A Study on Privacy Protection in ICN Networks using Multiple Encryption Keys

Toru HASEGAWA[†], ***Fellow member***, **Shota YAMADA**[†], ***nonmember***, **Yuki KOIZUMI**[†], ***Member***,

SUMMARY Information Centric Networking (ICN) is a promising Internet architecture wherein destinations are specified by data names. Despite usefulness of such name-based communications, there is a risk such that privacy of consumers is leaked from data names. Even if data names are encrypted, attackers reveal data names using frequency attacks under the condition that popularities of data are public. The vulnerability comes from the fact that the single encryption key is used for encrypting all the data names. The paper addresses the vulnerability under the condition that multiple encryption keys are used and considers a mitigation method against frequency attacks.

key words: *ICN, Frequency Attack, Privacy*

1. Introduction

Information Centric Networking (ICN) [1] provides name-based communications to support multicasting, caching, mobility and so on at the network level. Despite the advantages, name-based communications incur a risk that privacy of consumers is leaked from names of requested data [2,3]. That is, a human-readable name of Named Data Networking (NDN) [4], an ICN protocol, leaks much information of a consumer, i.e., a user.

Obfuscating/encrypting data names using encryption is a candidate solution to prevent attackers from knowing data names in plain text. However, Ghali et al. reveal that such encryption is not resilient against frequency attacks if attackers know popularities of the original data and if the single encryption key is used[5]. They show vulnerability in the case that a single data piece consists of a single data packet, and Enatsu et al. extend their work to the case that a single data consists of multiple data packets [6].

The vulnerability comes from the fact that the same encryption key is used for encrypting all the packet names of the same data. This scheme makes a frequency distribution of intercepted encrypted packet names and a request frequency distribution of data names similar, whereas this enables it for request packets to hit data packets in caches. Due to the similarity, an attacker of a frequency attack maps encrypted packet names to a data name in plain text.

In this paper, we design a mitigation method against frequency attacks by using multiple encryption keys. A provider uses different encryption keys for individual consumers so that the frequency distributions of encrypted data packets differ from that of the plain text ones. In this paper, we design the mitigation method and preliminarily evaluate it with simulation.

The paper is organized as follows: Section 2 defines the system model and the frequency attack. Section 3 proposes the mitigation method and Section 4 evaluates it with simulation. Section 5 summarizes the related work and Section 6 concludes the paper.

†The authors are with Osaka University, Suita-shi, Osaka, 565-0871 Japan.

2. System Model and Frequency Attack

2. 1 NDN and Name Encryption

In NDN, a data piece consists of multiple data packets. A consumer requests the data packets by sending request packets with names and the producer sends back the dada packets. We call a name of request/data packet and that of a data piece as a packet name and a data name, respectively.

A producer encrypts packet names and payloads. We call a sequence of such (encrypted) packets and that of their (encrypted) names as a (an encrypted) packet sequence and a (an encrypted) packet name sequence, respectively. Please note that the producer uses different (multiple) encryption keys to encrypt the packet sequence of the same data piece when different consumers request it.

2. 2 System Model

The producer provides N data pieces in set $\mathcal{D}$ and each data piece d_i consists of c_i packets. When a consumer requests data piece d_i, the producer chooses encryption key $k_{i,j}$ among key set $\mathcal{K}_i$. Then it encrypts packet sequence $\mathcal{P}_i$ using the encryption key to encrypted packet sequence $C_{i,j}$ and sends the packets in sequence at constant interval t.

The terminology is defined as follows:

$\mathbb{N}$: The set of natural numbers.

$\mathbb{R}^+$:The set of real numbers greater than 0.

N:The number of data names.

$\mathcal{D} = \{d_1, \ldots, d_N\}$:The set of data names. d_i is the i-th data piece. The data name is ordered in descending order of

popularity. Hereafter, we use interchangeably words: data name and data piece.
$s_i \in \mathbb{N}$:The number of packets of data piece d_i.
$\mathcal{P}_i = \{p_{i,1}, \dots, d_{i,}s_i\}$:The packets sequence of data piece d_i.
$\Lambda_i \in \mathbb{R}^+$:The expected rate of requests for data piece d_i (the request number per second).
$e_i \in \mathbb{N}$:The number of encryption keys for encrypting packet sequence $\mathcal{P}_i$.
$\mathcal{K}_i = \{k_{i,1}, \dots, k_{i,e_i}\}$:The set of encryption keys for packet sequence $\mathcal{P}_i$.
$r_{i,j}(\in \mathbb{N})$:The ratio that j-th key $k_{i,j}$ is used for encrypting packet sequence $\mathcal{P}_i$. Here, $\sum_{j=1}^{e_1} r_{i,j} = 1$.
$C_{i,j} = \{c_{i,j,1}, \dots, c_{i,j,s_i}\}$:The packet sequence to which packet sequence $\mathcal{P}_i$ is encrypted using encryption key $k_{i,j}$.
t:The interval at which data packets are sent by the producer.
T:The duration during which the attacker intercepts packets.

2. 2 Attack Model

The attacker knows all the data names in set $\mathcal{D}$, each packet number s_i of each data piece d_i and the fact that packets of the same request are encrypted with different encryption keys. She/he also knows popularities of data pieces. The popularity is the probability that a data piece is requested by consumers. However, she/he does not know the ratios that individual encryption keys are chosen. In the evaluations, it is assumed that data piece d_i is requested according to the exponential distribution with average $\frac{1}{\Delta_i}$.

The attacker is assumed global and semi-honest. It intercepts encrypted packets at all the links for duration T and infers the data name of in plaint text from the encrypted packet name sequence of the data piece.

2. 3 Frequency Attack

Frequency attacks leverage public information about data name popularities[5,6]. The idea behind them is that the frequency distribution of encrypted data names is similar to that of data name popularities. Assuming the single encryption key, a basic frequency attack lists encrypted data names in descending order of intercepting frequency and maps them to data names in descending order of popularity[5,6]. For example, the most frequently intercepted encrypted name is inferred as the most popular data name.

We extend the simple frequency attack to the case that data pieces consist of multiple data packets and that the same data may be encrypted by different encryption keys. The attacker infers a data name of an encrypted packet name sequence in the two steps. First, it extracts an encrypted packet sequence of the same request. This is achieved by intercept consecutive packets at access links to consumers.

Second, the attacker infers a data dame of a set of encrypted packet sequences encrypted from the same data piece. It divides the encrypted packet sequences to sets of which packet numbers are the same. Here, $\mathcal{EC}_s$ is the set of the encrypted sequences of which packet number is s. The attacker infers data names independently for individual sets. Hereafter, the frequency attack for $\mathcal{EC}_s$ is explained. d_i is i-th data piece in the set and the following procedure is iterated from the highest d_i to the lowest.
(1)The attacker calculates the expected intercepted frequency of encrypted packets of data piece d_i as $T * \Delta_i$.
(2)Since the attacker does not know the ratios of used encryption keys $r_{i,j}$, it extracts all the combinations of encrypted packet sequences in $\mathcal{EC}_{s_i}$ and calculates their intercepted frequencies. Then it chooses the combination of which aggregated intercepted frequency is the nearest to $T * \Delta_i$ as the data name of d_i. Then it removes the encrypted packet sequences in the combination from $\mathcal{EC}_{s_i}$ and goes back to (1) to infer the encrypted packet sequences of the next popular data piece d_{i+1}.

3. Mitigation Method using Multiple Encryption Names

3. 1 Overview

The vulnerability against frequency attacks comes from the fact that frequencies of encrypted packet sequences are different for data pieces. If all the frequencies were equal, the attacker could not differentiate them. However, this requires that a different encryption key is used at each request of the same data piece and obviously it makes impossible for requests to hit data packets in caches. To balance the two extreme cases, this paper designs an anonymity set generation algorithm. It generates anonymity sets in which encrypted packet sequences have the same frequency. Please note that original packet sequences are divided to groups encrypted by different encryption keys.

3. 2 $(k, \varepsilon) -$ Anonymity

The property of anonymity set is defined as $(k, \varepsilon) -$ anonymitiy. The following terms are used to define it and the anonymity set generation algorithm.
$\mathbb{C} = \{C_{i,j} | 1 \le i \le N, 1 \le j \le s_i\}$:The set of encrypted packet sequences.
$\Gamma_s = \{C_{i,j} | |C_{i,j}| = s\} (\subset \mathbb{C})$:The set of encrypted packet sequences of which packet numbers are s.
$\lambda_{i,j}(\in \mathbb{N})$:The rate of encrypted packet sequence $C_{i,j}$. $\lambda_{i,j}$ is equal to $\Delta_i * r_{i,j}$. That is, it is determined by the expected rate of d_i and the ratio of the encryption's being used $r_{i,j}$.
$k(\in \mathbb{N}^+)$:The natural number constant.
$\varepsilon(\in \mathbb{R}^+)$:The real number constant.

The anonymity set generation algorithm generates $\mathbb{C}$ with satisfying $(k, \varepsilon) -$ anonymitiy.

$\mathbb{C}$ is defined to satisfy $(k, \varepsilon) -$ anonymitiy when the following condition is satisfied:

All $\Gamma_s \subset \mathbb{C}$ satisfies $(k, \varepsilon) -$ anonymitiy.

Here, $(k, \varepsilon) -$ anonymitiy is defined as follows:

For arbitrary $C_{w,x} \in \Gamma_s$, at least *k-1* set of encrypted sequence $C_{y,z} \in \Gamma_s$ exists:

(1)$|\lambda_{w,x} - \lambda_{w,z}| \leq \varepsilon$

(2)$w \neq y$

This means that at least *k-1* encrypted packet sequences for every encrypted packet sequence of interest in the set satisfy the following condition: The rate differences are less than equal to ε. The attacker cannot differentiate such k encrypted packet sequences.

3. 3 Anonymity Set Generation Algorithm

The subsection describes a greedy anonymity set generation algorithm. It works for data pieces of which packet numbers are s. Hereafter, data piece d_i consists of s packets. The input is the set of expected packet rates $\Lambda_i \times s$ and the output is the set of encrypted packet rates each of which belongs to one anonymity set. Here, the encrypted packet rate is the rate of packets of d_i encrypted by encryption key $k_{i,j}$. The algorithm uses the following variables:

$\mathcal{R}_s = \{rs_1, \ldots, rs_{|\mathcal{R}_s|}\}$: The list of the rates of encrypted packet sequences sorted in descending order. The list initially holds the input of the algorithm.

$|\mathcal{R}_s|$:The number of rates in $\mathcal{R}_s$.

A basic idea behind the algorithm is dividing the largest rate to the two rates which are equal to some other rates. The iteration eventually generates sets of k equal rates. The algorithm is repetition of such division and it uses variable $index$ for controlling repetition and G which holds rates temporally. Their initial values are 0 an empty and G finally holds the output. The algorithm is described below:

(1) Termination

- If $index = |\mathcal{R}_s| + 1$, then the algorithm terminates. That is, there is no rate which should be divided.

-If $index \leq |\mathcal{R}_s| + 1 - \mathrm{k}$, then it goes to (2). That is, more than k rates remain to be divided.

-Otherwise ($|\mathcal{R}_s| + 2 - \mathrm{k} \leq index \leq |\mathcal{R}_s|$) it goes to (3). That is, the number of remaining rates is less than k

(2)Sort of rates in $\mathcal{R}_s$

It sorts rates of which order is larger than index in $\mathcal{R}_s$ and then extracts rates $rs_j (j \geq index)$ satisfying the condition: $\mathrm{rs_j} > rs_{index+k-1} + \varepsilon$.

-For each rs_j, it divides rs_j to the two rates: $rs_j - rs_{index+k-1}$ and $\mathrm{rs_{index+k-1}}$. It replaces original rs_j with $rs_{index+k-1}$ and adds $rs_j - rs_{index+k-1}$ to $\mathcal{R}_s$. It then goes to (1).

-Otherwise there are more than $k-1$ rates of which differences are less than ε. It adds the rates from $\mathrm{rs_{index}}$ to the minimum rate satisfying the above condition to set G.

(3)Re-division

Since k equal rates are not generated from the remaining rates of which orders are larger than rs_{index}. It re-divides some of rates in set G. It extracts the minimum rate g_{min} in set G and divides g_{min} to $g_{min} - rs_{index}$ and rs_{index}. It replaces rs_{index} with g_{min} and adds $g_{min} - rs_{index}$ to $\mathcal{R}_s$ and then goes to (1).

The producer uses the rates in set G as the ratios of encrypting by individual keys $r_{i,j}$.

3. 4 Evaluation

The anonymity set generation algorithm is applied to the following data. The rates of date pieces are generated with $N = 30$, $\Lambda_i = \frac{1}{7}$ and $s_i = 1, \ldots, 5$ as illustrated in Fig. 1. The number of packets of $data_i$ is randomly chosen from 1 to 5. Figure 1shows the top 20 rates. The X axis shows the data pieces arranged in descending order of rate and the y axis shows the rates. Figure 2 shows the 10 generated rates in the anonymity sets when (k, ε) is $(3, 0.01)$. In The algorithm successfully generates the anonymity sets. For example, the top rate in Fig.1 is divided to 17 rates in Fig.2.

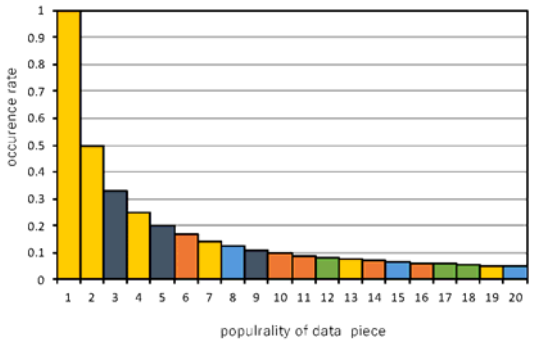

Fig. 1 Rates of Data Pieces

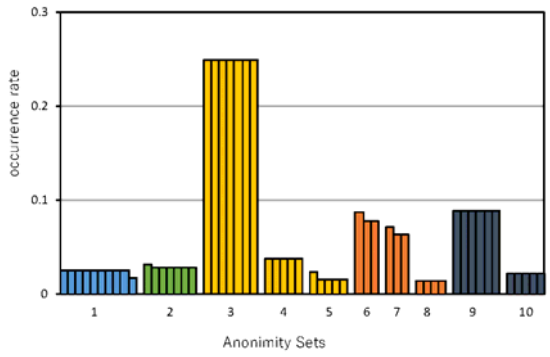

Fig. 2 Generated Anonymity Sets

4. Evaluation

This section evaluates attack success rates and cache hit rates for encrypted packets generated under the condition: $N = 100$, $\Lambda_i = \frac{1}{7}$ and $\mathrm{s}_i = 1, \ldots, 5, t = 0.1, T = 10{,}000, \varepsilon = 0.001$. k is changed from 1 to 6.

4.1 Attack Success Rate

The encrypted packets are generated 1,000 times according to the anonymity set generation algorithm and the frequency attack described in Section 2.3 is applied to the packets. Figure 3 shows the average attack success rates of data pieces in descending order of popularity of data piece.

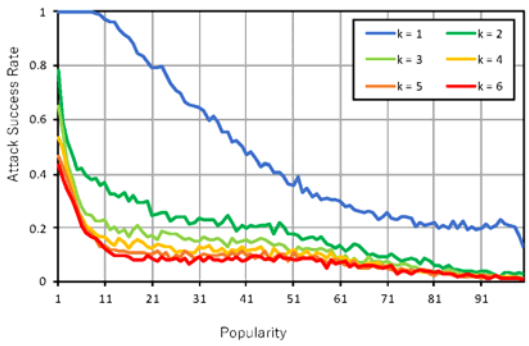

Fig. 3 Attack Success Rates

When $k = 1$, that is, when the single encryption key is used[6], the attack success rates are higher than the other k values. The larger becomes k, the lower the attack success rates become. Obviously, anonymization according to $(k, \varepsilon) -$ anonymitiy is useful to decrease attack success rates against frequency attacks.

4.2 Cache Hit Rate

$(k, \varepsilon) -$ anonymitiy has a negative effect on cache hit rates whereas it increases resiliency against frequency attacks. Expected rates of popular data pieces decrease as shown in Fig. 2. To evaluate the negative impact, we evaluate cache hit rates for the generated encrypted packet sequences. Here, we use an LRU cache with 20 packets. Table 1 shows cache hit rates for k values from 1 to 6. Obviously, the cache hit rates decrease as k becomes larger. High resiliency against frequency attacks and high cache hit rates have a tradeoff relationship.

Table 1 Cache Hit Rates for k Values

k	1	2	3	4	5	6
rate	0.29	0.139	0.087	0.063	0.049	0.041

5. Related Work

Privacy leakage is a long lasting weakness of ICN[7]. It iw well known that attackers easily infer names requested by victims by measuring RTT times of request and reply packets[2,8]. In order to mitigate such risks, a candidate solution is obfuscating/encrypting names of packets. However, Ghali and Wood et al. reveal that obfuscation is not resilient against frequency attacks if popularities of names are public and known by attackers[3,9]. Ghali et al. show vulnerability in the case that a data piece consists of a single packet and it is encrypted by the single key and Enatsu et al. extends their work to the case for multiple packets[6].

The paper addresses the problem in more general cases where a provider uses multiple encryption keys. This approach generates anonymity sets of which expected rates are similar to prevent attackers from differentiating encrypted packet rates. Such anonymization is important for Key Value (KV) systems and Grubbs et al. propose a similar anonymization method where all the rates are the same [10].

6. Conclusion

This paper addresses risks that plain data names are leaked from encrypted packets by frequency attacks in ICN networks. Assuming that multiple encryption keys are used for the same data piece, we develop a frequency attack. Then, we design an anonymization method to provide resiliency against the frequency attack. The method encrypts the same data piece with multiple encryption keys so that at least k encrypted sequences of different data pieces have the same intercepting rate. This improves resiliency against frequency attacks at the cost of decreasing cache hit rates.

Acknowledgments

This work has been supported by JSPS KAKENHI Grant Number 19K22843.

References

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher and B. Ohlman, “Survey of information-centric networking,” IEEE Communication Magazine, vol. 50, no.7, pp.26-36, July 2012.
[2] A. Chaabane, E.D. Cristofaro, M.-A. Kaafar, and E. Uzun, “Privacy in content-oriented networking: threats and countermeasures,” ACM SIGCOMM Computer Communication Review, vol.43, no.3, pp.25–33, 2013.
[3] C. Ghali, G. Tsudik, and C.A. Wood, “(the futility of) data privacy in content-centric networking,” ACM Workshop on Privacy in the Electronic Society, pp.143–152, Oct. 2016.
[4] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, “Named data networking,”ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, pp. 66–73, Jul. 2014
[5] C. Ghali, G. Tsudik, and C.A.Wood, “When encryption is not enough: Privacy attacks in content-centric networking,” Proceedings of ACM ICN 2017, pp.1–10, 2017.
[6] N. Enatsu, Y, Koizumi and T. Hasegawa, “A Study on Vulnerability of Data Name Encryption in ICN Networks against Frequency Attacks,” IEICE Technical Report, IN2019-109, Mar. 2020. (in Japanese)
[7] “Panle:new opportunities and challenges for internet privacy using icn, in proceedings of Am Icn 2016, Sept. 2016,” http://conferences2.igcomm.org/acm-icn/2016/slides/ClosingPanel/all.pdf.
[8] N. Anani, T. Braun and M. Gerla, “Betweenness centrality and cache privacy in information-centric networks,” in Proceedings of ACM ICN, pp.106-116, Sept. 2018.
[9] C.A.Wood and E. Uzun, “Flexible end-to-end content security in ccn,”IEEE CCNC, 2017.
[10] P. Grubbs, A. Khandelwal, L. M, L. Brown, L. Li, and T. Ristenpart, “ Pancake:frequeny smoothing for encrypted data stores,” Proceedings of Usenix Security, pp.2451–2468, Aug. 2020