Unsupervised Analysis of Encrypted Video Traffic Based on Levenshtein Distance

Luming Yang[1], Yingming Zeng[2], Shaojing Fu[1,3](✉), and Yuchuan Luo[1]

[1] College of Computer, National University of Defense Technology, Changsha, China
fushaojing@nudt.edu.cn
[2] Beijing Institute of Computer Technology and Applications, Beijing, China
[3] State Key Laboratory of Cryptology, Beijing, China

Abstract. It is effective for supervisors to monitor the network by analyzing traffic from devices.In this way, illegal video can be detected when it is played on the network. Most Internet traffic is encrypted, which brings difficulties to traffic analysis. However, many researches suggest that even if the video traffic is encrypted, the information of video segmentation leaked by DASH (Dynamic Adaptive Streaming over HTTP) can also be used to identify the content of encrypted video traffic without decryption. Moreover, each encrypted video stream can be represented by a fragment sequence. This paper presents two methods based on Levenshtein distance for encrypted video traffic analysis. Using the distance distribution fitted by gamma distribution functions, we calculated a threshold to determine whether two encrypted video traffic belonging to the same video. The accuracy of the judgment using the threshold reached 89%, stably. As far as I am concerned, it is the first work to apply unsupervised methods for content analysis of encrypted video traffic.

Keywords: Encrypted traffic · Levenshtein distance · Threshold · DASH.

1 Introduction

As the continuous development of network technology, there are millions of Internet video viewers online every day. More than half of Internet traffic will be video traffic nowadays. According to the survey, the proportion of video traffic has grown to 80% in 2019. In general, the video traffic is expected to increase 135 exabytes per month, approximately [7]. It is expected that more than 82% of Internet traffic will come from videos by 2022.

Dynamic Adaptive Streaming over HTTP (DASH) is used by most of the video streaming web sites, such as YouTube. DASH is a streaming method, designed to improve the quality of experience [10]. It uses HTTP for video transmission. DASH server divides each video into several short segments (typically a few seconds long), and encodes each segment with a different quality

Y. Xiang et al. (Eds.): SocialSec 2020, CCIS 1298, pp. 97–108, 2020.
https://doi.org/10.1007/978-981-15-9031-3_9

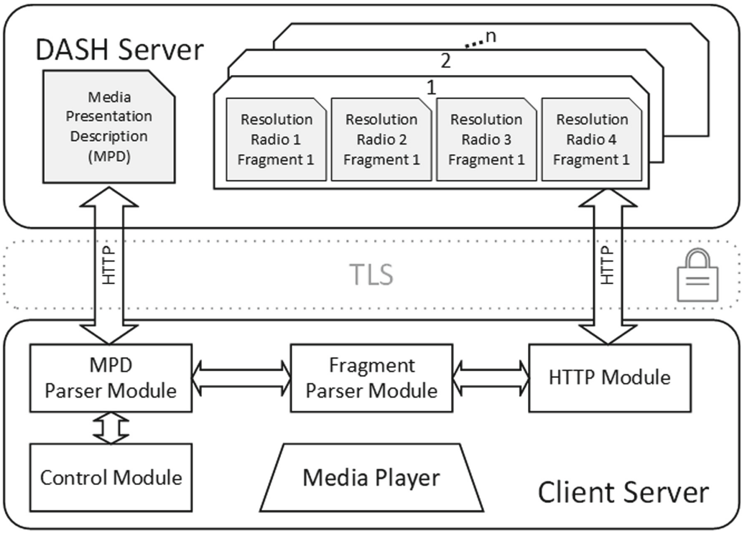
DASH Server
Media Presentation Description (MPD)
...n
2
1
Resolution Radio 1 Fragment 1
Resolution Radio 2 Fragment 1
Resolution Radio 3 Fragment 1
Resolution Radio 4 Fragment 1
HTTP
TLS
HTTP
MPD Parser Module
Fragment Parser Module
HTTP Module
Control Module
Media Player
Client Server

Fig. 1. Dynamic Adaptive Streaming over HTTP

representation level. A media presentation description (MPD) describes segment information, and will be transmitted firstly when transmission of a video is started. According to the network condition and client preferences, adaptive video segments will be transmitted by the DASH server later.

Transport Layer Security (TLS) is widely applied to protect content confidentiality, and adopted by almost all famous video sites. Consequently, traditional Deep Packet Inspection (DPI) [6] methods over plain network traffic do not work here. However, it does not mean that it is not possible to analyze the content of encrypted video traffic. DASH video is always streamed in segment-sized chunks, and it is typically segmented at the application layer [16]. Even though the stream is encrypted between the transport layer and the application layer (e.g.., using TLS), the sizes of segments are visible for network monitor. In a steady encrypted video stream, fragment sizes are correlated with the original segment sizes due to the variable-rate encoding.

Due to the difficulty of decryption, how to analyze encrypted traffic without decryption is a worthy studying direction. Deep learning has been used to analyze encrypted traffic in many works [1,18]. Background traffic and unencrypted part of the encrypted traffic mentioned in [3–5], which are useful for normal encrypted traffic, are insignificant for content identification of encrypted video traffic. Video traffic is usually long-session with a large amount of information transmitted. What is more, most existing encryption traffic analysis methods are based on supervised learning but are not functionally faced with unlabeled traffic data. In real life, the traffic data is basically unlabeled, which makes these existing methods impractical. How to analyze encrypted video traffic without prior knowledge is a problem that should be solved quickly.

In this paper, we proposed a new method based on sequence similarity for encrypted video traffic analysis. A similarity threshold was selected to determine whether two unknown encrypted video streams belong to the same video title. As far as we knew, it is the first work to apply unsupervised methods for content analysis of encrypted video traffic.

The paper's main contributions are:

1. This is the first work that used unsupervised methods to analyze encrypted video traffic. We proposed to measure the similarity of encrypted video streams using Levenshtein distance, innovatively. On this basis, we present an unsupervised methods (threshold) that are applicable to analyze the content of encrypted video traffic.
2. A threshold was computed using a Gamma distribution fitting to determine whether two unknown video streams belong to the same video title, and have achieved an acceptable probability of correct judgment.
3. We run through a set of experiments to prove the possibility and robustness of the threshold we computed.

The remainder of this paper is organized as follows. In Sect. 2 we review related work. In Sect. 3, we introduction the preliminaries - TLS protocol and Levenshtein distance. In Sect. 4 we introduced the generation of fragment sequence and two analysis methods. In Sect. 5 we introduced the dataset used in this paper. In Sect. 6 we computed a threshold to determine whether two unknown video streams belong to the same video title. Finally, we conclude in Sect. 7.

2 Related Work

Many works have suggested methods for encrypted traffic identification. Several works have examined different features.

Liu et al. [12] presented a method for video title classification of RTP/UDP traffic. Liu et al. [13] used the wavelet transform for constructing unique and robust video signatures with different compactnesses. Ashwin Rao et al. [15] showed that the streaming strategies vary with the type of the application, and the type of container used for video streaming by studying the network characteristics of Netflix and YouTube. Pablo Ameigeiras et al. [2] presented a characterization of the traffic generated by YouTube when accessed from a regular PC, and proposed a YouTube server traffic generation model. However, there are several changes in video traffic over the Internet. They do not fit modern streaming traffic as previous solutions operated on a time series with the granularity of single video frames [10].

In recent years, some work used machine learning algorithms for the identification of encrypted video traffic. Algorithms using custom KNN and SVM were presented by Ran Dubin [10] for encrypted HTTP adaptive video streaming title classification. Roei Schuster [16] showed that many video streams are uniquely characterized by their fragment patterns, and classifiers based on convolutional neural networks can accurately identify these patterns given very coarse network measurements. Yan Shi et al. [17] proposed a key idea to examine encrypted and tunneled video streaming traffic at a Soft-Margin Firewall (SMFW), which was located near the streaming client in order to identify undesirable traffic sources and to block or throttle traffic from such sources. These works showed that the content classification of encrypted video stream is possible although the content is not visible.

Even if some encouraging progress has been made, the timing characteristics of encrypted video stream have been ignored in these works, which contains valuable information. Moreover, the methods based on supervised learning are powerless when faced with unlabeled encrypted video traffic on the Internet. In view of this, we tried to use unsupervised learning to analyze encrypted video traffic, which requires no labels. As far as I am concerned, it is the first work using unsupervised methods to analyze the content of encrypted video traffic.

3 Preliminaries

3.1 TLS Protocol

Transport Layer Security (TLS) is cryptographic protocol that provides secure communication between two parties over the Internet by encapsulating and encrypting application layer data. It is used by most of the video sites in order to encrypt the network traffic. The TLS protocol is between application layer and transport layer, and it is application protocol independent [8].

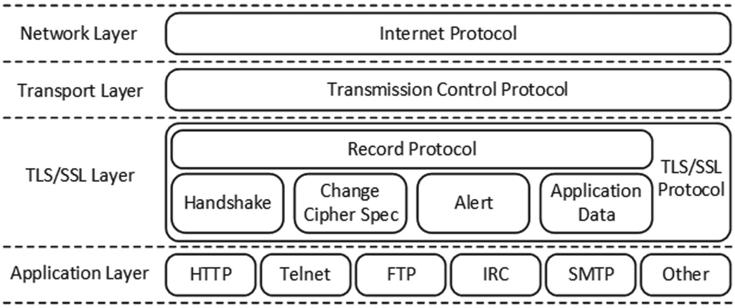

Fig. 2. Protocol layers

The TLS includes two protocol layers (as Fig. 2). The Record Protocol takes messages to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, and transmits the result [8]. Received data is decrypted, verified, decompressed, reassembled, and then delivered to higher-level clients . There are four protocols that use the record protocol, including the application data protocol. Application data message is carried by the record layer and are fragmented, compressed, and encrypted based on the current connection state.

3.2 Levenshtein Distance

The Levenshtein distance is a string metric for measuring the difference between two sequences. It is named after the Soviet mathematician Vladimir Levenshtein, who considered this distance in 1965 [11]. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutiond) required to change one word into other. It may also be rederred to as edit distance [14]. The Levenshtein distance between two string a, b (of length $|a|$ and $|b|$ respectively) is given by $lev_{a,b}(|a|, |b|)$ as follow:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & , if\ \min(i,j) = 0 \\ \min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{a_i \neq b_j} \end{cases} & , otherwise \end{cases} \quad (1)$$

where $1_{a_i \neq b_j}$ is the indicator function equal to 1 when $a_i \neq b_j$ and equal to 0 otherwise, and $lev_{a,b}(i,j)$ is the distance between the first i characters of a and the first j characters of b. i and j are 1-based indices.

For example, the Levenshtein distance between “kitten” and “sitting” is 3 because of the following edits change:

1. **k**itten → **s**itten
2. sitt**e**n → sitt**i**n
3. sittin → sittin**g**

The upper bounds of the Levenshtein distance is the length of the longer string. The Levenshtein distance is zero only if the strings are equal.

4 Methodology

Video traffic has some unique characteristics. Video sessions are usually long sessions with a large amount of information transmitted, while the amount of information transmission of non-video traffic is small relatively. Regarding the information leakage in terms of timing, the timing information is leaked due to the long duration of video traffic, relatively. Therefore, we focus on timing information of encrypted video traffic in order to analyze.

Levenshtein distance can compare the similarity of sequences with different length, which is suitable for video streams. Besides, Levenshtein distance is simple and effective. Consequently, it is chosen to measure the similarity of video streams. We have developed two methods based on Levenshtein distance for identifying encrypted video traffic. Before this, packet reorganization technique was applied to generate the fragment sequence of encrypted video traffic. We used the normalized Levenshtein distance to the content similarity of unknown video streams, and calculated a threshold to determine whether two streams belong to the same video by Gamma distribution fitting.

4.1 Fragment Sequence of Encrypted Video Traffic

Application data traffic accounts for most of the total encrypted traffic, especially for video traffic. Video compression and encoding algorithms cause that different video scenes contain different amounts of perceptually meaningful information. The meaningful information refers to the size of video fragments. Because of application protocol independence of TLS, this meaningful information is retained although the content of the message is encrypted.

In order to parse a TLS stream, first of all, we should reassemble the video traffic packets to TCP flow according to the TCP protocol. After that, we parse the TCP flow according to the TLS protocol. When packets reassembly and TLS parsing is completed, the TLS session exhibits a request-response pattern similar to HTTP interactive. A transaction between client and server in the TLS session, the payload sent by the server contains more than one application data. Consequently, fragment refers to the number of application data in encrypted video traffic.

After parsing the TLS stream, we can get a sequence representing, which we called Fragment Sequence, the number of the application data sent by the server per HTTP interactive. Because the encode in DASH is variable bitrate (VBR), the size of video fragments is related to the content complexity of video fragment. If the content of a video fragment is complex, the fragment size is large. Otherwise, the size of fragment is small when the content of a video fragment is simple. For example, fragment sequences of three video titles are listed as follows. We find it that videos of the same title have similar fragment sequences.

Cheerleader

seq1:

30-23-73-22-124-25-124-130-23-123-103-23-92-130-23-118-130-24-130-91-12

seq2:

30-23-73-22-124-25-124-130-23-116-130-23-123-103-23-92-130-23-118-130-24-130-91

seq3:

30-23-73-113-124-124-23-130-123-23-103-92-23-130-118-24-130-130-23-91

Fast_and_Furious_six

seq1: 14-9-44-90-16-145-2-221-21-21-253-20-21-7-214-20-258-27-39

seq2: 14-9-44-21-145-2-221-21-21-253-20-9-12-7-214-20-258-27-39

seq3: 14-9-44-21-145-2-221-21-21-253-20-21-221-20-258-27-39

Wo_sind_die_Clowns

seq1: 5-53-11-25-28-2-244-241-26-77-17

seq2: 5-53-25-27-2-244-241-26-77-17

seq3: 5-53-25-28-2-244-241-26-77-17

4.2 Threshold Selection

Because Levenshtein distance is affected by the length of sequences, it is necessary to normalize. The definition of normalized Levenshtein distance is as follows:

$$Normalized\_LD(a,b) = \frac{LD(a,b)}{\max\{|a|,|b|\}} \tag{2}$$

We used normalized Levenshtein distance of Fragment sequence to measure the similarity of two unknown encrypted video streams and determined whether the two streams belonging to the same video. To determine whether two unknown video streams belong to the same video, we need to set a threshold for normalized Levenshtein distance. If the distance is greater than the threshold, it is determined that two video streams belong to the different video. On the contrary, two video streams belong to the same video if the distance is less than the threshold.

The normalized Levenshtein distance of video streams is a random variable. The random variable X_1 and X_2 indicate the normalized Levenshtein distance of two video streams belonging to the same video and different video, respectively. After the analysis of samples, the results demonstrate that X_1 and X_2 conform to Gamma distribution. The parameters α and β of the gamma distribution can be calculated from the mean μ and variance σ^2 of the data.

$$X_1 \sim Ga(\alpha_1, \beta_1^2), \quad X_2 \sim Ga(\alpha_2, \beta_2^2) \qquad \alpha = \frac{\mu^2}{\sigma^2}, \quad \beta = \frac{\mu}{\sigma^2} \tag{3}$$

The probability density function were as follow:

$$f_1(x) = \frac{\beta_1^{\alpha_1}}{\Gamma(\alpha_1)} x^{\alpha_1 - 1} e^{-\beta_1 x}, \quad f_2(x) = \frac{\beta_2^{\alpha_2}}{\Gamma(\alpha_2)} x^{\alpha_2 - 1} e^{-\beta_2 x} \tag{4}$$

We set their distribution functions as $F_1(x) = \int_0^x f_1(x)\mathrm{d}x$ and $F_2(x) = \int_0^x f_2(x)\mathrm{d}x$. Let the sum of the correct judgment probabilities be P, which is defined as follow:

$$P = \int_0^x f_1(x)\mathrm{d}x + \int_x^1 f_2(x)\mathrm{d} = F_1(x) + F_2(1) - F_2(x) \tag{5}$$

One way to get the minimum of P is to take a derivative with respect to x, and look for the derivative being zero. The derivation process is as follow:

$$\frac{\mathrm{d}P}{\mathrm{d}x} = F_1'(x) - F_2'(x) = f_1(x) - f_2(x) \tag{6}$$

When $f_1(x) = f_2(x)$, the sum of the correct judgment probabilities is the smallest. Simplify the equation $f_1(x) = f_2(x)$ are as follows:

$$e^{(\beta_2 - \beta_1)x} = \frac{\beta_2^{\alpha_2}\Gamma(\alpha_1)}{\beta_1^{\alpha_1}\Gamma(\alpha_2)} x^{\alpha_2 - \alpha_1} \tag{7}$$

The equation is transcendental, so it does not have an analytical solution. We can use numerical analysis methods, like bisection method and Newton's method, to find the numerical solutions of this transcendental equation.

The threshold is one of the solutions.

5 Data

In this paper, we use the public dataset in [9]. It contains 10,000 YouTube streams of 100 video titles (100 streams per title). The video streams were collected via a real-world Internet connection over different real-world network conditions. They were collected by crawler using the Selenium web automation tool with ChromeDriver. The video titles in dataset are popular YouTube video from different categories such as sports, news, nature, etc. The traffic of the dataset were collected using Chrome browser because of it's popularity.

6 Threshold Experimental Evaluation

In this section, we calculated the threshold for judging the homology of video traffic, and used accuracy to evaluate its performance.

6.1 Metrics

Before the experiment, we should define two metrics of experiments, including Theoretical Accuracy (TA) and Real Accuracy (RA).

- **Theoretical Accuracy**: The normalized Levenshtein distance of Encrypted video traffic between the same video and different video subject to be Gamma distribution. Therefore, we are able to calculate the theoretical accuracy using the gamma distribution and the threshold x_0. Theoretical accuracy is calculated as follow:

$$TA = \frac{\int_{x_0}^{1} f_1(x)\mathrm{d}x + \int_{0}^{x_0} f_2(x)\mathrm{d}x}{\int_{0}^{1} f_1(x)\mathrm{d}x + \int_{0}^{1} f_2(x)\mathrm{d}x} \tag{8}$$

- **Real Accuracy**: Real accuracy is calculated using the statistical result of the experiment. The threshold x_0 is used to judge positive samples and negative samples. The real accuracy is as follow:

$$RA = \frac{TP + TN}{TP + FP + FN + TN} \tag{9}$$

6.2 Threshold Experimental

We extracted two encrypted video streams with the same title from the dataset, and calculated their similarity. We also extracted two encrypted video streams with different titles and calculated their similarity. Both operations were performed 10'000 times, and we get two sets of data about the video streams similarity (of the same title and of different titles).

As can be seen from the distributions in Fig. 3 and Table 1, there are two normal distributions with different mean and nearly the same variance ($X_1 \sim Ga(3.53, 17.48)$ and $X_2 \sim Ga(17.50, 39.46)$). It showed that the assumptions are reasonable. According to the Eq. 7, we computed the threshold to be 0.30167, and used it to test the accuracy. The experimental result indicated that the threshold we computed can distinguish whether the two streams belonging to the same video title with 89.00% probability. The theoretical accuracy is 88.03%. The error of theoretical accuracy is less than 1%. The experimental result is acceptable.

Table 1. Parameter of the gammma distribution

Variables	μ	σ	α	β
same (X_1)	0.2019	0.1075	3.5289	17.4815
diff (X_2)	0.4434	0.1060	17.4970	39.4571

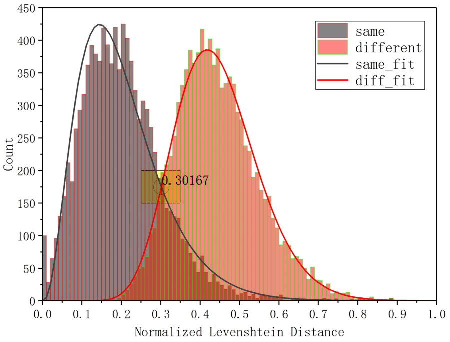

Fig. 3. Data statistics and gamma distribution fitting effect.

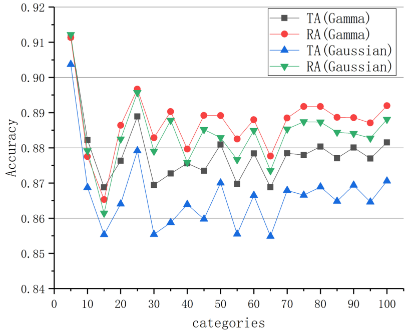

Fig. 4. The accuracy comparison of the fitting effect of gaussian distribution and gamma distribution.

6.3 Stability Test and Comparison of Different Fitting Methods

In order to test the stability of threshold and accuracy, we performed experiments on datasets with different number of categories. In each dataset, we selected 10'000 distance data for the same video and 10'000 distance data for different videos. The box-plot of datasets distribution of with different category number is depicted in Fig. 5. On the whole, it showed that the distance distribution of encrypted video stream does not vary with the category number of datasets. More importantly, Fig. 5 illustrates that the threshold we calculated has high generalization performance.

On this basis, we also compared the effect of two fitting methods, Gaussian distribution and Gamma distribution. The theoretical accuracy and real accuracy are shown in Fig. 4 and Table 2. As evident from figure, generally

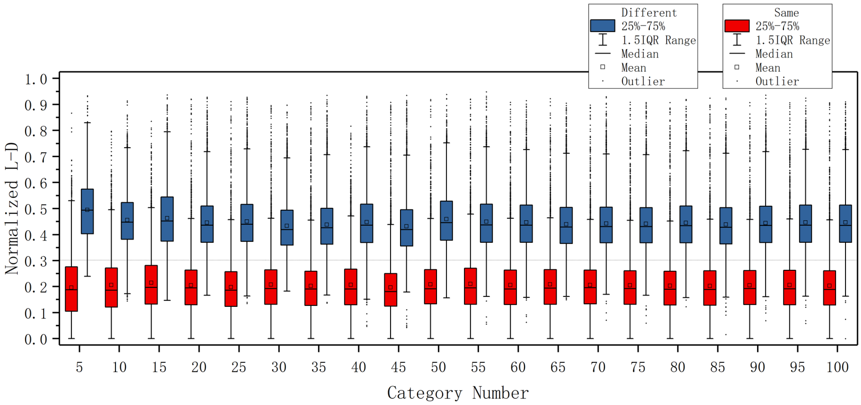

Fig. 5. Box-plot of data distribution for different datasets.

Table 2. The comparison of the fitting effect of gaussian distribution and gamma distribution on different datasets.

Categories	Gamma distribution				Gaussian distribution			
	Threshold	TA(%)	RA(%)	Error(%)	Threshold	TA(%)	RA(%)	Error(%)
5	0.33068	91.158	91.135	0.023	0.34833	90.371	91.218	0.847
10	0.31046	88.221	87.752	0.470	0.33210	86.871	87.916	1.045
15	0.31326	86.876	86.532	0.344	0.33863	85.534	86.142	0.608
20	0.30310	87.633	88.639	1.006	0.32538	86.406	88.247	1.840
25	0.30353	88.891	89.668	0.777	0.32379	87.918	89.570	1.652
30	0.29889	86.943	88.288	1.345	0.32072	85.540	87.905	2.365
35	0.29682	87.269	89.030	1.762	0.31988	85.875	88.781	2.906
40	0.30441	87.561	87.965	0.404	0.32675	86.394	87.588	1.195
45	0.29119	87.348	88.922	1.574	0.31433	85.973	88.521	2.548
50	0.31144	88.094	88.914	0.820	0.33334	87.001	88.290	1.289
55	0.30567	86.975	88.249	1.274	0.32960	85.550	87.666	2.115
60	0.30517	87.838	88.800	0.962	0.32663	86.652	88.491	1.840
65	0.30097	86.882	87.764	0.882	0.32381	85.488	87.354	1.865
70	0.30325	87.846	88.847	1.002	0.32396	86.790	88.534	1.744
75	0.30246	87.794	89.173	1.378	0.32335	86.652	88.743	2.091
80	0.30269	88.033	89.173	1.140	0.32397	86.890	88.733	1.843
85	0.29901	87.707	88.864	1.157	0.32062	86.484	88.444	1.960
90	0.30382	88.007	88.857	0.850	0.32450	86.940	88.407	1.467
95	0.30444	87.697	88.706	1.008	0.32638	86.462	88.276	1.814
100	0.30370	88.152	89.196	1.044	0.32472	87.056	88.811	1.756
Average	0.30475	87.846	88.724	0.961	0.32654	86.642	88.382	1.739

speaking, the Gamma distribution fitting performs better than Gaussian distribution fitting (about 1.0% performance improvement). The theoretical accuracy (TA) error of the Gamma distribution fitting is 0.35% approximately, which is better than the Gaussian distribution fitting (the error is about 1.7%). Besides, Fig. 4 also illustrates the performance of our method is steady, and the average accuracy is at about 89%.

7 Conclusion

Although many well-known video sites such as YouTube uses HTTPS, which is considered to protect user privacy, it still leaks content information of videos. In this paper, we showed that the Levenshtein distance of fragment sequences are able to assess the content similarity of encrypted video streams. We demonstrated it possible to analyse the content of encrypted video traffic with Unsupervised method. It is effective to analyze the encrypted video traffic when there is lacking the apriority knowledge.

First, a threshold was calculated by fitting with the Gamma distribution function. Our statistical analysis concluded that the threshold can determine whether two unknown video streams belong to the same video title with a probability of over 89%. Moreover, we illustrated the stability of the threshold and its judgment accuracy through further experiments. In another work, we also implemented the clustering of encrypted video streams using spectral clustering based on Levenshtein distance and achieved a good result.

Regardless, we can foresee a bright future for encrypted video stream analysis based on sequence similarity. It is necessary for unsupervised learning of encrypted video traffic. Future researchers should consider a new definition of sequence distance, which reflects the similarity of video content better.

Acknowledgments. This work is supported by the National Key Research and Development Program of China (No. 2018YFB0204301), Open Foundation of State Key Laboratory of Cryptology (No: MMKFKT201617), and the NUDT Research Grants (No. ZK19-38).

References

1. Aceto, G., Ciuonzo, D., Montieri, A., Pescapé, A.: Mobile encrypted traffic classification using deep learning: experimental evaluation, lessons learned, and challenges. IEEE Trans. Netw. Service Manag. **16**(2), 445–458 (2019)
2. Ameigeiras, P., Ramos-Munoz, J.J., Navarro-Ortiz, J., Lopez-Soler, J.M.: Analysis and modelling of youtube traffic. Trans. Emerg. Telecommun. Technol. **23**(4), 360–377 (2012)
3. Anderson, B., McGrew, D.: Identifying encrypted malware traffic with contextual flow data. In: Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security. pp. 35–46 (2016)
4. Anderson, B., Paul, S., McGrew, D.: Deciphering malware's use of tls (without decryption). J. Comput. Virol. Hacking Tech. **14**(3), 195–211 (2018)
5. Bagaria, S., Balaji, R., Bindhumadhava, B.: Detecting malignant tls servers using machine learning techniques. arXiv preprint arXiv:1705.09044 (2017)
6. Bujlow, T., Carela-Español, V., Barlet-Ros, P.: Independent comparison of popular dpi tools for traffic classification. Comput. Netw. **76**, 75–89 (2015)

7. Cisco, C.V.N.I.: The zettabyte era-trends and analysis, 2015–2020. white paper (2016)
8. Dierks, T., Rescorla, E.: Rfc 5246-the transport layer security (tls) protocol version 1.2. Internet Engineering Task Force (2008)
9. Dubin, R., Dvir, A., Pele, O., Hadar, O.: The video streams pcap files dataset. http://www.cse.bug.ac.il/title_fingerprinting/ (2017)
10. Dubin, R., Dvir, A., Pele, O., Hadar, O.: I know what you saw last minute-encrypted http adaptive video streaming title classification. IEEE Trans. Inf. Forensics Secur. **12**(12), 3039–3049 (2017)
11. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710 (1966)
12. Liu, Y., Ou, C., Li, Z., Corbett, C., Mukherjee, B., Ghosal, D.: Wavelet-based traffic analysis for identifying video streams over broadband networks. In: IEEE GLOBECOM 2008–2008 IEEE Global Telecommunications Conference. pp. 1–6. IEEE (2008)
13. Liu, Y., Sadeghi, A.-R., Ghosal, D., Mukherjee, B.: Video streaming forensic – content identification with traffic snooping. In: Burmester, M., Tsudik, G., Magliveras, S., Ilić, I. (eds.) ISC 2010. LNCS, vol. 6531, pp. 129–135. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-18178-8_11
14. Navarro, G.: A guided tour to approximate string matching. ACM Comput. Surv. **33**(1), 31–88 (2001)
15. Rao, A., Legout, A., Lim, Y.s., Towsley, D., Barakat, C., Dabbous, W.: Network characteristics of video streaming traffic. In: Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies. pp. 1–12 (2011)
16. Schuster, R., Shmatikov, V., Tromer, E.: Beauty and the burst: remote identification of encrypted video streams. In: 26th {USENIX} Security Symposium ({USENIX} Security 17). pp. 1357–1374 (2017)
17. Shi, Y., Ross, A., Biswas, S.: Source identification of encrypted video traffic in the presence of heterogeneous network traffic. Comput. Commun. **129**, 101–110 (2018)
18. Yao, H., Liu, C., Zhang, P., Wu, S., Jiang, C., Yu, S.: Identification of encrypted traffic through attention mechanism based long short term memory. IEEE Trans. Big Data (2019)