bogus packets that skip the verification process and then are forwarded to the next node does not exceed η. ■

Theorem 2. Assume an edge router receives data packets from a content producer who publishes genuine packets with probability rep. When the edge router performs batch verification for data packets and discards packets in the invalid batch with probability p_d in function (3), the ratio of bogus packets that are verified but not discarded will not exceed η.

Proof. Suppose the edge router receives Q packets, in invalid batches, in expectation there are $Qp_v(1-rep)$ bogus packets , $Qp_v(rep-rep^n)$ genuine packets, and $Qp_v(1-rep^n)(1-p_d)$ packets not discarded. The probability of not discarding exactly k bogus packets is:

$$P(k) = \frac{\binom{p_v(1-rep)Q}{k}\binom{p_v(rep-rep^n)\times Q}{p_v(1-rep^n)(1-p_d)Q-k}}{\binom{p_v(1-rep^n)Q}{p_v(1-rep^n)(1-p_d)Q}},$$

where $k \in [0,K]$ and $K = \min\{p_v(1-rep)Q, p_v(1-rep^n)(1-p_d)Q\}$. Therefore, the mean of k, referred to as n_2, is

$$n_2 = E(k) = \sum_{k=0}^{K} k \times P(k) = Qp_v(1-rep)(1-p_d).$$

If $rep \geq 1-\frac{\eta}{1-\alpha}$, we have:

$$n_2 \leq Q\alpha(1-rep)\left(\frac{\eta}{\alpha(1-rep)}\right) = \eta Q.$$

If $rep < 1-\frac{\eta}{1-\alpha}$, we have:

$$n_2 \leq Q\left(1-\frac{\eta}{1-rep}\right)(1-rep)\left(\frac{\eta}{1-rep-\eta}\right) = \eta Q.$$

Therefore, $n_2 \leq \eta Q$ holds all the time. The ratio of bogus packets that are verified but not discarded and then forwarded to the next node does not exceed η. ■

The total number of bogus packets that are forwarded from the edge router to the next node is $n_{all} = n_1 + n_2 \leq 2\eta Q$. So in the first stage, the ratio of bogus packets that are forwarded from the edge router to the next node does not exceed 2η.

The second stage is in the probabilistic verification module in intermediate routers. Some bogus packets may be forwarded from an intermediate router to the next node due to the verification probability.

Each intermediate router verifies packets with a certain probability $1-\gamma^{\frac{1}{n_h}}$, where n_h is the number of intermediate routers. Since the verification probability of every intermediate router is independent, the ratio of bogus packets that are not verified by any intermediate router is

$$\left[1-\left(1-\gamma^{\frac{1}{n_h}}\right)\right]^{n_h} = \gamma.$$

Combining the analysis of the above two stages, the ratio of bogus packets that are forwarded by the edge router and all the intermediate routers on the path is $2\eta\gamma$. Its value is as small as 0.008 in our experiment. With so few admitted bogus packets, their impact on the network and users is negligible. We can defend against content poisoning attacks.

VI. Performance Analysis

A. Algorithm Implementation

We implement the batch verification algorithm by use of GNU Multiple Precision Arithmetic (GMP) library and Pairing-Based Cryptography (PBC) library. All the experiments are conducted on the Ubuntu 20.04 LTS with a 2.7 GHz Intel Core i5 processor and 16 GB RAM.

We measure the execution time of the batch verification algorithm and RSA verification algorithm (including checking the certificate). In Fig. 3, it takes 4.9 ms to verify a signature in RSA and 4.1 ms in batch verification. The verification time of both algorithms increases linearly as the number of signatures increases, but the growth rate of RSA is much higher than ours. It is clear that batch verification is always more efficient than RSA, especially when verifying a large number of signatures.

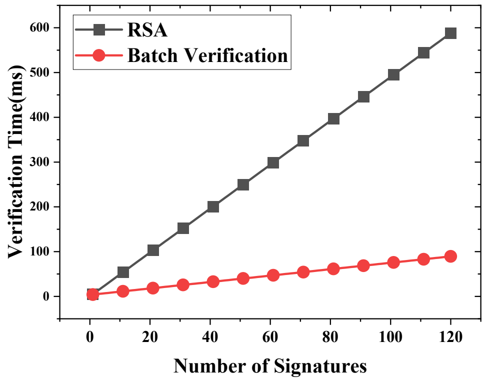

Fig. 3. Algorithm Cost.

To present the difference in communication overhead, we compare the extra payload introduced by the two signature algorithms. In Table I, to achieve the same degree of security, the signature length in batch verification is 320 bits, which is much smaller than 1024 bits in RSA. Moreover, RSA requires a certificate, while batch verification doesn't need it because we adopt the identity-based signature algorithm.

TABLE I
EXTRA PAYLOAD

Scheme	Signature Length	Certificate
RSA	1024 bits	yes
Batch Verification	**320 bits**	**no**

B. Network Simulation

We simulate our complete scheme RPBV in ndnSIM 2.9. We compare it with individual verification (verifying every packet individually, denoted Individual). We also compare it with ESS [14] as it's the most efficient batch verification scheme for data packets in NDN. In ESS, every router performs batch verification and discards all invalid batches. The topologies are generated by BRITE using the Waxman model, which has 1000 nodes and 1865 edges. The link has a 100 Mbps bandwidth and a 0.5 ms delay. One thousand pieces of content are requested, and their distribution follows the Zipf-Mandelbrot distribution with parameter q = 0.7 and parameter s=0.1. Users request content at a rate of 10000 pieces per second, and each piece of content is 1 KB. Parameter choice in our scheme is as follows: MI = 0.1 s, $\alpha = 0.1$, $\beta = 0.2$, $\eta = 0.005$, $\rho = 4$, $M = 100$, $N = 10$, $\gamma = 0.8$.

Fig. 4 shows the computation complexity for verifying files of different sizes. For all these schemes, the computation complexity increases with the growing file size. Among them, individual verification takes far more time to verify packets than the other two. RPBV has the lowest cost due to the use of batch verification and probabilistic verification. It decreases about 32.1% verification time compared to ESS, and 81.7% verification time compared to individual verification.

Fig. 5 shows the accuracy of reputation estimation. We estimate the reputation of different producers whose probability of sending valid content varies from 0.2 to 1 and we present the average estimation result within 10 seconds. In each condition, the estimated reputation is basically the same as the actual reputation, with the standard deviation less than 0.8%. So our scheme can obtain a fairly accurate value of reputation.

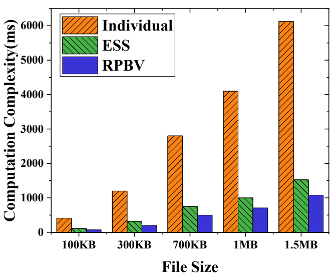

Fig. 4. Computation Complexity.

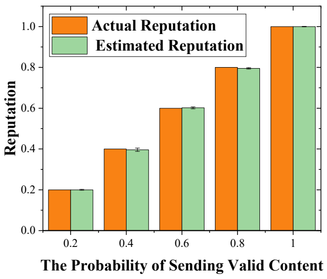

Fig. 5. Reputation Accuracy.

Fig. 6 shows the file retrieval time for different file sizes. As the file gets bigger, it takes more time to obtain it in these schemes. Obviously, RPBV outperforms ESS and individual verification, which is about 12.3% lower than ESS and 44.5% lower than individual verification. Fig. 7 measures the file retrieval time for different content distributions. The file size is 100 KB. As parameter s in Zipf-Mandelbrot distribution increases, popular content is requested more frequently and thus is more probable to be cached. Therefore, more interest packets can be satisfied by intermediate routers with less retrieval time, which gives rise to the decreasing trend of the curves. RPBV outperforms ESS and individual verification by about 12.7% and 44.1% on average, respectively. Fig. 8 shows the file retrieval time for different hop counts. The file size is 100 KB. The time increases with the increase of the hop count, and these schemes have almost the same increasing rate. RPBV performs best with about 61.2 ms reduced than ESS and around 342.3 ms reduced than individual verification.

VII. Conclusion

In this paper, we proposed RPBV, a reputation-based probabilistic batch verification scheme for named data networking. In RPBV, we designed a dynamic and adaptive probabilistic verification method based on the estimated reputation to decrease the number of verification operations. Besides, we adopted an efficient batch verification algorithm to reduce overhead greatly. Furthermore, we implemented our scheme on the edge router to avoid repeated verification and make the probabilistic batch verification mechanism work better. Security analysis shows that RPBV guarantees

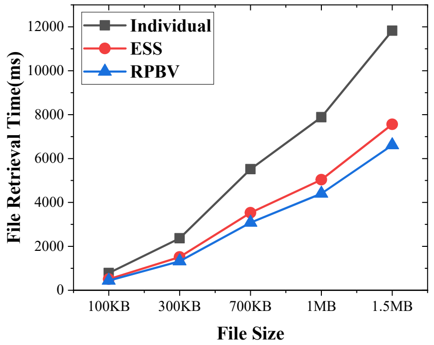

Fig. 6. Retrieval Time vs. File Size

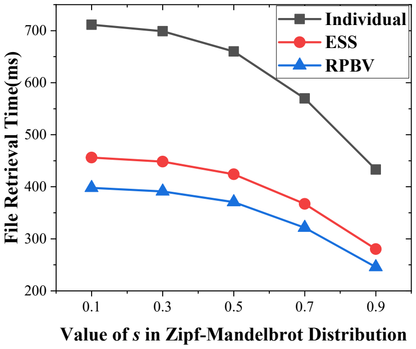

Fig. 7. Retrieval Time vs. Content Distribution

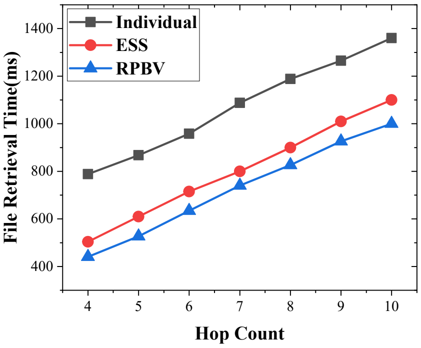

Fig. 8. Retrieval Time vs. Hop count

signature unforgeability and resist content poisoning attacks and DoS attacks. Experimental evaluation shows that our scheme achieves lower overhead and shorter content retrieval time.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61972371, and Youth Innovation Promotion Association of the Chinese Academy of Sciences (CAS) under Grant No. Y202093.

References

[1] L. Zhang, A. Afanasyev, J. Burke *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[2] R. Tourani, S. Misra, T. Mick, and G. Panwar, "Security, privacy, and access control in information-centric networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 566–600, 2017.

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

[4] K. Xue, P. He, J. Yang, Q. Xia, and D. S. Wei, "SCD2: Secure content delivery and deduplication with multiple content providers in information centric networking," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1849–1864, 2022.

[5] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS and DDoS in named data networking," in *Proceedings of the 2013 IEEE International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2013, pp. 1–7.

[6] K. Xue, P. He, X. Zhang *et al.*, "A secure, efficient, and accountable edge-based access control framework for information centric networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1220–1233, 2019.

[7] Q. Li, X. Zhang, Q. Zheng, R. Sandhu, and X. Fu, "Live: Lightweight integrity verification and content access control for named data networking," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 308–320, 2015.

[8] C. Ghali, G. Tsudik, and E. Uzun, "In content we trust: Network-layer trust in content-centric networking," *IEEE/ACM Transactions on Networking*, vol. 27, no. 5, pp. 1787–1800, 2019.

[9] D. Qu, G. Lv, S. Qu, H. Shen, Y. Yang, and Z. Heng, "An effective and lightweight countermeasure scheme to multiple network attacks in NDN," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 515–528, 2022.

[10] B. K. Saha and S. Misra, "Mitigating NDN-based fake content dissemination in opportunistic mobile networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1375–1386, 2020.

[12] W. Cui, Y. Li, Y. Xin, and C. Liu, "Feedback-based content poisoning mitigation in named data networking," in *Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2018, pp. 759–765.

[11] S. DiBenedetto and C. Papadopoulos, "Mitigating poisoned content with forwarding strategy," in *Proceedings of the 2016 IEEE International Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2016, pp. 164–169.

[13] M. Baugher, B. Davie, A. Narayanan, and D. Oran, "Self-verifying names for read-only named data," in *Proceedings of the 2012 IEEE International Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2012, pp. 274–279.

[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 Transactions on Computational Social Systems*, vol. 8, no. 5, pp. 1108–1120, 2021.

[15] R. Li, H. Asaeda, and J. Wu, "DCAuth: Data-centric authentication for secure in-network big-data retrieval," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 15–27, 2018.

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

[17] K. Xue, J. Yang, Q. Xia *et al.*, "CSEVP: A collaborative, secure, and efficient content validation protection framework for information centric networking," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1761–1775, 2022.

[18] M. Anisetti, C. A. Ardagna, F. Berto, and E. Damiani, "A security certification scheme for information-centric networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2397–2408, 2022.

[19] P. Kar, S. Misra, A. K. Mandal, and H. Wang, "SOS: NDN based service-oriented game-theoretic efficient security scheme for IoT networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3197–3208, 2021.

[20] T. Nguyen, H.-L. Mai, G. Doyen *et al.*, "A security monitoring plane for named data networking deployment," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 88–94, 2018.

[21] T. Nguyen, X. Marchal, G. Doyen, T. Cholez, and R. Cogranne, "Content poisoning in named data networking: Comprehensive characterization of real deployment," in *Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017, pp. 72–80.

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

[23] V. Sivaraman and B. Sikdar, "A defense mechanism against timing attacks on user privacy in ICN," *IEEE/ACM Transactions on Networking*, vol. 29, no. 6, pp. 2709–2722, 2021.

[24] F. Hess, "Efficient identity based signature schemes based on pairings," in *Proceedings of the 2002 Springer International Workshop on Selected Areas in Cryptography*. Springer, 2002, pp. 310–324.

[25] A. L. Ferrara, M. Green, S. Hohenberger, and M. Ø. Pedersen, "Practical short signature batch verification," in *Proceedings of the 2009 Springer Cryptographers' Track at the RSA Conference*. Springer, 2009, pp. 309–324.

[26] M. Zhang, Y. Cheng, X. Deng *et al.*, "Accelerating transactions relay in blockchain networks via reputation," in *Proceedings of the 2021 IEEE/ACM International Symposium on Quality of Service (IWQOS)*. IEEE, 2021, pp. 1–10.