A Cache Privacy Protection Mechanism based on Dynamic Address Mapping in Named Data Networking

Yi Zhu[1*], Haohao Kang[1] and Ruhui Huang[1]

[1] School of Computer Science and Communication Engineering, Jiangsu University,
Zhenjiang, China
[e-mail: zhuyi@ujs.edu.cn, 2211608006@stmail.ujs.edu.cn, ruhuihuang@ujs.edu.cn]
*Corresponding author: Yi Zhu

Received September 3, 2017; revised March 2, 2018; revised April 3, 2018; accepted August 9, 2018; published December 31, 2018

Abstract

Named data networking (NDN) is a new network architecture designed for next generation Internet. Router-side content caching is one of the key features in NDN, which can reduce redundant transmission, accelerate content distribution and alleviate congestion. However, several security problems are introduced as well. One important security risk is cache privacy leakage. By measuring the content retrieve time, adversary can infer its neighbor users' hobby for privacy content. Focusing on this problem, we propose a cache privacy protection mechanism (named as CPPM-DAM) to identify legitimate user and adversary using Bloom filter. An optimization for storage cost is further provided to make this mechanism more practical. The simulation results of ndnSIM show that CPPM-DAM can effectively protect cache privacy.

Keywords: Cache privacy, Privacy protection, Bloom filter, Cache snooping, NDN

http://doi.org/10.3837/tiis.2018.12.027 **ISSN : 1976-7277**

1. Introduction

With the rapid development of Internet, content services have gradually become the body of network application. In this situation, the traditional IP architecture based on host-to-host cannot satisfy current network requirements. Since 2006, several new network architectures for next generation Internet have been proposed, including data-oriented network architecture (DONA)[1], publish-subscribe internet routing paradigm (PSIRP)[2], named data networking(NDN)[3,4], etc. Among these projects, NDN is more representative and attract more attentions.

Caching mechanism is the core design of NDN. Each NDN node has a content store (CS) for caching passed data packets. User can obtain requested data packet from the nearby node using name routing. This design can reduce redundant transmission, accelerate content dissemination and alleviate network congestion. But as an open data exchange platform, cache mechanism also leads to several new security problems while promoting network performance. One important security threat caused by caching mechanism is cache privacy leakage [5-7]. In this attack, adversary send requests for a specific content and measure its round-trip time (RTT), and then determine whether the target content is stored in the closest router or not. If the target exists, that means some legitimate users nearby adversary recently requested the same content. By measuring retrieve time, adversary can infer the hobby of his/her neighbors for privacy contents. This snooping attack behavior is also called as timing attack.

To solve this problem, most current researches focus on how to increase time or space ambiguity of cached content, such as dynamically increasing response delay [8–11]. But the protection of cache privacy of these solutions is at the expense of decreasing network delivery capacity. In fact, the fundamental and effective countermeasure is to identify the request of adversary and then deal with it differently from legitimate user. But it is difficult to realize in NDN due to the interest packet of NDN doesn’t carry any address information.

In this paper, we propose a new mechanism to identify legitimate user and adversary using Bloom filter, then protect cache privacy by adding extra delay for the response of adversary. We name it as cache privacy protection mechanism based on dynamic address mapping (CPPM-DAM). This mechanism utilizes a group of hash functions with specific permutation to map the name of requested content to an address of multi-dimensional matrix. When users send interest packet, they should first select a permutation indicator of hash functions and append it after content name. Because different permutation will lead to different mapping address even for the same content name, the requests from different users can be distinguished by NDN router. Simulation results of ndnSIM show that the CPPM-DAM can protect against privacy leakage effectively.

The contributions of this paper can be summarized as two-fold.

First, we propose a novel mechanism to protect cache privacy. Today, limited to the special design of NDN – no address information within the interest packet, how to detect the timing attack is difficult. The traditional solution is to set radom miss responses when hit event occurs. But this idea will seriously decrease the effectiveness of NDN.Our idea is different and novel. The proposed mechanism by us protects cache privacy through distinguishing the legitimate user and adversary. And we don’t add any address in interest packet, just use Bloom filter to realize the identification.

Second, we analyze the impact of attack rate in simulation and then show that lower attack rate will lead to higher risk of cache privacy leakage. In previous works, researchers mainly focused on other impact factors, such as cache size and content popularity. But we find the attack rate is a more important factor for timing attack. When the interval between two successive requests of adversary become larger, adversary will avoid to hit the content retrieved by himself/herself before. So, select a suitable attack rate is the key for realizing a successful timing attack. On the other hand, deeply understand the impact of attack rate will help to disclose the intrinsic characteristic of timing attack.

The rest of this paper are organized as follows. Section 2 introduces the implementation of cache privacy attack and the related works in this field. Section 3 describes the design of CPPM-DAM in detail. Section 4 presents the simulation results. Section 5 discusses the storage optimization of CPPM-DAM. Finally, we conclude the paper in section 6.

2. Cache privacy snooping

2.1 Attack description

Timing measurement is the primary way for snooping cache privacy in NDN [8, 9]. Now we give an example to illustrate the snooping process. As shown in **Fig. 1**, the topology of example is a L-level cascaded network, A_1 is the adversary and U_1 is the legitimate user, they are located within the same edge router R_1. By measuring the round-trip time (RTT), A_1 can infer whether U_1 recently requested the specific content or not. First, A_1 measures the round-trip time from source server (defined as RTT_S) by requesting a low-popularity content which is not cached in network, then measures the round-trip time from the nearby router R_1 (defined as RTT_C) by requesting the same content again. After the above preparation, A_1 further snoops the victim content by observing its round-trip time (defined as RTT_A). According to snooping result, A_1 will reach a decision from the following three available cases.

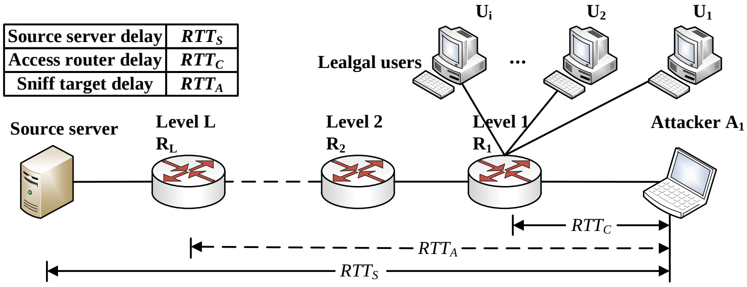


Fig. 1. Cache privacy snooping in NDN

(1) If $|RTT_A - RTT_C| < \varepsilon, (\varepsilon \to 0)$, the target content obviously exists in the edge router R_1. Then adversary A_1 can infer that its neighbors have requested this content before a short time interval, where the time interval is equal to the average residence time of cached content in CS.

(2) If $RTT_A > RTT_C$ and $RTT_A < RTT_S$, the target content does not exist in R_1, but it should be cached in network somewhere (such as R_2). In this case, A_1 can infer that its neighbors requested the target before a relative long time, but not recently.

(3) If $|RTT_A - RTT_S| < \varepsilon$, the target content should be responded from source server. So A_1 can infer that its neighbors didn't request the target content during a long time.

For adversary, timing attack can effectively track the visiting information of neighbor user within one hop. Even if the adversary has multiple neighbors, it can also mine the privacy of legitimate user with some prior knowledge.

2.2 Related works

In literature [8], three important security problems existing in NDN were described in detail, including cache pollution, PIT (Pending Interest Table) flooding and cache privacy snooping. For timing attack, authors suggested to add an extra response delay to defense it, where the extra delay should be equal to the retrieve time from source server. The authors of literature [9] also suggested similar solutions of adding extra random relay or cryptographic delay. Gergely et al. [10] theoretically analyzed this problem and then proposed to randomly generates k miss responses for requested content to protect cache privacy, this method named as Random First k Delay (RFKD) is a typical method among current solutions. Although the above two methods can effectively protect cache privacy, the extra delay will counteract the advantages brought by cache mechanism in NDN.

Based on literature [10], Mohaisen et al. tried to identify the origin of interest packet by user id or face id [11,12], but the identification of user id is difficult to realize in NDN. For the proposal of face id, if router maintains the history of each cached content, it will result in heavy overhead.

Recently, some researchers began to explore the method of access control. Silva et al. [13] & Ion et al. [14] designed an access policy within the data by introducing attribute-based encryption. But it only protects the content privacy. In the suggestion of Tao Chen et al. [15], when an interest packet is received, router will deny the Interest from the unauthorized user after the failure of finding the user from active user table. Although this method is effective, its computation cost is too heavy and current routers cannot afford it even if facing normal traffic.

Other some works from different perspective also provide valuable reference. Lauinger et al. [16-17] pointed out the content privacy is closely related to its popularity, the content with lower popularity always owns higher privacy. So, router should first distinguish non-privacy content and privacy content, and then only carry out protection strategy for privacy content. Chaabane et al. [18] discussed privacy issues in NDN and pointed out that collaborative caching and probabilistic caching are also potential solutions. Arianfar et al. [19] presented an approach of forcing the adversary to perform sizable computations to reconstruct each request. This approach does not provide ideal privacy protection, but it makes adversary more hardly to snoop cache privacy. The idea of Xingwen Zhao et al. [20] is also close to [19]. In their scheme, the users are anonymous and the shared keys are valid within a specified time period so the adversary is infeasible to precompute the name matching datasets during the valid time period of the key.

To clearly analyze the existing solutions, **Table 1** gives their strengths and limitations.

Table 1. The Strengths and Limitations of existing solutions

Mothed	**Feature**	**Cache privacy protection**	**Content retrieved delay**	**Limitations**
Content specific delay[8,9]	Add extra delay before reply	Strong	Very high	Caching advantage of NDN lost totally
RFKD[10]	Generates random misses for requested content	Enough large misses can achieve strong protection	Very high	Caching advantage of NDN lost totally

User id recognition[11,12]	Distinguish legitimate user by user id	Strong	Normal	Non-compliance with the rule of NDN
Face id recognition[11,12]	Distinguish legitimate user by face id	Medium	High	Face id isn't an accurate indicator for identification
ABE[13,14]	Encrypt data using the attributes provided by specific user	Weak	Normal	Effective for content privacy
Probabilistic-Based Access Control[15]	Find requester from active user table	Strong	Normal	Computation cost is too heavy
Collaborative caching[18]	Increase space ambiguity of cached content	Medium	High	Increase the complexity of network
Probabilistic caching[18]	Increase time ambiguity of cached content	Weak	High	Only provide limitation protection of cache privacy

From these literatures, we understand that the security of cache privacy and the effectiveness of caching mechanism are a pair of contradictions. Undoubtedly, by generating extra delay or multiple miss responses, it is possible to realize perfect protection of cache privacy, but the advantages of caching mechanism will be lost totally.

Furthermore, we can also see that how to recognize the identity of requester is the key point for cache privacy protection, but it is difficult for current NDN. In next section, we will propose a novel method to solve this problem. Different from current works, we don't insert user id into interest packet or encrypt interest packet, just use Bloom filter to distinguish the adversary and legitimate user.

3. Design of CPPM-DAM

In this section, a dynamic address mapping mechanism for protecting cache privacy is given, we called as CPPM-DAM in short. The core idea of CPPM-DAM includes two points. First point is to map the content name of received interest packet to an address of multi-dimensional matrix by a Bloom filter [21], this Bloom filter is consisted of a group of hash functions. Second point is to append a tag to content name of interest packet, which is used to indicate the permutation of hash functions of Bloom filter. Because different permutation indicators will lead to different mapping address for the same content name, NDN router can identify different requester when they adopt different permutation indicators. Next, we will introduce the detailed design of CPPM-DAM.

(1) Each NDN router maintains a Bloom filter. The received interest packet must first be filtered by the Bloom filter, then searched within CS, PIT, FIB (Forwarding Information Base) in turn. This Bloom filter consists of n independent hash functions (defined as $h_1, h_2, \cdots, h_n$) and K n- dimensional matrices $Y^k\left(y_{b_1,b_2,\cdots,b_n}\right)$, where K denotes total classes of content popularity ($1 \le k \le K$), each class corresponds to an n-dimensional matrix. Using a group of hash functions, each content name can be mapped into a specific address of n-dimensional matrix according to its class. Let X^k denotes the set of content

names of the k th class, $B^k = \left\{ b_i^k, 1 \le i \le n \right\}$ denotes the address coordinates in matrix Y^k, then

$$h: X^k \to B^k \Rightarrow \begin{cases} b_1^k = h_1(x) \bmod B_{\max}^k \\ b_2^k = h_2(x) \bmod B_{\max}^k \\ \quad\vdots \\ b_n^k = h_n(x) \bmod B_{\max}^k \end{cases} \qquad (1)$$

Where, $B_{\max}^k$ is the maximum size of the k th matrix, $1 \le b_i^k \le B_{\max}^k, 1 \le i \le n$; x is a name of received interest packet.

(2) Considering that there is $n!$ permutations of n hash functions, we use $s_1 \sim s_{n!}$ to denote these permutations. To add the permutation indicator to interest packet, a new string field is appended to the content name. For an example, if the original content name is "/ ujs.edu.cn/Computer_Networks/Lecture_1.mpeg", then the modified name should be "/ ujs.edu.cn/Computer_Networks/Lecture_1.mpeg/~s2", where "~s2" at the end of name indicates that the router must adopt the hash functions permutation of s_2 for calculating the mapping address. In this mechanism, user can randomly select a permutation when generating first request for a specific content, and then later requests for the same content must keep the same permutation selection. Once user changes his/her request target content, a new permutation is randomly generated and appended to interest packet.

(3) When NDN router receives interest packet, it firstly extracts the content name and permutation indicator from the packet. Then calculates the mapping address of requested name according to the permutation indicator. Because there is $n!$ different permutations, even if content name is same, there will be $n!$ available mapping addresses. When a new request for content in class k arrives, the value of its mapping address in corresponding matrix will be set to 1, namely, $x^k \to \left\{ b_1^k, b_2^k, \cdots, b_n^k \right\}, y^k_{b_1^k, b_2^k, \cdots, b_n^k} = 1$. When the content is removed from CS because of cache replacement, the values of all available addresses of this name should be cleared to 0.

(4) After calculating the mapping address, router further checks the value of this address. A none-zero value indicates that the current requester has previously requested the same content and this content is existing in CS. In this case, router should directly return the data packet to requester. On the other hand, a zero value indicates that the requester is a freshman for this content or this content does not exist in CS. So, the router should delay γ_C before returning the data packet to requester, where γ_C is the retrieve time from source server. **Table 2** gives the detailed processing of CPPM-DAM when router receives a interest packet.

Table 2. The Interest Packet Processing of CPPM-DAM

```
Algorithm: Interest Packet Processing of CPPM-DAM
While Interest Packet Received ()
    Extracts the content name and permutation indicator;
    Calculates the mapping address $\{b_1^k, b_2^k, \cdots, b_n^k\}$ according to the permutation indicator using equation (1);
```

```
        **If** (The value of this address = 1)
            Router directly returns the data packet;
        **Else**
            Router delays $\gamma_C$ before returning the data packet;
            The value of this address is set to 1;
        **End if**
    **End while**
```

Fig. 2 clearly shows the principle of CPPM-DAM. Benefit from adding the permutation indicator into interest packet, NDN router can distinguish different requesters with high probability. Even if the adversary requests the same content name as legitimate user, the router can distinguish them due to adversary don't know the permutation indicator selected by legitimate user. Only if the adversary selects the same permutation indicator as legitimate user, can the adversary successfully snoop cache privacy. But this collision is an event with small probability. Obviously, the privacy leakage probability will become smaller while the number of hash functions increase. After identification of new requester, an extra delay is added to response for protecting the cache privacy.

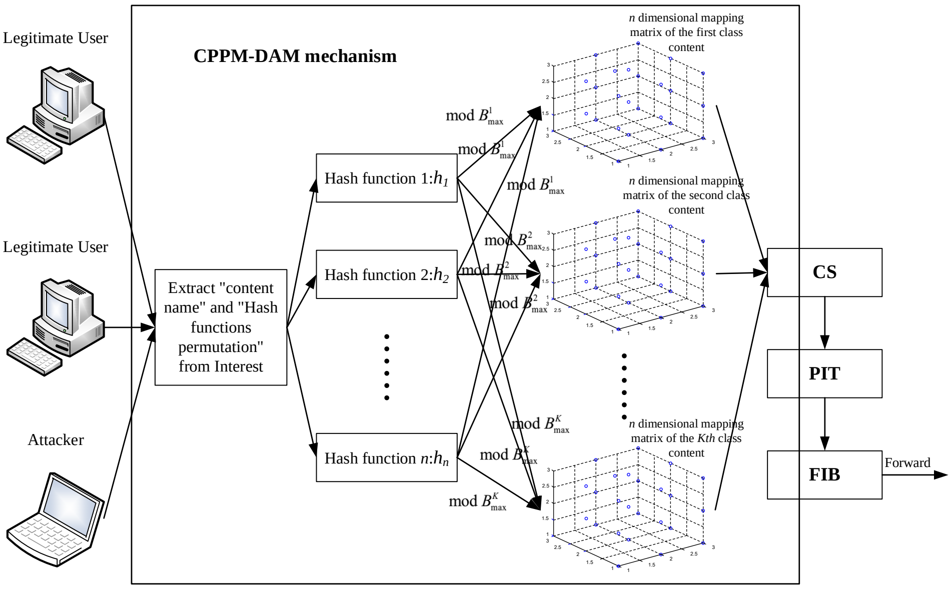

Legitimate User
Legitimate User
Attacker
CPPM-DAM mechanism
Extract "content name" and "Hash functions permutation" from Interest
Hash function 1:h_1
Hash function 2:h_2
Hash function n:h_n
mod B^1_{max}
mod B^2_{max}
mod B^K_{max}
n dimensional mapping matrix of the first class content
n dimensional mapping matrix of the second class content
n dimensional mapping matrix of the Kth class content
CS
PIT
FIB
Forward

Fig. 2. The principle of CPPM-DAM

The Bloom filter is the core of CPPM-DAM, and the parameters of n and B^k_{max} will directly affect the performance of CPPM-DAM. Next section, we will evaluate the effectiveness and reliability of CPPM-DAM, and then analyze the affecting factors of Bloom filter.

4. Simulation analysis

This section, we compare the performance of CPPM-DAM with the typical solution-RFKD[10]. The simulation tool is ndnSIM[22, 23], it is deployed in a high-performance computing platform with four Intel (R) Xeon (R) CPU E7-4830 and 256GB memory, its operating system is CentOS 6.5. The simulation settings [24] are list as follows.

(1) The network topology is shown in **Fig. 1**. To simplify the analysis, we set the network includes only one router R_1, one source server, a legitimate user U_1 and an adversary A_1. The U_1 and A_1 are located within the access router R_1.

(2) The source server provides contents with $K = 50$ classes, each class contains $m = 2500$ files, and each file has the same size of 1024 bytes.

(3) For naming each content, we use $\mathrm{"class\_id"}$ as the tag of class, use $\mathrm{"item\_id"}$ as the tag of file. For example $\mathrm{"/dst1/class\_id/item\_id/{\sim}s1"}$ is a content name generated in simulation, where $\mathrm{s1}$ is the permutation indicator of hash functions.

(4) For legitimate user, the request generated process is modeled as a Poisson process of intensity λ_U, the contents of class k are requested with probability q_k which obey *Zipf* distribution, where $q_k = c / k^{\alpha}$, $1 \le k \le K$. Furthermore, content items requested in each class obey uniform distribution. So, the arrival rate of requests for class k should be $\lambda_U(k) = \lambda_U q_k$ and the arrival rate of requests for a specific content in class k should be $\lambda_U q_k / m$. In simulation, the parameter of *Zipf* distribution is set as $\alpha = 0.8$ [25], the intensity of Poisson process is set as $\lambda_U = 10^5$ interest packet/s.

(5) For adversary, the requests are only generated for a specific victim class with constant rate. If the attack target is the k th class of content, we define the attack request rate of the adversary as $\lambda_A(k)$. Based on the definition of request rate of legitimate user and adversary, we further define the relative attack rate $\nu = \lambda_A(k) / \lambda_U(k)$. It illustrates the ratio of request rate between A_1 and U_1. A reasonable attack rate is the foundation of successful snooping.

(6) Router R_1 adopts LRU (Least Recently Used)[26] as cache replacement policy, its cache size is defined as $C = 2500$ files.

(7) In the implementation of CPPM-DAM, the xxhash hash algorithm is used[27], which is an efficient non-cryptographic open source hash algorithm. By giving several different seeds, a group of independent hash functions is generated from xxhash hash algorithm.

(8) In RFKD method, for each cached content, the router will randomly generate w miss responses to the first w requests, where w is uniformly distributed on $[0, W]$. With the change of W, the cache privacy protection ability of RFKD will correspondingly change. So, we will test different value of W in simulation.

Considering that the contents cached in CS have two available originations, one is retrieved by the requests of legitimate user, the other is retrieved by the request of adversary. When adversary send requests for attacking, a successful snooping means the hit content is retrieved by legitimate user. If the hit content is retrieved by adversary himself/herself, adversary cannot obtain any privacy information. Based on above consideration, we further define the attack success probability as evaluation indicator in simulation.

Definition 1. (Attack success probability). During a statistics period, for adversary, we define that $H_{hit\_total}(k)$ is the total number of hit event in class k, $H_{hit\_u}(k)$ is the number of hit contents in class k which are retrieved by legitimate user. If we use $p(k)$ to denote the attack success probability of class k, $p(k)$ can be expressed as

$$p(k) = \frac{H_{hit\_u}(k)}{H_{hit\_total}(k)} \qquad (2)$$

Obviously, the attack success probability directly reflects the effect of cache snooping. High attack success probability means effective snooping and serious privacy leakage.

Next, the cache snooping results are derived under no any protection, CPPM-DAM and RFKD. And we evaluate the network performance from two aspects, one is attack success probability, it is used to show the protection ability; another is average RTT of hit contents, it is used to show the effectiveness. The running time of following experiments are all 50 simulation-second, and each experiment is repeated 20 times for statistics.

4.1 Performance of CPPM-DAM and RFKD

To disclose the performance of CPPM-DAM, now we select the first class as snooping target. The parameters of Bloom filter are set as $n = 5$ and $B_{\max}^{k} = 8$. **Fig. 3** and **Fig. 4** show the attack success probability and average RTT under three relative attack rates – 0.5, 1.0, 1.5.

From **Fig. 3**, we can see (1) Without any protection, attacker can snoop cache privacy with high successful probability. It proves that the timing attack is an effective way to damage user's privacy. (2) The attack rate is the key factor affecting the effect of timing attack. The higher the attack rate is, the lower the attack success probability is. If the attacker wants to implement a successful attack, he/she should work with a low and reasonable attack rate. (3) The defense capability of RFKD depends on the range of random misses, it can protect cache privacy satisfactorily if W is large enough. In our simulation, the attack success probability can be controlled less than 20% when we set $W = 5$ .If we continue to increase W , the defense capability of RFKD will be enhanced commensurately. (4) Compared with RFKD, the defense capability of CPPM-DAM is powerful, it can achieve perfect protection of cache privacy. Whatever attack rate changes, the attack success probability under CPPM-DAM is still less than 5%. It is shown that CPPM-DAM is an excellent solution for the problem of cache privacy of NDN.

Fig. 4 compares the effectiveness of CPPM-DAM and RFKD. From the simulation data, we can find the defense capability of RFKD is based on the compromise of visiting delay. With the increasing of W , although the defense capability is increased, the average RTT of hit contents also becomes large due to the requesters will face more miss responses. Predictably, when W approaches infinity, RFKD can also achieve perfect protection of cache privacy, but the advantage of caching in NDN will be lost totally at the same time. On the other hand, most of content delivery capability of NDN is kept under CPPM-DAM. According to the design of CPPM-DAM, only the requester is considered as a freshman for specific content by router, it will add an extra response delay. So, this mechanism affects slightly for most legitimate user. From **Fig. 4**, we can see the average RTT of CPPM-DAM is obviously less than RFKD. Another interesting thing is that the average RTT of CPPM-DAM increases a little with the increasing of relative attack rate. The reason is the address collision of Bloom filter will increase when attacker sends more requests.

To clearly disclose the impact of attack rate in timing attack, we draw the **Fig. 5**. In this Figure, we change the relative attack rate from 0.01 to 1.8. Without protection, the attack success probability will approach 1 when attack rate approaches 0. And we also happy to see CPPM-DAM always keeps excellent protection whatever attack rate is high or low.

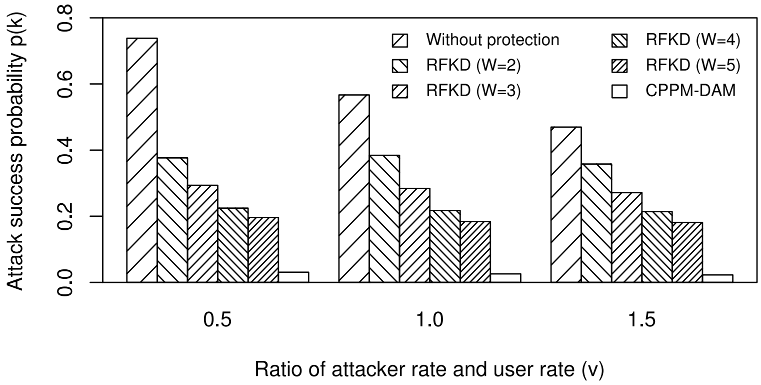


Fig. 3. The comparison of attack success probability

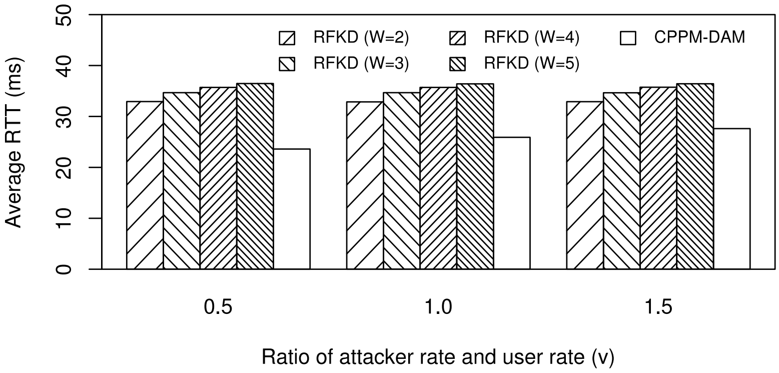


Fig. 4. The comparison of average RTT

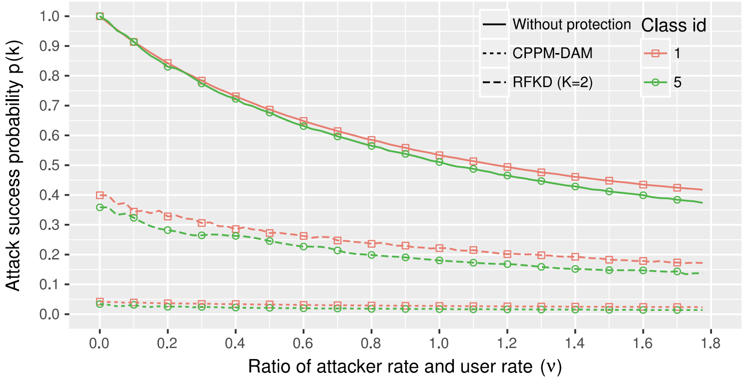


Fig. 5. The impact of relative attack rate

4.2 The number of hash functions and matrix size

Although CPPM-DAM can protect cache privacy powerfully, there still exists the possibility of cache privacy leakage. In fact, there are two available failure cases.

(1) The adversary selects the same permutation indicator of hash functions as legitimate user occasionally.

(2) For different contents in the same class, their mapping addresses maybe occur collision occasionally. This case is also called as False Positives.

The former case depends on the number of hash functions, and the latter depends on the matrix size. But how many hash functions and how large size of matrix can guarantee the expected privacy protection effect? **Fig. 6** gives the curves of attack success probability under the variation of n and $B^k_{\max}$, where we set $\nu = 0.2$ and snooping target is the first class.

From **Fig. 6**, we clearly see that the attack success probabilities significantly decrease along with the increasing of n and $B^k_{\max}$. If the number of hash functions is too small, the adversary can easily guess the permutation choice of legitimate user and then lead to privacy leakage. If the matrix size is too small, the phenomenon of False Positives will become serious and then lead to abnormal work. So, enough hash functions and large matrix size are benefit for cache privacy protection. But this improvement is found at the expense of the storage cost of Bloom filter.

If we require the matrix can hold all available mapping addresses of target class, the storage cost in CPPM-DAM should satisfy the condition of expression (3). In Exps.3, m denotes the number of files in each class and $(B^k_{\max})^n$ is the matrix size for class k, the storage cost will be huge for large n and $B^k_{\max}$.

$$B^k_{\max} \geq m \times n! \tag{3}$$

For example, when we set $n = 5$ and $B^k_{\max} = 20$, the attack success probability can be restrained below 0.01. In the meanwhile, the storage cost of each class is 20^5 bits and the total cost of 50 classes is 10.07 MB. Although this storage cost is just a trifling number, the reason is only 50 classes and 125,000 files considered in simulation. For real world, the contents provided by network should be amazing. If we assume the file amount of each class reach 10^5, the Bloom filter should be configured as $n = 6$ and $B^k_{\max} = 40$ for satisfying expression (3). In this case, the required storage space for single matrix is 40^6 bits and the total storage cost of 50 classes is 16.97 GB, which is a considerable cost.

To make CPPM-DAM more practical, how to optimize the storage cost will be further discussed in next section.

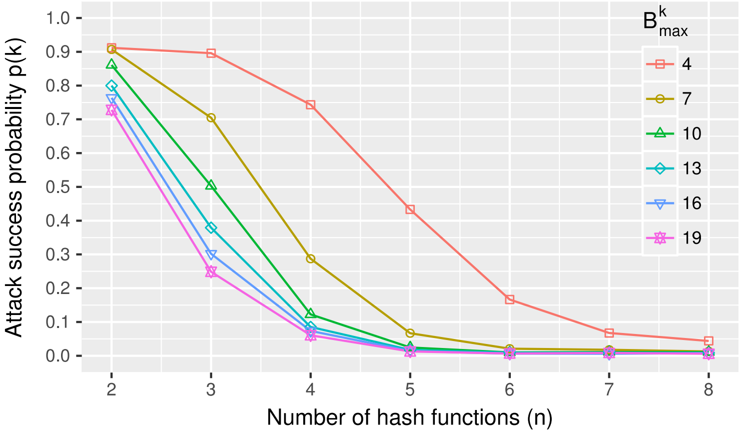


Fig. 6. Attack success probability vs. n and $B^k_{\max}$

5. Storage cost optimization

In the design of CPPM-DAM, we has already configured mapping matrix for different classes respectively, it is an implicit design for restricting matrix size. But it is not enough for storage cost optimization. According to the characteristics of cache privacy and request behavior in NDN, we suggest optimizing storage cost from the following two aspects.

(1) The contents should be divided into privacy contents and the non-privacy contents based on their popularity, and then be deal with separately. For the request of non-privacy content, it should be directly processed by original NDN mechanism. For the request of privacy content, router should process it according to CPPM-DAM. Definition 2 provide a recommended way to distinguish the privacy and non-privacy content.

Definition 2 (Privacy content). Follow the definition of information entropy [28], we define I_k as information entropy of privacy leakage for the content in class k, I_{av} as the average information entropy of privacy leakage for all classes, as shown in expression (4). If I_k is bigger than I_{av}, the content is defined as privacy content, otherwise it is defined as non-privacy content.

$$\begin{cases} I_k = \log_2 \frac{1}{q_k},\ I_{av} = \sum_{k=1}^{K} q_k \times \log_2 \frac{1}{q_k} \\ I_k > I_{av} \Rightarrow \mathrm{Pr}\,ivacy \\ I_k < I_{av} \Rightarrow non-\mathrm{Privacy} \end{cases} \tag{4}$$

Based on above definition, the router can determine the privacy and non-privacy content by the statistics of visiting history.

(2) According to the design of CPPM-DAM, all available mapping addresses in matrix are cleared to “0” when the corresponding content is removed from CS. That means, even if a sufficient large space is allocated for the matrix of privacy class, there will be only few non-zero elements existing in matrix at any time, and the most area of matrix should be empty. Further considering the requests sent by users become sparser with the decreasing of target popularity. So, the lower the popularity of content class is, the higher the empty rate of the corresponding matrix is. In other words, we can reduce the matrix size by assigning smaller $B_{\max}^k$ for lower popularity class.

From the above analysis, we can understand the key of reducing the storage cost of CPPM-DAM. That is to restrict the matrix size of Bloom filter according to its popularity. Next, we will optimize the matrix size of privacy class k on the assumption of LRU replacement policy. Now we introduce a new variable τ which is the characteristic time of router[29], it is used to describe the average residence time of cached content in CS. Within τ, the average number of requests $Num(k)$ from legitimate user for privacy contents in class k can be calculated as equation (5) [29].

$$\begin{cases} Num(k) = \lambda_U q_k \tau \\ \sum_{k=1}^{K} m\left(1 - e^{-\lambda_U q_k \tau / m}\right) = C \end{cases} \tag{5}$$

Using equation (5), we give an example of analyzing the average requests for privacy contents in class k within characteristic time τ by Matlab. The analysis settings of this example are as follow, the provided contents in NDN are divide into $K = 200$ classes, each class contains $m = 10^5$ files, the cache size is $C = 2500$ files, the request rate is $\lambda_U = 10^5$ interest packet/s, and the popularity parameter is $\alpha = 0.8$

Table 3. The average requests of top 10 privacy classes within $\tau = 0.025s$

Parameter of popularity $\alpha = 0.8$										
Class id	20	21	22	23	24	25	26	27	28	29
Average requests	23	22	21	20	20	19	18	18	17	17

According to Definition 2, we classify the contents after 19th class as privacy under $\alpha = 0.8$.Based on equation (5), the characteristic time τ is solved, its value is about 0.025s. Then the average requests for the contents of the top 10 privacy classes within τ can be given, as shown in **Table 3**.

From the results of **Table 3**, although the files of each class reach 10^5, the average requests for the top 10 privacy classes don't exceed 100 under the heavy traffic of 10^5 interest packet/s. That is to say, for class 20, there will be at most $23 * n!$ non-zero elements simultaneously appeared in the matrix within the characteristic time, not $10^5 * n!$. Therefore, how many requests will arrive within the characteristic time for a specific class, mostly depends on its arrival rate and the cache size (small cache size means small characteristic time). It has nothing to do with the amount of contents provided by network. This observation guides us an important direction for optimizing storage cost in CPPM-DAM. So, we can replace the condition of expression (3) by expression (6). For the storage cost of class k, it is enough with satisfying the following condition.

$$(B_{max}^{k})^{n} \geq Num(k) \times n! \tag{6}$$

For the example in this section, before optimization, we set B_{max}^{k} as 40 and n as 6 for all classes, which consumes $16.97\,\text{GB}$ storage cost; after optimization, we can set B_{max}^{k} as 6 and n as 6 for all classes, which totally only consumes only $1\,\text{MB}$ storage cost. And under this setting, the CPPM-DAM can still achieve the attack success probability less than 10^{-3} and realize effective cache privacy protection.

6. Conclusions and Future Recommendations

Focusing on the issue of cache privacy protection in NDN, we propose a protection mechanism named CPPM-DAM to identify the request from adversary and legitimate user. The design of CPPM-DAM is based on Bloom filter. By appending permutation indicator of hash functions into interest packet, it can map content name into $n!$ available addresses of n-dimensional matrix, and then recognize the identity of request. After identification, an extra response delay is added for the new requester and realize cache privacy protection. The simulation results show that the CPPM-DAM can achieve perfect protection to cache privacy by setting suitable matrix size and number of hash functions. By optimizing its storage cost, the feasibility of CPPM-DAM is also effectively improved.

Nowadays, implementing access control with digital signature is an important way to solve most security issues in NDN. It is also an important research direction for cache privacy protection. How to design reasonable signature and feasible verification mechanism to protect cache privacy are the keys in future research.

References

[1] T. Koponen, M. Chawla, B.G. Chun, et al., "A data-oriented(and beyond) network architecture," *ACM SIGCOMM Computer Communication Review*, vol. 37, no.4, pp. 181–192, 2007. Article (CrossRef Link)

[2] S. Tarkoma, M. Ain, K. Visala, "The publish/subscribe internet routing paradigm (psirp): Designing the future internet architecture," *Future Internet Assembly*, pp. 102–111, April, 2009.

[3] L. Zhang, D. Estrin, J. Burke, et al., "Named data networking (ndn) project," in *Proc. of Relat´orio T´ecnico NDN-0001*, Xerox Palo Alto Research Center-PARC ,2010.

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

[5] E. Ngai, B. Ohlman, G. Tsudik, et al., "Can We Make a Cake and Eat it Too? A Discussion of ICN Security and Privacy," *ACM SIGCOMM Computer Communication Review*, vol. 47, no.1, pp. 49-54, 2017. Article (CrossRef Link)

[6] Vasilakos, V. Athanasis, Z. Li, et al., "Information centric network: Research challenges and opportunities," *Journal of Network and Computer Applications*, vol. 52, pp.1-10, 2015. Article (CrossRef Link)

[7] D. Saxena, V. Raychoudhury, N. Suri, et al., "Named data networking: a survey," *Computer Science Review*, vol.19, pp.15-55, 2016. Article (CrossRef Link)

[8] T. Lauinger, "Security & scalability of content-centric networking," *Master's thesis, Technische University*, 2010.

[9] Dogruluk, Ertugrul, A. Costa, and J. Macedo, "Evaluating privacy attacks in Named Data Network," in *Proc. of 2016 IEEE Symposium on Computers and Communication (ISCC)*, pp. 1251-1256, 2016. Article (CrossRef Link)

[10] G. Acs, M. Conti, P. Gasti, et al., "Cache privacy in named-data networking," in *Proc. of 2013 IEEE 33rd International Conference on Distributed Computing Systems (ICDCS)*, pp. 41–51, 2013. Article (CrossRef Link)

[11] A. Mohaisen, X. Zhang, M. Schuchard, et al., "Protecting access privacy of cached contents in information centric networks," in *Proc. of Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pp. 173–178, 2013. Article (CrossRef Link)

[12] A. Mohaisen, H. Mekky, X. Zhang, et al., "Timing attacks on access privacy in information centric networks and countermeasures," *IEEE Transactions on Dependable and Secure Computing*, vol.12, no.6, pp. 675–687, 2015. Article (CrossRef Link)

[13] Da Silva, Roan Simões, Sergio Donizetti Zorzo, "An access control mechanism to ensure privacy in named data networking using attribute-based encryption with immediate revocation of privileges," in *Proc. of the 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 128-133, 2015. Article (CrossRef Link)

[14] M. Ion, J. Zhang, E. M. Schooler, "Toward content-centric privacy in ICN: Attribute-based encryption and routing," in *Proc. of Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, ACM, pp. 39-40, 2013. Article (CrossRef Link)

[15] T. Chen, K. Lei, K. Xu, et al., "An encryption and probability based access control model for named data networking," in *Proc. of 2014 IEEE International Performance Computing and Communications Conference (IPCCC)*, IEEE, pp. 1-8, 2014. Article (CrossRef Link)

[16] T. Lauinger, N. Laoutaris, P. Rodriguez, et al., "Privacy risks in named data networking: what is the cost of performance?," in *Proc. of ACM SIGCOMM Computer Communication Review*, vol.42, no.5, pp. 54–57, 2012. Article (CrossRef Link)

[17] T. Lauinger, N. Laoutaris, P. Rodriguez, et al., "Privacy implications of ubiquitous caching in named data networking architectures," *Technical Report TR-iSecLab-0812-001*, ISecLab, Tech. Rep., 2012.

[18] A. Chaabane, E. De Cristofaro, M. A. Kaafar, et al., "Privacy in content-oriented networking: Threats and countermeasures," in *Proc. of ACM SIGCOMM Computer Communication Review*, vol.43, no.3, pp. 25–33, 2013. Article (CrossRef Link)

[19] S. Arianfar, T. Koponen, B. Raghavan, et al., "On preserving privacy in content-oriented networks." in *Proc. of Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, ACM, pp. 19-24, 2011. Article (CrossRef Link)

[20] Z. Xingwen, H. Li, "Privacy Preserving Data Sharing Scheme in Content Centric Networks against Collusion Name Guessing Attacks," *IEEE Access*, vol.5, pp.23182-23189, 2017. Article (CrossRef Link)

[21] B. H. Bloom, “Space/time trade-offs in hash coding with allowable errors,” *Communications of the ACM* , vol. 13, no. 7, pp. 422-426, 1970. Article (CrossRef Link)

[22] S. Mastorakis, A. Afanasyev, L.Zhang, “On the Evolution of ndnSIM: an Open-Source Simulator for NDN Experimentation,” in *Proc. of ACM SIGCOMM Computer Communication Review*, vol. 47, no.3, pp. 19-33, 2017. Article (CrossRef Link)

[23] S. Mastorakis, A. Afanasyev, I. Moiseenko, et al., “ndnsim 2.0: A new version of the ndn simulator for ns-3,” *NDN*, Technical Report NDN-0028, 2015.

[24] M. Mangili, F Martignon, S Paraboschi, “A cache-aware mechanism to enforce confidentiality, trackability and access policy evolution in content-centric networks,” *Computer Networks*, no.76, pp.126-145, 2015. Article (CrossRef Link)

[25] C. Fricker, P. Robert, J. Roberts, N. Sbihi, “Impact of traffic mix on caching performance in a content-centric network,” in *Proc. of 2012 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, pp. 310–315, 2012. Article (CrossRef Link)

[26] I. Psaras, R. G. Clegg, R. Landa, et al, “ Modeling and Evaluation of CCN-Caching Trees,” *IFIP Networking*, pp.78-91, 2011.

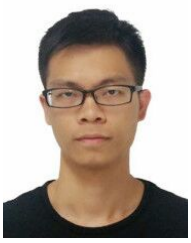
[27] Y. Collet, “xxhash-extremeley fast hash algorithm,” 2016, Article (CrossRef Link).

[28] T. M. Cover, J. A. Thomas, “Elements of information theory,” *John Wiley & Sons*, New York, 2012.

[29] N. Laoutaris, H. Che, I. Stavrakakis, “The lcd interconnection of lru caches and its analysis,” *Performance Evaluation*, vol. 63, pp. 609–634, 2006. Article (CrossRef Link)

Yi Zhu, PhD, associated professor of Jiangsu University. His research interests include information-centric networking, and software defined networking.

Haohao Kang, graduate student of communication and information system in Jiangsu University. His research interest is Named data networking.



Ruhui Huang, graduate student of electronics and communications engineering in Jiangsu University. His research interest is Named data networking.