Cache Protection Method Based on Prefix Hierarchy for Content-oriented Network

Takashi Kamimoto, Kenta Mori, Sayaka Umeda, Yuri Ohata, Hiroshi Shigeno
Graduate School of Science and Technology
Keio University
3-14-1 Hiyoshi Kohoku-ku
Yokohama-shi Kanagawa Japan
{kamimoto, mori, umeda, ohata, shigeno}@mos.ics.keio.ac.jp

***Abstract*—Cache pollution attack is one of the prominent problems for Named Data Networking (NDN). Attackers pollute a cache of NDN by contents which are not popular among normal users. Normal users take more time to obtain contents due to the attack. There are some countermeasures against cache pollution attack in NDN, but most of them focus on full content names. Using full names needs a large amount of storage cost. In this paper, we propose a cache protection method against cache pollution attack based on hierarchy of content name prefixes in Named Data Networking (CPMH). CPMH can defend caches from malicious users and alleviate the influence of attack after detecting it. The purpose of CPMH is maintaining the accessibility of normal users to obtain contents. CPMH alleviates cache hit ratio degradation caused by attack. In CPMH, each router identifies the prefixes of contents requested by attackers based on hierarchy of content name prefixes and protects its own cache. We compare CPMH with the related work by simulation. As a result, we confirm the effectiveness of CPMH against cache pollution attack.**

***Index Terms*—Information-Centric Networking; Named Data Networking; Cache pollution attack; prefix hierarchy.**

I. Introduction

Research of new network architectures which are considered to replace with Internet Protocol (IP) is emerging recently. Named Data Networking (NDN) is one of them [1]. NDN is a content-oriented network in which users assign content names when they want to obtain data. Users can obtain contents without depending on location unlike IP in NDN. Each NDN router has a cache, called Content Store (CS). Utilizing this cache, users can obtain contents faster in contrast with the case that caching function is not implemented in all routers. A NDN data packet contains a signature, so NDN implements the authentication of contents. Moreover, NDN is tolerant of DoS attack for specific users which is prominent problem in IP because there is no identifier of users corresponding to IP addresses in NDN.

Cache pollution attack, however, is one of the prominent problems for NDN. Attackers pollute the cache by contents which are not popular among normal users and makes normal users take more time to obtain contents [2]. Countermeasures against cache pollution attack in NDN are proposed in some research [3], [4], [5]. Most of them use full content names to detect or alleviate the attack. Using full names takes a large amount of storage cost, and routers cannot identify the unpopular contents until attackers request the contents.

In this paper, we propose a cache protection method against cache pollution attack based on hierarchy of content name prefixes in Named Data Networking, called CPMH. This method focuses on content name prefixes and defends caches from pollution after detecting attacks. In CPMH, each router identifies the prefixes of contents requested by attackers based on hierarchy of content name prefixes instead of full content names. Then, the router protects its own cache by using the information of the identification.

There are two purposes of CPMH as below.

- Reducing a storage cost as possible not to become much larger than only detecting attack
- Protecting caches without damaging normal users

To achieve the purposes, CPMH makes use of hierarchy of content name prefixes. The name prefix hierarchy is like a tree structure. Some different contents have a same prefix, so indicating one prefix means indicating some contents. Using content name prefixes, not using full content names, reduces the storage cost. A large aggregation, however, has a risk that a blacklist contains prefixes which are popular among normal users. CPMH uses prefix length which corresponds to depth of tree to avoid this risk. Thus, CPMH enables routers to achieve above two purposes.

The rest of paper is organized follows. In section Ⅱ, we introduce Named Data Networking. In section Ⅲ, we discuss related work against cache pollution attacks in NDN. In section Ⅳ, we describe our method, CPMH, in detail. In section Ⅴ, we show the effectiveness of CPMH by simulation and conclude this paper in section Ⅵ.

II. Overview of Named Data Networking

Named Data Networking is a content-oriented network which is considered to replace of a location-oriented network, IP [1]. In NDN, users obtain contents with assigning content names, thus naming functions are important for NDN [6]. There are two types of NDN packets, Interest packet and Data packet. Interest is a request packet generated by users and forwarded from users to producers via some routers. Otherwise, Data is a response packet sent by producers against Interest and contains content.

978-1-4673-9292-1/16/$31.00 ©2016 IEEE

To forward packets, each NDN router has three elements, FIB, PIT and CS. Forwarding Information Base (FIB) and Pending Interest Table (PIT) are tables which are used to forwarding Interests or Data. Content Store (CS) is a cache for storing contents according to the router's caching policy. Caching policy is a rule which contents the router stores. If there are Interests which request the same content stored in CS, the router can send Data to users. In this way, NDN helps users to obtain contents more efficiency.

Data which can be retrieved globally have globally unique names, but local Data have local names which are only used in local communications. A content name is divided into several components, and each component is bound with '/'. For example, "/Photos/ePid45Fs/data1" is a content name, and "/Photos", "/ePid45Fs" and "/data1" are components. One part of content names which starts from the head of name is called prefix. The shortest prefix is composed of only the first component. Prefixes of NDN have a hierarchy like a tree structure. The closer components are located by the head of content name, the more contents contain them. In other words, prefixes aggregate some content names. Each producer announces name prefixes they have, and each router constructs FIB by using the announcement. For example, a producer which has "/Photos/ePid45Fs/data1" and "/Photos/ePid45Fs/data2" announces "/Photos/ePid45Fs/" instead of each content names.

III. Related work

In this section, we introduce cache pollution attack and countermeasures about the attack.

A. Cache pollution attack

Cache pollution attack is that attackers pollute caches and make normal users take more time to obtain contents. It is said that there are two classes in cache pollution attack [2]. One is Locality-disruption attack, and the other is False-locality attack. A locality of cache is the property of reflecting the popularity of nodes located near the router. In Locality-disruption attack, malicious users request a lot of different contents which are not popular among normal users. This attack pollutes caches with contents which have low request rate. False-locality attack is done by requesting few unpopular contents. Each of unpopular contents has high request rate, so routers cannot distinguish between popular contents and unpopular contents. This indecision causes a problem that routers store unpopular contents requested by malicious users. In this way, NDN router has false locality.

The influence of cache pollution attack depends on cache replacement algorithm of routers. Cache replacement algorithm is a rule that decides which content is erased when a cache becomes full [7]. The major algorithms are Least Recently Used (LRU) and Least Frequently Used (LFU). In LRU algorithm, content requested least recently is preferentially erased. This algorithm is ineffective in both Locality-disruption and False-locality [5]. LFU algorithm is the algorithm in which least frequently requested content is preferentially erased. This algorithm is effective in Locality-disruption attack because request frequency of contents requested by attackers is low, but ineffective in False-locality attack.

B. Countermeasures in NDN

CacheShield [3] is one of the countermeasures against cache pollution attack in NDN. This method focuses on especially Locality-disruption attack. In CacheShield, each NDN router keeps track of a requested count for each of contents. Routers do not store contents until the counts of contents become beyond a threshold. This is valid for Locality-disruption attack, but not for False-locality attack. Moreover, each router has a large amount of contents data.

There is a method which focuses on the randomness of Interests against cache pollution attack in NDN [4]. This method is based on the idea that requests of normal users have randomness, and checks it. If the randomness is less than a threshold, the router decides that there is an attack. This detection method can only detect attack and apply for Locality-disruption attack. Randomness calculations of Interests need high computational complexity.

There is a method which is considered after above two work (we call this method Lightweight attack detection in this paper) [5]. In this work, each router checks the request rate of each of selected contents. Request rate $p(i)$ of content i is calculated as:

$$p(i) = \frac{n_r(i)}{\sum_{j \in S} n_r(j)}, \tag{1}$$

where S means a set of selected monitoring contents, $n_r(i)$ means the number of requested count of content i and $\sum_{j \in S} n_r(j)$ means the total requested count of whole selected contents. Then, routers compare the newest rate with the past rate, and calculate the variation of request rate δ_i as:

$$\delta_i = (\frac{n_r^m(i)}{N_r^m} - p(i)), \tag{2}$$

where m means the past measurement, and N_r^m means the requested count of whole selected contents at the past. The variations of monitoring contents are summed up and compared with threshold. If the total variation is beyond threshold, router decides that there is an attack. This method can apply for both Locality-disruption attack and False-locality attack. This can only detect attack, but cannot protect cache.

IV. CPMH

We propose a cache protection method against cache pollution attack based on hierarchy of content name prefixes in Named Data Networking, called CPMH. In this section, we describe about our method, CPMH.

A. Overview

CPMH can defend caches from malicious users after attack detection. This method uses the hierarchy of content name prefixes to identify the unpopular contents requested by attackers from all requested contents. CPMH is a countermeasure against cache pollution attack, especially False-locality attack.

Attackers request fewer contents in False-locality attack than Locality-disruption attack. Identifying prefixes of unpopular contents is easier against False-locality attack.

CPMH has two purposes. One is reducing a storage cost as possible not to become much larger than only detecting attack. To detect cache pollution attack, we have the use of ideas of Lightweight attack detection because it implements low calculation and low storage cost. On the other hand, this method only focuses on detection, and the amount of information needed to protect caches is lacked due to select monitoring contents. We use the idea of this method without selecting contents and implement low storage cost using content name prefixes. Second purpose is protecting caches without damaging normal users. It is not desirable that CPMH inhibits normal users from obtaining contents.

B. Three steps of CPMH

CPMH is composed of three steps as below.

- **Identifying the attackers' prefixes.** Identifying the prefixes of contents requested by attackers and storing them in blacklist
- **Cache Recovery.** Recovering a cache with removing contents which have prefix contained in the blacklist
- **Cache Protection.** Protecting a cache with avoiding storing contents which have prefix contained in the blacklist

When routers detect attacks, they identify the prefixes of contents requested by attackers. After that, if a prefix of arrival content is included in the blacklist, router does not store the content to protect CS. As a result, routers can maintain cache hit ratio, and normal users can utilize the cache storage. The rest of this section, we describe each step of CPMH in detail.

C. Identifying the attackers' prefixes

To protect caches, routers identify prefixes which are requested by attackers and not popular among normal users. CPMH is based on content name prefixes though Lightweight attack detection focuses on selected content names [5]. There are two reasons of using prefixes. First reason is that we need to reduce the storage cost. We can reduce storage cost by using prefixes because a prefix of name tree contains his child contents. Second reason is that we can extend the range of identifying contents requested by attackers. If identifying contents based on content full names, each router cannot decide whether the arrival content is malicious or not until a router find the content used by attackers at first time. Otherwise, routers can decide the attackers' contents which have prefixes contained in a blacklist are malicious by using prefixes. Thus, CPMH makes use of content name prefixes.

To identify the prefixes requested by attackers, CPMH has three steps. First, routers calculate Request rate Variation per Prefix (RVP), but using only this value causes a problem that routers mistake popular prefixes for attackers' prefixes. Thus, routers calculate Weighted RVP (WRVP) in the next step. Finally, they make a blacklist by using the WRVP.

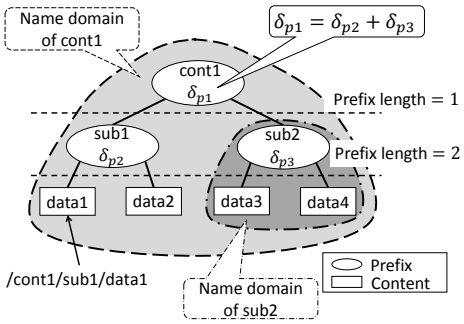

Fig. 1. Prefix hierarchy and RVP

1) Request rate Variation per Prefix: For the effective protection, CPMH needs to identify the attackers' prefixes without containing prefixes which are popular among normal users. CPMH implements this by using RVP and prefix length on the basis of the name hierarchy.

RVP δ_p is calculated as:

$$\delta_p = \sum_{i \in S_p} \delta_i, \qquad (0 \leq \delta_p \leq 1) \tag{3}$$

where the variation of contents δ_i is calculated by Equation 2 when routers detect attack, and S_p means a set of contents which have same prefix p. δ_i is not normalized, but δ_p is normalized by the total amount of δ_i. According to this equation, prefixes near the root of tree have larger variation. They contain more a lot of contents. If routers use RVP to identify the prefixes of unpopular contents, there is a probability that they fail to detect popular prefixes as unpopular prefixes. This is a problem because one of the purposes of CPMH is protecting caches without damaging normal users.

Figure 1 shows an example of the relation between a name prefix hierarchy and RVP. Prefixes have hierarchy like a tree structure. Here, we define that the RVPs of '/cont1', '/sub1' and '/sub2' are δ_{p1}, δ_{p2} and δ_{p3} respectively. The RVP of '/cont1' is calculated by adding the RVP of '/sub2' to the RVP of '/sub1'. Hence, δ_{p1} is bigger than δ_{p2} or δ_{p3}, which means that the prefix located near the root of hierarchy has a larger RVP. There is a risk of damaging normal users because the prefixes near the root tend to contain a lot of contents, especially popular contents.

2) Weighted RVP: To solve the problem which is caused by using only RVP, CPMH uses length of prefix. Routers need to identify as far prefix from root as they can, but do not have to leave attackers' prefixes as normal prefixes. Prefix length corresponds to the depth of tree. Length is equal to the number of components which constitute prefixes. For example, the length of "/cont1/sub1/subsub3/" is three. We reflect the depth in RVP to limit the range of prefix identification. Here, we introduce the value which is the variation reflected depth, WRVP. WRVP δ_{wp} is calculated as:

$$\delta_{wp} = \delta_p \times w_l, \qquad (0 \leq \delta_{wp} \leq 1,\ 0 \leq w_l \leq 1) \tag{4}$$

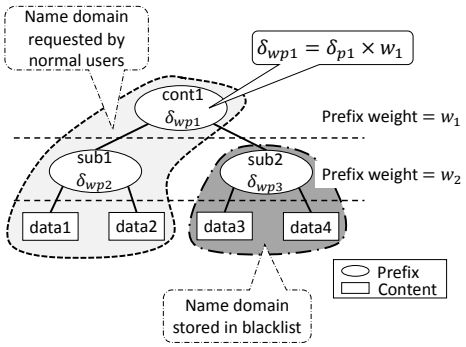

Fig. 2. Prefix hierarchy and WRVP

where l means a length of prefix, w_l is the weight which is for reflecting length in variation. The longer prefixes are, the larger w_l become, but maximum value is 1. In this way, WRVP reflects both the probability that is used by attackers and the depth of tree structure.

Figure 2 shows the example of the relation between a name prefix hierarchy and WRVP. The weight multiplied to RVP is allocated to prefix length. The weight of prefix whose length is one is w_1, and whose length is two is w_2. w_2 is larger than w_1. δ_{p1} is larger than δ_{p2} because of name hierarchy, but the magnitude correlation between δ_{wp1} and δ_{wp2} does not depend on the hierarchy. They rely on the probability whether malicious users intentionally request them or not. Using WRVP, routers can correctly decide that '/cont1/sub2' is used by malicious users while they incorrectly decide that '/cont1' is used when they use only RVP. Normal users who request popular contents obtain contents with utilizing caches.

3) Making a blacklist: After calculating WRVP, CPMH selects prefix candidates which have high WRVP. Selected candidates have high probability that they requested by malicious users. Blacklist threshold τ $(0 \leq \tau \leq 1)$ is calculated by using WRVP of candidates. Prefixes which have higher WRVP than threshold τ are registered in a blacklist. τ is set to the value to register at least one prefix in a blacklist, because there is an attack when this procedure is executed.

We adopt the following way to calculate τ. τ is dynamically computed based on WRVP. A router selects a prefix which has the largest WRVP from some longest prefixes, which is determined based on requested contents. The threshold is set to be equal to WRVP of the selected prefix. In this way, at least a selected prefix is blacklisted whenever an attack is detected.

D. Cache Recovery

Cache recovery means that caches of routers attacked by malicious users recover from polluted state. There is the necessity of recovering cache from cache pollution attack by using a blacklist. When attack is detected by routers, they are polluted by unpopular contents. After making a blacklist, each router checks contents stored in its own cache with a blacklist. If there are contents which prefixes are contained in the blacklist, routers erase the contents from cache. Depending on cache replacement to erase unpopular contents, routers take more time to recover their own cache. Using cache recovery, the free space of cache storage come from cache recovery is utilized to store the popular contents.

TABLE I
SIMULATION PARAMETERS

Network simulator	ns-3.20
NDN module	ndnSIM 1.0
Simulation time	7200 sec
Topologies	XC, DFN
Cache replacement algorithm	LRU
Capacity of router's CS	1% of whole contents
Contents requested by attackers	Equal to CS capacity
Interest of attackers	Equal to total of normal interests [/sec]

E. Cache Protection

After cache recovery, CPMH starts cache protection function against cache pollution attack. There are two methods to protect caches. One method is to avoid storing unpopular contents requested by attackers, and the other is discarding malicious Interests which request the unpopular contents. CPMH adopts first method. Unpopular contents are requested less frequently by normal users, but there is the probability that normal users request them. To protect cache, storing truly popular contents satisfies the purpose of CPMH. In CPMH, if a prefix of arrival content is included in the blacklist, the router does not store the content. Routers defend caches by avoid storing the contents requested by attackers and can maintain cache hit ratio. Thus, normal users can utilize CS.

V. EVALUATIONS

To evaluate CPMH, we show the results of simulations. This section explains the setup of simulation and the result.

A. Simulation setup

We confirm the effectiveness of CPMH against cache pollution attack by simulation. We use two topologies and evaluate two items. A simulator we used is ns-3 [8] and NDN module is implemented by ndnSIM [9]. In the simulations, normal users request contents followed by Zipf-distribution [10]. Attackers do cache pollution attack, especially False-locality attack, by requesting few unpopular contents followed by uniform-distribution. The simulation parameters are shown in Table I. We compare CPMH with Lightweight attack detection, so most of parameters used in simulation follow those in [5]. Simulation time is two hours. In the former half of simulation, contents are requested only by normal users. In the latter half, both normal users and attackers request contents.

We use two topologies, XC [3] and DFN [11]. Figure 3 shows XC, and Fig. 4 shows DFN. XC has a characteristic that there are some nodes which have same condition except for the distance from attackers. This topology is suitable for comparing between routers which locate the different distance from attackers. DFN is a more complex and realistic topology than XC. DFN is suitable for confirming the effect of method in realistic environment.

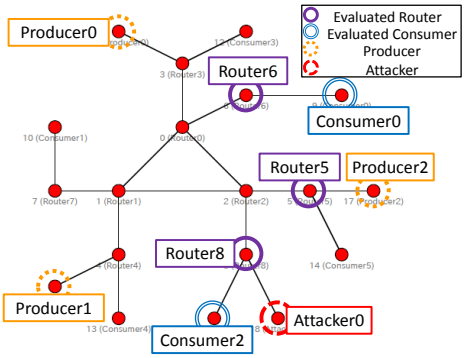

Fig. 3. XC topology

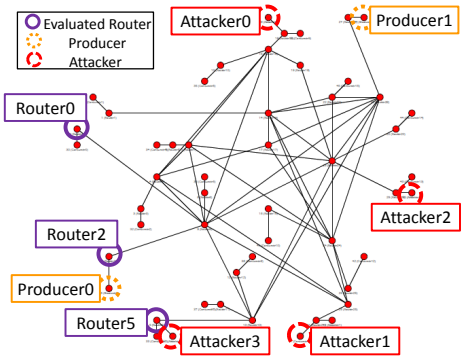

Fig. 4. DFN topology

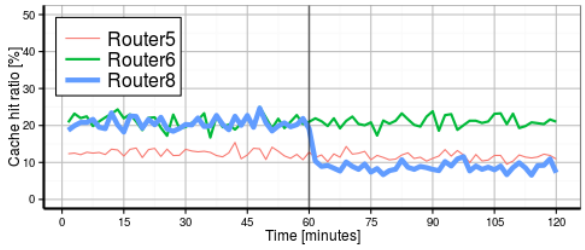

Fig. 5. Hit ratio of Lightweight attack detection on XC topology

B. Evaluation items

There are two evaluation items to confirm the effectiveness of CPMH. First item is cache hit ratio of normal users at each router. Cache pollution attack degrades the cache hit ratio of normal users because the domain of cache polluted by attackers cannot be utilized for retrieving popular contents. With monitoring this value, we can confirm the effectiveness of CPMH against cache pollution attack.

Second item is hop count which is the sum of Interest hops and Data hops to obtain contents. Degradation of cache hit ratio causes increment of hop counts, and normal users take longer time to get contents. We use the hop count to see if CPMH affects the time of retrieving contents.

C. Cache hit ratio

In this simulation, we evaluate all routers on both of two topologies, and we classify routers into three categories based

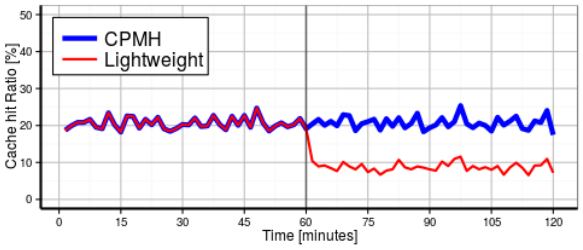

Fig. 6. Comparison of Hit ratio on XC topology

on characteristics of results: upstream routers, downstream routers next to no attackers, and downstream routers next to attackers. We describe typical routers from three classes due to lack of space. Selected routers in XC and DFN are shown as Fig. 3 and Fig. 4. The position of attackers and normal users (which are indicated as consumers) is also shown in those figures. Cache hit ratio equal to 100% means that all of Interests arrived at a router are satisfied in the router.

Figure 5 shows the hit ratio of Lightweight attack detection on XC topology. A horizontal line indicates time course and a vertical line shows cache hit ratio. Router 8 locates next to an attacker, Router 6 is next to a normal user, and Router 5 is an upstream router. When Attacker 0 starts attack at 60 minutes, cache hit ratio of Router 8 falls down quickly. This means that Router 8 is strongly polluted by cache pollution attack. The cache hit ratios of Router 5 and Router 6 vary slightly between before and after attack because they are not located near the attacker. Thus, there is a necessity that CPMH protects Router 8 from cache pollution attack.

We focus on cache hit ratio of Router 8 on XC and compare the result of CPMH with that of Lightweight attack detection (which is indicated as Lightweight) as shown in Fig. 6. According to the figure, hit ratio of CPMH is not degraded by attack. To evaluate quantitatively, we calculate the average hit ratios of before and after attack. When Lightweight attack detection is implemented in Router 8, cache hit ratio of before attack is 20.5%, and that of after attack is 8.8%. The variation is below -10%, thus the hit ratio drastically drops. On the other hand, cache hit ratio of CPMH varies from 20.5% to 20.7%. The variation with CPMH is 0.2%, therefore the cache hit ratio of CPMH in this simulation do not degrade regardless of cache pollution attack.

This result shows that CPMH can protect caches of routers from cache pollution attack. CPMH identifies the prefixes of contents requested by attackers and makes a blacklist, then cache recovery and cache protection functions are executed. Thus, CPMH can alleviate the influence of cache pollution attack at the router which is located near attackers on XC.

We use DFN topology to examine the effectiveness of CPMH on more complex topology. We also selected three routers, but we describe only a downstream router next to an attacker, Router 5, because the influence of attack is strongly appeared. Figure 7 shows the hit ratio of Router 5 when we use Lightweight attack detection or CPMH on DFN. The cache

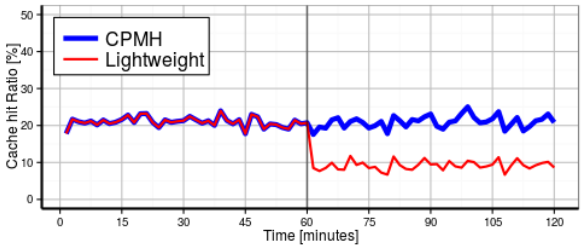

Fig. 7. Comparison of Hit ratio on DFN topology

TABLE II
AVERAGE HOP COUNT OF LIGHTWEIGHT ATTACK DETECTION

Monitoring user	Before attack	After attack	Variation
User 0	6.59	6.61	0.02
User 2	6.63	7.18	0.55

TABLE III
AVERAGE HOP COUNT OF CPMH

Monitoring user	Before attack	After attack	Variation
User 0	6.59	6.61	0.02
User 2	6.63	6.64	0.01

hit ratio of the router degrades drastically like Router 8 of XC when routers implement Lightweight attack detection. In contrast, CPMH alleviates the influence of attack on the router located near the attacker. We evaluate the result of using DFN quantitatively too. The hit ratio of Lightweight attack detection is 21.0% before attack starting and 9.2% after attack. On the other hand, that of CPMH is 21.0% before attack starting and is also 21.0% after attack. Thus, CPMH is effective in cache pollution attack in more realistic topology.

D. Hop count

To examine whether CPMH has an effect on the delay of retrieving contents, we monitor the hop counts required by normal users to obtain contents. The reason why we use hop counts, do not use retrieving time of contents, is that retrieving time in our simulation is too small to decide the effect of CPMH. We cannot distinguish between influence of CPMH and acceptable error. We selected two downstream users, Users 0 and 2, on XC which are shown in Fig. 3 as consumers. User 2 is located near the attacker. Maximum hop count of them is 8. It is difficult to select routers which have a same maximum hop count in DFN, so we use only XC topology in this simulation.

Table II shows the hop count of Lightweight attack detection. The hop count of User 2 increases drastically even though that of User 0 increases little. According to the result of cache hit ratio, the hit ratio of Router 8 which is next to User 2 degrades drastically when there is an attack. Thus, cache pollution attack damages delay of retrieving contents of users located near the attackers strongly.

Table III shows the hop count of using CPMH. Using CPMH, the variation of User 2 is 0.01. This means that CPMH alleviates the influence of attack at the normal user located near the malicious user. According to the result of cache hit ratio using CPMH, routers can maintain the cache hit ratio of normal users though there is an attack. Thus, normal users utilize caches efficiently because they can obtain contents not only from producers but also from routers which have corresponding contents to Interests generated by normal users. CPMH can defend nodes from cache pollution attack in terms of delay of retrieving contents.

VI. CONCLUSION

There are some works which propose countermeasures against cache pollution attack in NDN, but most of them focus on full content names. In this paper, we proposed a countermeasure against cache pollution attack based on hierarchy of content name prefixes, called CPMH. In CPMH, each router makes a blacklist of prefixes by using prefix hierarchy and protects a cache from attack.

Simulation evaluations show that CPMH could alleviate the influence of cache pollution attack at nodes which were strongly affected by attack. Protecting caches from attack, CPMH alleviates the degradation of cache hit ratio over 10% in both XC and DFN topology. Moreover, CPMH alleviates the increment of hop counts in XC topology. Thus, CPMH can defend caches of routers, and normal users can obtain contents even though there is attack. We confirmed the effectiveness of CPMH against cache pollution attack.

REFERENCES

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 44, no. 3, pp. 66–73, Jul 2014.

[2] L. Deng, Y. Gao, Y. Chen, and A. Kuzmanovic, "Pollution attacks and defenses for internet caching systems," *Comput. Netw.*, vol. 52, no. 5, pp. 935–956, apr 2008.

[3] M. Xie, I. Widjaja, and H. Wang, "Enhancing cache robustness for content-centric networking," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 2426–2434.

[4] H. Park, I. Widjaja, and H. Lee, "Detection of cache pollution attacks using randomness checks," in *IEEE International Conference on Communications (ICC)*, June 2012, pp. 1096–1100.

[5] M. Conti, P. Gasti, and M. Teoli, "A lightweight mechanism for detection of cache pollution attacks in named data networking," *Computer Networks*, vol. 57, no. 16, pp. 3178–3191, 2013.

[6] C. Tschudin and M. Sifalakis, "Named functions and cached computations," in *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*, Jan 2014, pp. 851–857.

[7] S. Tarnoi, K. Suksomboon, W. Kumwilaisak, and Y. Ji, "Performance of probabilistic caching and cache replacement policies for content-centric networks," in *IEEE 39th Conference on Local Computer Networks (LCN)*, Sept 2014, pp. 99–106.

[8] "ns3," http://www.nsnam.org/, Retr. January 2015.

[9] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Tech. Rep., October 2012.

[10] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: evidence and implications," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, Mar 1999, pp. 126–134 vol.1.

[11] O. Heckmann, M. Piringer, J. Schmitt, and R. Steinmetz, "On realistic network topologies for simulation," in *Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research*, ser. MoMeTools '03, 2003, pp. 28–32.