Traffic Pattern Based Content Leakage Detection for Trusted Content Delivery Networks

Citation:

Hiroki Nishiyama, Desmond Fomo, Zubair Md. Fadlullah, and Nei Kato, "Traffic Pattern Based Content Leakage Detection for Trusted Content Delivery Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 2, pp. 301-309, Feb. 2014.

URL:

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6463398

Traffic Pattern Based Content Leakage Detection for Trusted Content Delivery Networks

Hiroki Nishiyama, *Member, IEEE,* Desmond Fomo, *Student Member, IEEE,* Zubair Md. Fadlullah, *Member, IEEE and* Nei Kato, *Fellow, IEEE*

***Abstract*—Due to the increasing popularity of multimedia streaming applications and services in recent years, the issue of trusted video delivery to prevent undesirable content-leakage has, indeed, become critical. While preserving user privacy, conventional systems have addressed this issue by proposing methods based on the observation of streamed traffic throughout the network. These conventional systems maintain a high detection accuracy while coping with some of the traffic variation in the network (e.g., network delay and packet loss), however, their detection performance substantially degrades owing to the significant variation of video lengths. In this paper, we focus on overcoming this issue by proposing a novel content-leakage detection scheme that is robust to the variation of the video length. By comparing videos of different lengths, we determine a relation between the length of videos to be compared and the similarity between the compared videos. Therefore, we enhance the detection performance of the proposed scheme even in an environment subjected to variation in length of video. Through a test bed experiment, the effectiveness of our proposed scheme is evaluated in terms of variation of video length, delay variation, and packet loss.**

***Index Terms*—Streaming content, leakage detection, traffic pattern, degree of similarity.**

I. Introduction

In recent years, with the rapid development of broadband technologies and the advancement of high-speed wired/wireless networks, the popularity of real-time video streaming applications and services over the Internet has increased by leaps and bounds. YouTube and Microsoft Network (MSN) video are notable examples of such applications. They serve a huge population of users from all around the world with diverse contents, ranging from daily news feeds to entertainment feeds including music, videos, sports, and so forth, by using streaming transmission technologies. In addition, real-time video streaming communications such as web conference [1], [2], [3] in intra-company networks or via Internet with Virtual Private Networks (VPNs) are being widely deployed in a large number of corporations as a powerful means of efficiently promoting business activities without additional costs [4].

A crucial concern in video streaming services is the protection of the bit stream from unauthorized use, duplication and distribution. One of the most popular approaches to prevent undesirable contents distribution to unauthorized users and/or to protect authors' copyrights is the Digital Rights Management (DRM) technology. Most DRM techniques employ cryptographic or digital watermark techniques [5], [6], [7], [8], [9]. However, this kind of approaches have no significant effect on re-distribution of contents, decrypted or restored at the user-side by authorized yet malicious users. Moreover, re-distribution is technically no longer difficult by using Peer to Peer (P2P) streaming software [10]. Hence, streaming traffic may be leaked to P2P networks.

On the other hand, packet filtering by firewall-equipped egress nodes is an easy solution to avoid leakage of streaming contents to external networks. In this solution, the packet header information (e.g., destination and source Internet Protocol (IP) addresses, protocol type, and port number of outgoing traffic) of every streamed packet is inspected. In case the inspected packets do not verify the pre-defined filtering policy, they are blocked and dropped [11]. However, it is difficult to entirely prevent streaming content leakage by means of packet filtering alone because the packet header information of malicious users is unspecified beforehand and can be easily spoofed.

In this work, we focus on the illegal re-distribution of streaming content by an authorized user to external networks. The existing proposals in [12], [13], [14] monitor information obtained at different nodes in the middle of the streaming path. The retrieved information are used to generate traffic patterns which appear as unique waveform per content [15], just like a fingerprint. The generation of traffic pattern does not require any information on the packet header, and therefore preserves the user's privacy. Leakage detection is then performed by comparing the generated traffic patterns. However, the existence of videos of different length in the network environment causes a considerable degradation in the leakage detection performance. Thus, developing an innovative leakage detection method robust to the variation of video lengths is, indeed required. In this paper, by comparing different length videos, we determine a relationship between the length of videos to be compared and their similarity. Based on this relationship, we determine decision threshold enabling accurate leakage detection even in an environment with different length videos. The remainder of the paper is organized as follows. A typical video leakage scenario, detection system and procedures are described in section II. In section III, first we depict the drawback of the existing scheme due to the variation of video length in realistic environment, then we described the proposed leakage detection scheme, and we evaluate its calculation cost in comparison to that of the existing scheme. Furthermore, in section IV, we evaluate the effectiveness and the accuracy of the proposed scheme with respect to different length videos, and its robustness to network environment changes. Finally, we conclude this paper

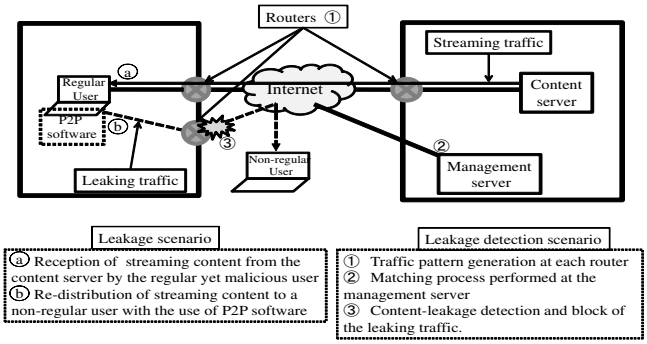


Fig. 1. Overview of a leakage scenario and leakage detection scenario.

in Section V.

II. Content Leakage Detection

In this section, we first take a look at a typical video leakage scenario, and we present an overview of existing traffic pattern based leakage detection technologies.

A. Typical video leakage scenario

Due to the popularity of streaming delivery of movies, development of P2P streaming software has attracted much attention. These technologies enhance the distribution of any type of information over the Internet [13]. A typical content-leakage scenario can be described by the following steps as depicted in Fig. 1. First, a regular user in a secure network receives streaming content from a content server. Then, with the use of a P2P streaming software, the regular yet malicious user redistributes the streaming content to a non-regular user outside its network. Such content-leakage is hardly detected or blocked by watermarking and DRM based techniques.

B. Leakage detection procedures

Throughout the video streaming process, the changes of the amount of traffic appear as a unique waveform specific to the content. Thus by monitoring these information retrieved at different nodes in the network, content-leakage can be detected.

An overview of the network topology of the proposed leakage detection system is shown in Fig. 1. This topology consists of two main components, namely the traffic pattern generation engine embedded in each router, and the traffic pattern matching engine implemented in the management server. Therefore each router can observe its traffic volume and generate traffic pattern. meanwhile, the traffic pattern matching engine computes the similarity between traffic patterns through a matching process, and based on specific criterion, detects contents leakage. The result is then notified to the target edge router in order to block leaked traffic.

C. Pattern generation algorithm

Here, we describe the traffic pattern generation process performed in conventional methods. Traffic pattern generation process is based on a either time slot-based algorithm or a packet size-based algorithm. The traffic pattern generated is expressed as an N-dimension vector as follows,

$$X_N = (x_1, x_2, ..., x_N)^T, \tag{1}$$

where x_i indicates the volume of the i^{th} chunk, and N is the total number of chunks.

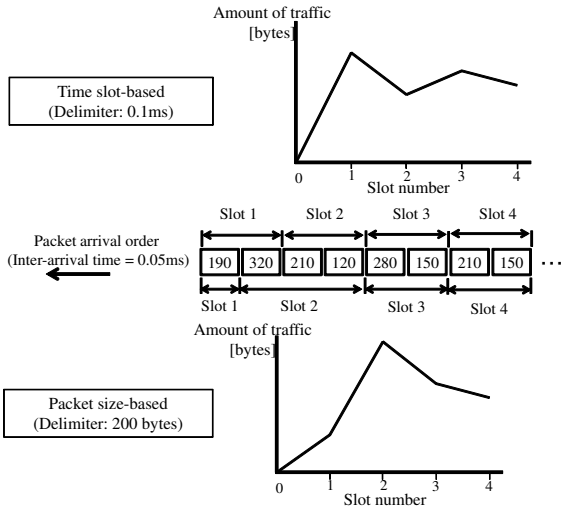


Fig. 2. Traffic pattern generation process.

Time slot-based algorithm is a straightforward solution to generate traffic patterns by summing the amount of traffic arrival during a certain period of time, Δt. In case some packets are delayed, they may be stored over the following slot, x_{i+1}, instead of the primary slot, x_i. Therefore, delay and jitter of packets distorts the traffic pattern, and as a consequence, decreases the accuracy in pattern matching. Moreover, time slot-based algorithm is affected by packet loss.

Packet size-based algorithm defines a slot as the summation of amount of arrival traffic until the observation of a certain packet size. This algorithm only make use of the packet arrival order and packet size, therefore is robust to change in environment such as delay and jitter. However, packet size-based algorithm shows no robustness to packet loss.

Fig. 2. describes an example of time slot-based generation process and packet size-based generation process. Here, the time-slot, Δt is set to 0.1 milliseconds for time slot-based process, while for packet size-based process, slots are generated by summing the amount of arrival traffic until observing a packet of size less than 200 bytes.

D. Pattern matching algorithm

In pattern recognition, the degree of similarity is defined to be the similarity measure between patterns [16]. The server-side traffic patterns represents the original traffic pattern and is expressed as $X_S = (x_1, x_2, ..., x_S)^t$ according to Eq. 1. The user-side traffic pattern is expressed as $Y_U = (y_1, y_2, ..., y_U)^t$. Here, S and U are number of slots, and the length of the user-side observation is shorter than that of the server-side, i.e., $S > U$.

The outline of comparison of traffic patterns is shown in Fig. 3. The comparison of traffic patterns consists of three

TABLE I
COMPARISON OF EXISTING LEAKAGE DETECTION METHODS

	Traffic pattern generation algorithm	Traffic pattern matching algorithm	Decision threshold	Robustness
T-TRAT	Time slot-based	Cross-correlation matching	Dynamic (Chebyshev based)	-
P-TRAT	Packet size-based		Static (Fixed value)	Delay and jitter
DP-TRAT		DP matching		Delay, jitter, packet loss

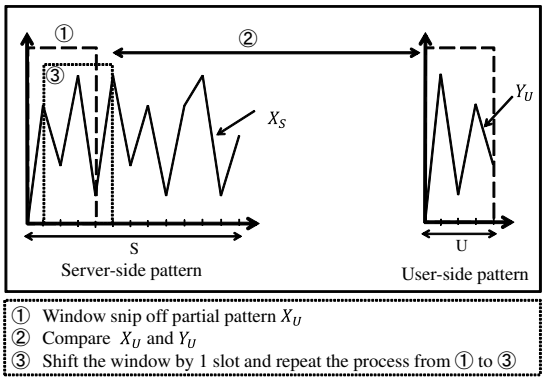


Fig. 3. Traffic pattern matching.

steps. First, we set a window of size, U, which snips off a partial pattern, X_U, from the server-side traffic pattern, X_S. Next, we compute the similarity between the partial pattern, X_U, and the user-side pattern, Y_U, (partial similarity). The window is then moved from left to right by one slot. These three steps are repeated until the window reaches the rightmost part of the server-side pattern. Thus, we obtain $(S-U+1)$ values of similarity. The maximum value is then retrieved and represents the degree of similarity of the compared videos.

The fundamental method to quantify the similarity of traffic patterns called Cross-correlation matching algorithm, consist of computing the cross-correlation coefficient, which is used as a metric of similarity between the various traffic patterns. Before calculating the similarity between the partial pattern X_U and the server-side pattern Y_U, X_U and Y_U are normalized as

$$X'_U = \begin{pmatrix} (x_1-\overline{x})/s_x \\ (x_2-\overline{x})/s_x \\ \vdots \\ (x_U-\overline{x})/s_x \end{pmatrix}, \qquad Y'_U = \begin{pmatrix} (y_1-\overline{y})/s_y \\ (y_2-\overline{y})/s_y \\ \vdots \\ (y_U-\overline{y})/s_y \end{pmatrix}. \quad (2)$$

Here, $\overline{x}$ and $\overline{y}$ are the means of each vector. s_x and s_y are the standard deviations. After normalization, the means and variance of X'_U and Y'_U are zero and one, respectively. The cross-correlation coefficient between X_U and Y_U is given by the following equation.

$$R_{X_UY_U} = \frac{{X'_U}^t Y'_U}{\sqrt{\|X'_U\|^2\|Y'_U\|^2}}, \qquad -1 \le R_{X_UY_U} \le 1. \quad (3)$$

Another pattern matching algorithm is the Dynamic Programming (DP) matching based on the dynamic programming technique [18], [19]. DP matching utilizes the distance [20] between the compared patterns in U-dimensional vector space as metric representing their similarity.

E. Leakage detection criterion

The cross correlation matching algorithm is performed on both the traffic patterns generated through time slot-based algorithm and those generated through packet size-based algorithm [12]. The similarity data obtained from the matching of time slot-based generated traffic patterns are considerably small and their distribution is considered to be normally distributed around zero, since the distribution of cross-correlation coefficient values of two random waveforms is approximated to a normal distribution [17]. Therefore, [12] uses a dynamic decision threshold based on the Chebyshev's inequality, and given by the following equation:

$$\Theta = \min(\mu_R + 4\sigma_R, 1.0), \quad (4)$$

where μ_R and σ_R represent the mean and variance of the set of cross-correlation coefficient $R_{X_UY_U}$, respectively. Here, whether or not compared patterns are similar is decided by comparing the maximum value of $R_{X_UY_U}$ with Θ from Eq. 4. Meanwhile, during the matching process of packet size-based generated traffic patterns, the similarity resulting from the comparison of different videos is considerably small, while the similarity resulting from the comparison of similar videos is considerably large. A suitable fixed value is therefore used as the decision threshold [12]. To determine whether or not the compared traffic patterns are similar, the maximum value of $R_{X_UY_U}$ is retrieved and compared to the decision threshold, i.e., $\max(R_{X_UY_U}) >$ threshold, which indicates that the compared traffic patterns are similar.

On the other hand, the DP matching algorithm is perfomed on traffic patterns generated through packet size-based algorithm. therefore, a fixed predefined value is used as the decision threshold [13]. Whether or not patterns are similar is decided by comparing the distance computed through DP matching with the decision threshold, i.e., the distance less than the threshold indicates that the compared traffic patterns are similar.

F. Summary of the conventional methods

The conventional approaches, namely, Time slot-based Traitor Tracing (T-TRAT), Packet size-based Traitor Tracing (P-TRAT) and Dynamic Programming based Traitor Tracing (DP-TRAT), based on the aforementioned algorithms are summarized in Table I. The time slot-based pattern generation algorithm used in T-TRAT is influenced by packet delay and jitter, which deteriorate the user-side traffic pattern. On the other hand, P-TRAT and DP-TRAT utilize a traffic pattern generation method based on packet size instead of time-slot. As a result, P-TRAT and DP-TRAT show robustness against packet delay and jitter. The cross-correlation coefficient is

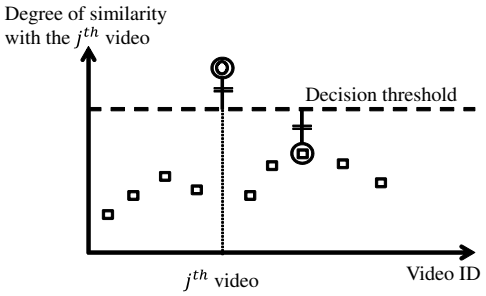


Fig. 4. Description of decision threshold in existing leakage detection schemes.

widely use in pattern recognition. However, it is considerably influenced by packet loss that may occur between the streaming server and the user. Meanwhile, DP matching dynamically alleviates this issue, and shows high robustness to variation in network environment such as the occurrence of packet loss. The determination of the predefined decision threshold used in P-TRAT and DP-TRAT is depicted in Fig. 4 by computing the median between the degree of similarity resulting from the comparison with the same video and the maximum value of the degree of similarity resulting from the comparison with different videos.

III. Enhancement of Detection technique to handle video contents of different lengths

Among the conventional methods, DP-TRAT method shows high robustness to packet delay, jitter, and packet loss. However, the existence of videos of different lengths subjected to time variation in real content delivery environment causes DP-TRAT's accuracy to decrease. In this section, we take a look at the issue caused by the existence of different length videos in network environments. While focussing on DP-TRAT, we introduce a new threshold determination method based on an exponential approximation, and evaluate the computation cost of both the proposed scheme and an eventual enhancement of the previous scheme.

A. Issue due to different lengths of videos

Traffic patterns of streaming videos represent the skeleton carrying their characteristics [21], and are unique per content. Therefore, the longer the traffic pattern is, the more information on the video it displays. In conventional methods, it is assumed that a certain length of content can always be obtained through the network for all contents. Therefore it is possible to utilize a fixed decision threshold in both P-TRAT and DP-TRAT methods. However, there is no such guarantee in actual network environments. Fig. 5. shows an illustration of the occurrence of an erroneous decision in a network environment with different length videos.

B. Exponential approximation-based threshold determination and leakage detection

1) Threshold determination process (pre-processing)

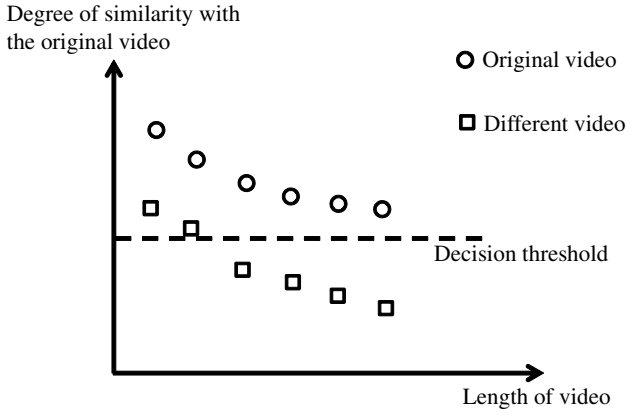


Fig. 5. Example of erroneous decision in comparison of different length videos

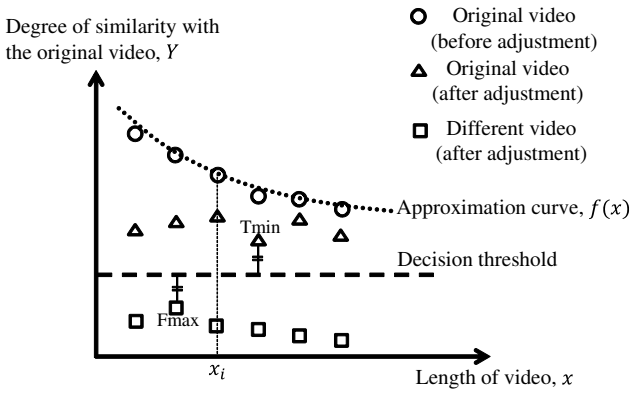


Fig. 6. Determination of the decision threshold for detecting leakage.

Fig. 6 depictes the determination of the decision threshold. From the original video, we create portions of videos of varying lengths, and we generate their corresponding traffic patterns. These patterns are then compared to the original traffic pattern to perform a sampling of the length of videos and their corresponding degree of similarity. With the distribution of the sampling result, we perform an exponential approximation [22] of the form

$$f(x) = exp(\alpha \cdot x + \beta). \tag{5}$$

This exponential curve represents the estimation of the degree of similarity with the original video for a certain length x. It is computed based on the least-squares method. Where α and β are given by Eq. 6 and Eq. 7.

$$\alpha = \frac{n \cdot C - B \cdot D}{n \cdot A - D^2}, \tag{6}$$

$$\beta = \frac{A \cdot B - C \cdot D}{n \cdot A - D^2}, \tag{7}$$

here,

$$A = \sum_{i=0}^{n} x_i^2, \qquad B = \sum_{i=0}^{n} \ln(f(x_i)),$$
$$C = \sum_{i=0}^{n} x_i \cdot \ln(f(x_i)), \qquad D = \sum_{i=0}^{n} x_i,$$

where, (x_i) $i = 0, 1, ..., n$ represent the set of video lengths.

We adjust the degree of similarity obtained through the matching process, by dividing it with the corresponding value obtained using the approximation curve defined in Eq. 5. After the adjustment, we retrieve the minimal value, T_{min}. The adjustment process is given by the following equation:

$$Y' = \frac{Y}{f(x)}, \tag{8}$$

where x and Y represent the pattern size, and the degree of similarity obtained through the matching process, respectively. Similarly, we compare the original traffic pattern with traffic pattern of portions of video obtained from different videos. Likewise following Eq. 8, we adjust the resulting degree of similarity and retrieve the maximum value, F_{max}. In order to perform an accurate video comparison, we consider the median given by Eq. 9 as the decision threshold.

$$Threshold = \frac{(F_{max} + T_{min})}{2}. \tag{9}$$

By realizing this process for every video, we provide criteria for adequate decision threshold specific to each video.

2) Leakage detection

Before the leakage detection process, we compute the approximation curve based on the original traffic pattern. Then, we compare the target traffic pattern to the original traffic pattern, and we adjust the obtained degree of similarity using the approximation curve following Eq. 8, where x is the size of the target traffic pattern. Finally, we compare the adjusted degree of similarity to the decision threshold specific to the original video, and detect whether or not there is a leakage. The problem in comparison of short length videos is then solved, and a flexible and accurate leakage detection method is possible.

C. Cost evaluation

Here, we analyse the cost of the proposed scheme and that of an eventual enhancement of the previous scheme. It is worth mentioning that in this paper, we define the cost in terms of number of matching operations performed in the threshold determination process. The use of the approximation curve enable accurate comparison independently of the length of video.

Let N_V denote the total number of videos in our environment. We consider all the corresponding traffic patterns to be of size L. It is worth noting that the size of a pattern is determined by the number of slots. Let $\Omega = \{l, l \leq L\}$ the set of all the possible sizes of patterns retrieved from the original pattern of size L. As described in section II, while comparing a pattern of specific size l and the original pattern of size L, we perform $|L - l| + 1$ matching.

For a specific length, l, let M be the number of patterns of size l to be retrieved from the original pattern. The decision threshold associated to the size l is computed by comparing the M patterns of size l retrieved from the original traffic pattern and the original traffic pattern. Let $match(l)$ be the number of matching corresponding to a specific size l. $match(l)$ is then given by

$$match(l) = M \times (|L - l| + 1). \tag{10}$$

1) Enhancement of the previous scheme

To efficiently handle the comparison between patterns of various sizes, it is necessary to consider every size less or equal to L. For a given video, V_x, we compare the M patterns of size l retrieved from the original traffic pattern of each other video present in our environment and the traffic pattern of the video V_x. The process being performed for the N_V different videos in our environment, we have a total number of matching, Γ_1, given by

$$\Gamma_1 = N_V \times N_V \sum_{l \in \Omega} match(l). \tag{11}$$

2) Proposed scheme

The proposed scheme is based on computing an approximation curve of the distribution of the pattern size and their associated degree of similarity. Based on the computed curve, we determine the decision threshold specific to each video in our streaming environment. To compute such a curve, we focus on a certain number of pattern size less or equal to L. Let ω be the following sub-set

$$\omega = \{l_i, i = 1, 2, ..., k, l_i \in \Omega, l_i < l_{i+1}\}.$$

The total number of matching, Γ_2, necessary to determine decision threshold specific to each video in our environment is given by

$$\Gamma_2 = N_V(A_C + D_T) \tag{12}$$

where, $A_C = \sum_{l \in \omega} match(l)$ and $D_T = (N_V - 1) \sum_{l \in \omega} match(l)$ represent the number of matching necessary for the computation of the approximation curve and the number of matching necessary to determine the decision threshold specific to a video, respectively.

3) Cost Comparison

Here we show that our proposed scheme is cost effective in comparison with an eventual enhancement of the previous scheme. From Eq. 11 and Eq. 12, we can obtain the following equation,

$$\Delta\Gamma = \Gamma_1 - \Gamma_2 = {N_V}^2 (\sum_{l \in \Omega} match(l) - \sum_{l \in \omega} match(l)). \tag{13}$$

It is worth noting that the expression of $match(l)$ in Eq. 10 includes M, which has different range of value per size l. Patterns of size l are retrieved from a given traffic pattern of size L, similar as the macthing process depicted in Fig. 3. Therefore M has to satisfy the following condition,

$$M \leq L - l + 1.$$

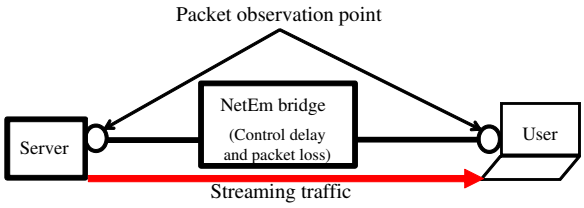


Fig. 7. Topology of our conducted experiment.

When it is supposed that M is bounded by a constant value $M_{\max}$, in order to achieve a high detection accuracy with a low computation cost, M is defined as follows,

$$M = \min(|L - l| + 1), M_{\max}). \tag{14}$$

Thus, M can be regarded as a constant value $M_{\max}$ in realistic environments, because the traffic pattern observed at the user-side router, l, is considerably shorter than the size of the original pattern, L, i.e., $M_{\max} \ll |L - l|$. As a result, we get

$$\Delta\Gamma = {N_V}^2 M_{\max}(\sum_{l \in \Omega - \omega} |L - l| + 1). \tag{15}$$

Here, since ω is a subset of Ω, the dimension of the set $(\Omega - \omega)$ is equal to $(L - k)$. As a conclusion, we can derive the following expression,

$$\Delta\Gamma = O({N_V}^2 L^2). \tag{16}$$

Based on the above result, it is clear that in a realistic environment, our proposed scheme achieves a lower computation cost in comparison to the enhancement of the previous scheme, and it becomes much more effective as the number and or size of videos increase.

IV. Performance evaluation

In this section, we describe the performance evaluation experiment carried out using a real network environment. We evaluate the effectiveness and the accuracy of the use of a dynamic decision threshold in a network environment with videos of different length. Moreover, we evaluate the robustness of our scheme to network environment changes. The proposed decision threshold determination technique is implemented into the DP-TRAT which employs the packet size-based traffic generation algorithm and the DP-matching algorithm, because DP-TRAT shows high robustness to network environment changes compare to other schemes.

A. Experiment setup

Fig. 7. displays the testbed used for the experiment. Streaming contents are sent from the delivery server to the user, and the traffic is observed at the server side and the user side. Traffic patterns are then generated at the packet observation points as displayed in Fig. 7, and sent to the server, where the matching process is performed. To handle variation in network environment such as delay, jitter and packet loss, we placed the NetEm bridge [23] between the server and the user. P-TRAT and DP-TRAT based detection performances are used as comparison to our proposed method.

As an evaluation metric for the performance of the proposed leakage detection method, we define the accuracy, P_r, and the recall ratio, R_e [24]. The accuracy P_r represents the ratio of outflow correctly detected as similar to the original video on the outflow seen as similar to the original, including erroneous judgement. The recall ratio R_e is an index of completeness, representing the ratio of outflow correctly detected as similar to the original video on the outflow of the targeted contents. These index are widely used in recognition techniques and performance evaluation of web information retrieval systems. P_r and R_e are defined as follow:

$$P_r = \frac{C}{A}, \tag{17}$$

$$R_e = \frac{C}{W}. \tag{18}$$

Where, C, A and W represent the number of traffic correctly indicating that the compared video are the same, the number of traffic indicating that the compared videos are the same, including both correct and incorrect judgement, and the number of traffic showing that the videos to be compared are supposed to be the same, respectively.

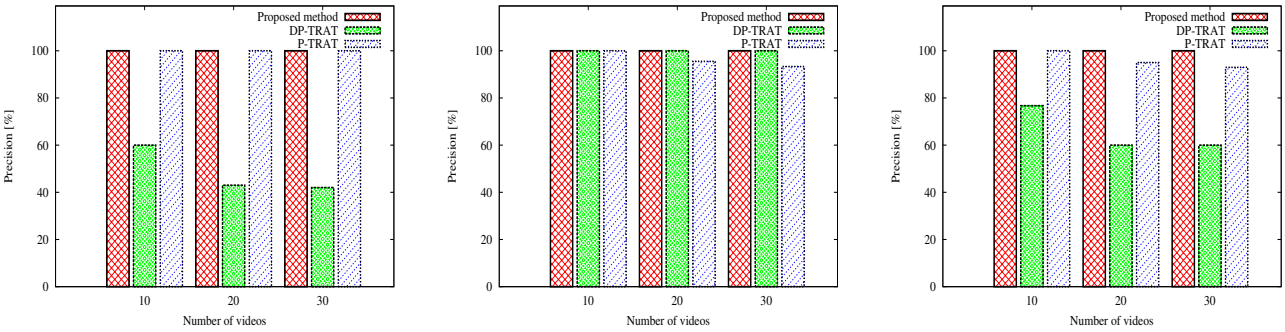
It is worth noting that the bigger the accuracy and the recall ratio, the better the leakage detection performance. However, a trade-off relation exist between the accuracy and the recall ratio. We consider both and define their harmonic mean F-measure : F [25]. F-measure is given by the following equation,

$$F = \frac{2 \times P_r \times R_e}{P_r + R_e} \quad . \tag{19}$$

B. Performance for different lengths of videos

In this experiment, we use a set of different videos having the same length and can be perfectly distinguished using the conventional methods, P-TRAT and DP-TRAT. From this set, we generate portions of video of different lengths varying from 30 to 300 seconds. From the generated portions of videos, we randomly choose and send 10, 20 and 30 videos from the server to the user. We then observe the amount of traffic, generate the traffic pattern and perform the matching process. In other word, the performance degradation observed in this experiment can be considered to be caused by the existence of videos with different lengths. P-TRAT and DP-TRAT are used for comparison.

Fig. 8(a) demonstrates that with the DP-TRAT, the increase in the number of videos decreases the accuracy. the absence of an adequate method to set the decision threshold handling videos of different length causes the occurrence of erroneous decision in the detection performance of the DP-TRAT. Fig. 8(b) shows that with the conventional methods (P-TRAT, DP-TRAT), the recall ratio is slightly affected by the variation of video length. Fig. 8(c) shows that compare to the conventional methods, the proposed scheme is not affected by the variation of video length.



(a) Accuracy (b) Recall ratio (c) F-measure

Fig. 8. Performance on variation of videos length.

C. Robustness to network environment changes

To evaluate the robustness of the proposed scheme to the variation in network environment, we perform two experiments. Here, we consider a network environment similar to the previous, with 30 videos of lengths varying from 30 to 300 seconds. For the first experiment, we generate delay at the NetEm varying from 0 to 200ms every 25ms. Fig. 9. shows that none of the methods is affected by delay. This is due to the fact that all of these methods generate traffic patterns using the packet size-based generation algorithm, which shows robustness against packet delay jitter.

For the second experiment, with the NetEm, we generate packet loss. The generated packet loss rate varies from 0.1% to 5%. Fig. 10(a) shows that the accuracy in both the conventional methods and the proposed method is not affected by packet loss. While Fig. 10(b) shows that for P-TRAT, the recall ratio decreases rapidly when the packet loss exceeds 0.3%. Thus, P-TRAT that uses the cross-correlation matching technique, deals ineffectively with variation of traffic amount per slot due to packet loss. From Fig. 10(c), we can see that DP-TRAT shows a fair detection performance, while being slightly affected by packect loss. Meanwhile, our proposed method is not affected by packet loss, and keep a high detection performance.

These two experiments show that the proposed method outperforms the conventional methods. Moreover, it results in high robustness against change in network environment.

V. Conclusion

The content leakage detection system based on the fact that each streaming content has a unique traffic pattern is an innovative solution to prevent illegal re-distribution of contents by a regular, yet malicious user. Though three typical conventional methods, namely T-TRAT, P-TRAT, DP-TRAT, show robustness to delay, jitter or packet loss, the detection performance decreases with considerable variation of video lengths. This paper attempts to solve these issues by introducing a dynamic leakage detection scheme. Moreover, in this paper, we investigate the performance of the proposed method under a real network environment with videos of different lengths. The proposed method allows flexible and accurate streaming content leakage detection independent of the length of the streaming content, which enhances secured and trusted content delivery.

References

[1] Y. Chu, S. G. Rao, S. Seshan and H. Zhang, “Enabling conferencing applications on the Internet using an overlay multicast architecture,” in Proc. ACM SIGCOM, pp.55-67, California, USA, Aug. 2001.

[2] Z. Yang, H. Ma, and J. Zhang, “A dynamic scalable service model for SIP-based video conference,” in Proc. 9th International Conference on Computer Supported Cooperative Work in DE.

[3] Y. Chu, S. G. Rao, S. Seshan and H. Zhang, “Enabling conferencing applications on the Internet using an overlay multicast architecture,” in Proc. ACM SIGCOM, pp. 55-67, California, USA, Aug. 2001.

[4] O. Adeyinka, “Analysis of IPSec VPNs Performance in A Multimedia Environment,” School of Computing and Technology, University of East London.

[5] E.I. Lin, A.M. Eskicioglu, R.L. Lagendijk, and E.J. Delp, “Advances in digital video content protection,” Proc. IEEE, vol.93, no.1, pp.171-183, Jan. 2005

[6] S. Craver, N. Memon, B. L. Yeo, and M. M. Yeung, “Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications,” IEEE J. Sel. Areas Commun., vol.16, no.4, pp.573-586, May 1998.

[7] M. Barni and F. Bartolini, “Data hiding for fighting piracy,” IEEE Signal Process. Mag., vol.21, no.2, pp.28-39, Mar. 2004.

[8] K. Su, D. Kundur, and D. Hatzinakos, “Statistical invisibility for collusion-resistant digital video watermarking,” IEEE Trans. Multimedia, vol.7, no.1, pp.43-51, Feb. 2005.

[9] E. Diehl and T. Furon, “Watermark: Closing the analog hole,” in Proc. IEEE Int. Conf. Consumer Electronics, pp.52-53, 2003.

[10] Y. Liu, Y. Guo, and C. Liang, “A survey on peer-to-peer video streaming systems,” Peer-to-Peer Networking and Applications, Vol.1, No.1, pp.18-28, Mar. 2008.

[11] E. D. Zwicky, S. Cooper, and D. B. Chapman, “Building Interent Firewalls (2nd ed.),” O’Reilly and Associates, 2000.

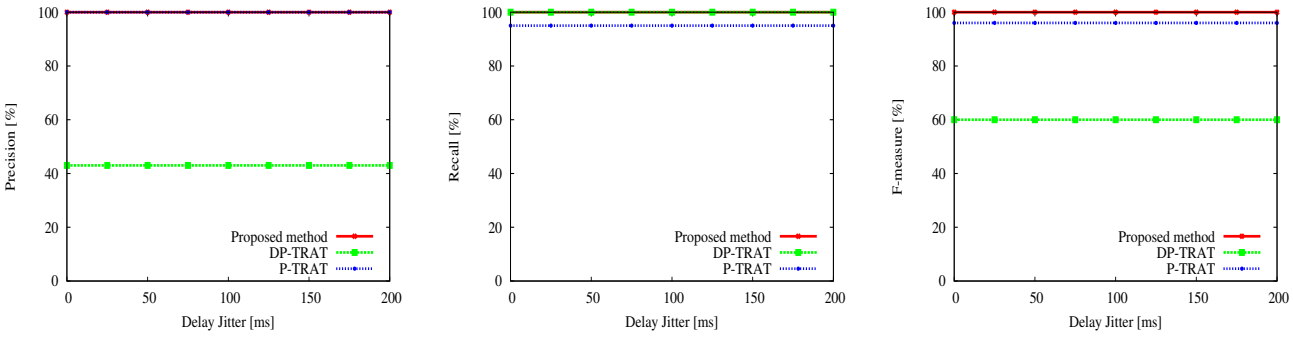
[12] M. Dobashi, H. Nakayama, N. Kato, Y. Nemoto, and A. Jamalipour, “Traitor Tracing Technology of Streaming Contents Delivery using Traffic Pattern in Wired/Wireless Environments,” in Proc. IEEE Global Telecommunications Conference, pp.1-5, San Francisco, USA, Nov./Dec. 2006.

[13] K. Matsuda, H. Nakayama, and N. Kato, “A Study on Streaming Video Detection using Dynamic Traffic Pattern,” IEICE Transactions on Communications (Japanese Edition), vol.J19-B, no.02, 2010.

[14] Atsushi Asano, Hiroki Nishiyama, and Nei Kato, “The Effect of Packet Reordering and Encrypted Traffic on Streaming Content Leakage Detection (Invited Paper),” International Conference on Computer Communication Networks 2010 (ICCCN 2010), Zurich, Switzerland, Aug. 2010.

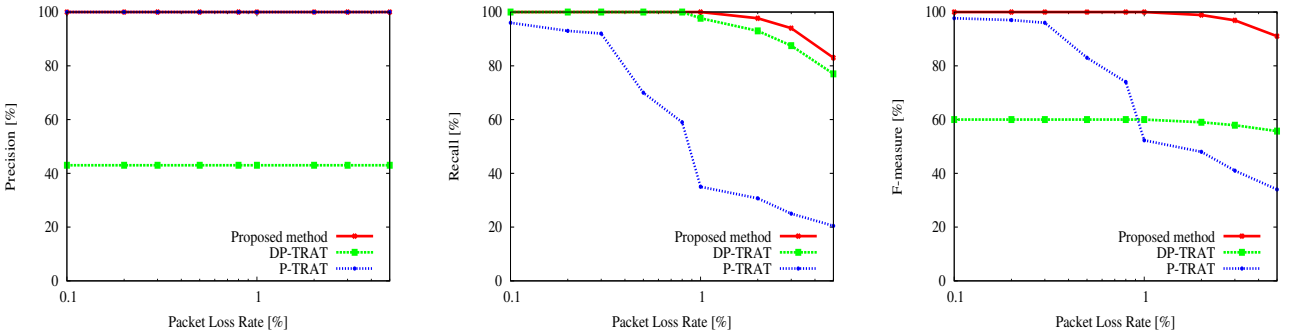
[15] S. Amarasing and M. Lertwatechakul, “The Study of Streaming Traffic behavior,” KKU Engineering Journal, vol.33, no.5, pp.541-553, Sept.-Oct. 2006.

[16] Y. Gotoh, K. Suzuki, T. Yoshihisa, Hideo Taniguchi, and M. Kanazawa, “Evaluation of P2P Streaming Systems for Webcast,” 6th International Conference on Digital Information Management.



(a) Accuracy (b) Recall ratio (c) F-measure

Fig. 9. Performance on delay of packets.

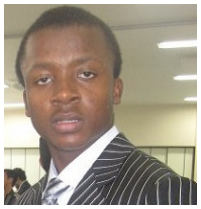


(a) Accuracy (b) Recall ratio (c) F-measure

Fig. 10. Performance on packet loss.

[17] R. Duda, P. Hart and D. Stock, *Pattern Classification,* 2nd ed. New York: Wiley Interscience, 2000.
[18] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic Programming for Detecting, Tracking, and Matching Deformable Contours," in Proc. IEEE Transactions on Pattern Analysis and Matchine Intelligence, vol.17, no.3, pp.294-302, Mar. 1995.
[19] R. S. Naini and Y. Wang, "Sequential traitor traicing," IEEE Trans. Inf. Theory, vol.49, no.5, pp.1319-1326, May 2003.
[20] Y. Zhang, P. Ma, and X. Su, "Pattern Recognition Using Interval-valued intuitionistic Fuzzy set and Its similarity Degree," IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009.
[21] A. Golaup, and H. Aghvami, "A multimedia traffic modeling framework for simulation-based performance evaluation studies," Computer Network, vol. 50, no. 12, pp. 2071-2087, 2006.
[22] Ram M. Narayanan, "First Order Exponential Approximation for Small Arguments," IEEE Magazine on Aerospace and Electronic Systems, vol. 9, no. 2, pp. 33-35, Feb. 1994.
[23] S. Hemminger, "Network Emulation with NetEm," in Proc. Linux Conference Australia, Canbera, Australia, April 2005.
[24] R. Zanibbi, D. Blostein, and J. R. Cordy, "Historical recall and precision: summarizing generated hypotheses," in Proc. 8th International Conference on Document Analysis and Recognition, pp. 202-206, 2005.
[25] B. Larsen and C. Aone, "Fast and effective text mining using linear-time document clustering," in Proc. the 5th ACM SIGKDD, pp.16-22, California, USA, Aug. 1999.

Hiroki Nishiyama (M'08) (bigtree@it.ecei.tohoku.ac.jp) received his M.S. and Ph.D. in Information Science from Tohoku University, Japan, in 2007 and 2008, respectively. He was a Research Fellow of the prestigious Japan Society for the Promotion of Science (JSPS) until the completion of his PhD, following which he went on to become an Assistant Professor at the Graduate School of Information Sciences (GSIS) at Tohoku University. He was promoted to his current position of an Associate Professor at GSIS in 2012, when he was just 29 years old. He was acclaimed with the Best Paper Awards in many international conferences including IEEE's flagship events, namely the IEEE Wireless Communications and Networking Conference in 2012 (WCNC'12) and the IEEE Global Communications Conference in 2010 (GLOBECOM'10). He is a young yet already prominent researcher in his field as evident from his valuable contributions in terms of many quality publications in prestigious IEEE journals and conferences. He was also a recipient of the IEICE Communications Society Academic Encouragement Award 2011 and the 2009 FUNAI Foundation's Research Incentive Award for Information Technology. He received the Best Student Award and Excellent Research Award from Tohoku University for his phenomenal performance during the undergraduate and master course study, respectively. His research covers a wide range of areas including traffic engineering, congestion control, satellite communications, ad hoc and sensor networks, and network security. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).



Desmond Fomo (S'12) (desmond@it.ecei.tohoku.ac.jp) received his B.Sc degree in mathematics from the University of Yaounde I, Cameroon, in 2008. Currently, he is pursuing the M.S. degree at the Graduate School of Information Sciences, Tohoku University, Japan. His research interests include content delivering networks, and load balancing among multipath.

Zubair Md. Fadlullah (S'07, M'11) (zubair@it.ecei.tohoku.ac.jp) received B.Sc. degree with Honors in computer sciences from the Islamic University of Technology (IUT), Bangladesh, in 2003, and M.S. and Ph.D. degrees from the Graduate School of Information Sciences (GSIS), Tohoku University, Japan, in 2008 and 2011, respectively. Currently, he is serving as an Assistant Professor at Tohoku University. His research interests are in the areas of smart grid, network security, intrusion detection, game theory, and quality of security service provisioning mechanisms. Dr. Fadlullah was a recipient of the prestigious Deans and Presidents awards from Tohoku University in March 2011.

Nei Kato (A'03-M'04-SM'05-F'13) (kato@it.ecei.tohoku.ac.jp) received his Bachelor Degree from Polytechnic University, Japan in 1986, M.S. and Ph.D. Degrees in information engineering from Tohoku University, Japan, in 1988 and 1991, respectively. He joined Computer Center of Tohoku University at 1991, and has been a full professor with the Graduate School of Information Sciences since 2003. He has been engaged in research on satellite communications, computer networking, wireless mobile communications, smart grid, image processing, and pattern recognition. He has published more than 300 papers in peer-reviewed journals and conference proceedings. He currently serves as the Vice Chair of IEEE Ad Hoc & Sensor Networks Technical Committee, the Chair of IEEE ComSoc Sendai Chapter, the steering committee member of WCNC and voting member of GITC, an editor of IEEE Wireless Communications(2006-), IEEE Wireless Communications(2006-), IEEE Network Magazine(2012-), IEEE Transactions on Wireless Communications(2008-), IEEE Transactions on Vehicular Technology(2010-), IEEE Trans. on Parallel and Distributed Systems. He has served as the Chair of IEEE Satellite and Space Communications Technical Committee(2010-2012), a co-guest-editor of several Special Issues of IEEE Wireless Communications Magazine, a symposium co-chair of GLOBECOM07, ICC10, ICC11, ICC12, Vice Chair of IEEE WCNC10, WCNC11, ChinaCom08, ChinaCom09, Symposia co-chair of GLOBECOM12, ICC14, and workshop co-chair of VTC2010. His awards include Minoru Ishida Foundation Research Encouragement Prize(2003), Distinguished Contributions to Satellite Communications Award from the IEEE Communications Society, Satellite and Space Communications Technical Committee(2005), the FUNAI information Science Award(2007), the TELCOM System Technology Award from Foundation for Electrical Communications Diffusion(2008), the IEICE Network System Research Award(2009), the IEICE Satellite Communications Research Award(2011), the KDDI Foundation Excellent Research Award(2012), IEICE Communications Society Distinguished Service Award(2012), IEEE GLOBECOM Best Paper Award(twice), IEEE WCNC Best Paper Award, and IEICE Communications Society Best Paper Award(2012). Besides his academic activities, he also serves on the expert committee of Telecommunications Council, Ministry of Internal Affairs and Communications, and as the chairperson of ITU-R SG4 and SG7, Japan. Nei Kato is a Distinguished Lecturer of IEEE Communications Society(2012-213) and the co-PI of A3 Foresight Program(2011-2014) funded by Japan Society for the Promotion of Sciences(JSPS), NSFC of China, and NRF of Korea. He is a fellow of IEICE.