

Research Article

Noise-Resistant Video Channel Identification

Mingkai Wang¹, Zengkun Xie², Xiangdong Tang¹, and Fei Chen¹

¹College of Computer Science & Technology, Qingdao University, Qingdao, China

²Department of electrical and new energy engineering, Yantai Engineering & Technology College, Yantai, China

Correspondence should be addressed to Fei Chen; feic@qdu.edu.cn

Received 30 January 2022; Accepted 22 June 2022; Published 18 August 2022

Academic Editor: Zengpeng Li

Copyright © 2022 Mingkai Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As the video streaming traffic grows exponentially nowadays, variable bitrate (VBR) encoding has been widely utilized by modern live video streaming service providers, such as YouTube, TikTok, and Twitch. However, video bitrate can be a delicate fingerprint of the video streaming, leading to risks of privacy leakage. There are several studies that attempt to eavesdrop the privacy from encrypted video streaming, but most of them presume strict requirements on the implementation environments and have great limitations when noise interference exists. Actually, the video traffic from the multimedia edge server is distinct from inter-application traffic flows due to device customization and can be identified even if there are noise interferences or the victim in a weak network condition. In this paper, a video traffic identification method is proposed to identify the encrypted video streaming from multimedia edge server under the interference of irrelevant traffic flows. Initially, we use an interapplication filter to identify the traffic from the edge server. Then, a longest-common-subsequence (LCS)-based method is developed for similarity matching to resist the noise interference from unpredictable burst traffic and network environment variations. In order to evaluate the system performance, we setup the prototype system with an AWS EC2 server and a raspberry pi device, then utilize the real-world trace data for pushing movies to victims. The experimental results show that the accuracy of our proposed strategy can reach 89.1% within 140 seconds eavesdropping even mixed with 14% noise interference.

1. Introduction

With the improvement of the network bandwidth, the video streaming service has been popular in recent years, which quickly sweeps across the world and takes up the viewers' free time by high-quality content in live e-commerce, sports events, or video games. For example, according to the report of Statista, which is a global business data platform, shows that the number of monthly active users of TikTok worldwide has exceeded 1 billion [1]. Meanwhile, the number of monthly active users of YouTube has exceeded 2.3 billion. However, the growing number of users has brought great bandwidth pressure to video data center. Thanks to the development of edge computing in recent years, more and more Internet service providers try to save server resources and reduce the round-trip time by handing user tasks to edge servers, such as computation offloading [2] and video delivery [3]. In the foreseeable future, more and more applications will be handled by edge servers with the performance improvement of edge devices and popularity of 5G infrastructure.

Conventionally, the bitrate-based fingerprint carried by video traffic flow can be identified by video traffic pattern analysis even with the transportation layer security (e.g., TLS) encryption. There are many studies attempting to eavesdrop the content of videos from viewers which are under TLS encryption in recent years [4–6], but most of these works assume that the encrypted video stream can be directly observed by attackers without interference of irrelevant traffic flows. Some studies also proposed noise-resistant fingerprint identification methods, but all of them are not suitable for video bitrate fingerprints [7]. Actually, the video traffic is usually delivered from content delivery network (CDN) which may serve multiple websites or applications at the same time [8]. Therefore, the complete and noiseless bitrate-based traffic fingerprint can be hardly identified from the real-world trace data. Furthermore, the effectiveness of traffic fingerprints is highly sensitive to network fluctuations, and the partial features of traffic fingerprint will drift seriously during unstable network conditions.

The prevalence of edge server brings a new risk to video traffic identification due to the customization of the edge devices. Conventionally, the CDN server usually undertakes several tasks including video delivery and static resource delivery using the same domain name, which will make it difficult to identify the video traffic flow encrypted by TLS. However, the multimedia edge server hardly delivers the irrelevant traffic due to the customization of the edge device, which leads to the possibility of identifying the bitrate-based traffic flows from it. Therefore, the video traffic from the edge server is easier to identify and the traffic features are more stable. In this paper, we will present a noise-resistant video traffic identification method for VBR traffic flow. We will show that the traffic fingerprint from the real-world trace data captured from multimedia edge server can also match the bitrate fingerprint after appropriate preprocess. Initially, a simple traffic filter which only uses three labels from the unencrypted traffic is used to filter out the traffic that is from the multimedia edge server. After that, an LCS-based fingerprint-matching method is proposed to eliminate the interference of the remaining two types of noise and match the traffic fingerprint and bitrate fingerprint.

The rest of this paper is organized as follows: The literature is explored in Section 2. The data analysis is presented in Section 3. The system design is presented in Section 4. The traffic filter and LCS-based matching method are illustrated in Section 5 and Section 6. The system performance is evaluated in Section 7. Finally, Section 8 concludes this paper.

2. Related Work

2.1. Privacy Leakage and Protection. With the growth of Internet applications, new security issues arise with the development of Internet infrastructures. On the one hand, the new paradigms could bring facilities to our daily life such as recommendation system [9, 10], computation offloading [2, 11, 12], and route planning [13, 14]. On the other hand, the privacy defense strategy also needs to consider more aspects with the upgrading of infrastructure: mobile devices [15], Internet of things (IoT) device [16–18], and cloud server [14, 19]. Specifically, machine learning [20] and edge computing are developed rapidly, which brings more complex privacy leakage problems [21]. With the improvement of bandwidth and device performance, more video streaming service providers use edge servers to cache and distribute video content in order to reduce the pressure of data center, which leads to the popularity of research of multimedia privacy protection on edge server [22, 23]. In this paper, we will discuss the privacy leakage caused by encrypted video under noise interference.

2.2. Privacy Leakage from Video Stream. The side channel attack caused by privacy leakage of encrypted video has attracted extensive attention in recent years. Saponas et al. [4] makes fingerprints by using multiple sliding windows to divide the video into segments of several milliseconds based on VBR encoded video, but they only achieve 62% accuracy

with 10 minutes eavesdropping without noise interference. Gu et al. [24] improved the DTW algorithm to make it suitable for DASH protocol and made a classifier that can identify videos from both Netflix and YouTube, but they claim that the low bandwidth and high packet loss rate are not in their consideration since users will normally leave video streaming immediately because of the bad experience. As the prevalence of machine learning, neural network has an advantage of feature extraction in a sophisticated environment. Schuster et al. [5] modeled the fingerprints and proposed a CNN-based model to identify the fingerprints for VBR-based videos from YouTube and Netflix. Nevertheless, all the bitrate-based video identification strategies need the assumption of stable network. Otherwise, both weak network condition and burst traffic will have a serious interference on traffic fingerprint, which will inevitably lead to wrong identification results because the points in bitrate fingerprint will be matched incorrectly. In the following part, we will analyze the noise interference and then propose a noise-resistant video traffic identification method.

2.3. Sequence Matching Method. Sequence matching methods are essential in solving many pattern recognition problems such as anomaly detection, speech recognition, and other domains [25]. The popular methods usually consider using points for matching (e.g., Edit Distance on Real Sequence (EDR) [26], Dynamic Time Warping (DTW) [27]), using shape for matching (e.g., Frechet distance [28]), and segmenting the sequence for matching (e.g., One Way Distance [29]). Nevertheless, most sequence matching methods do not consider the matching effectiveness in interference environment. Thanks to the powerful representation ability of deep learning, similarity learning can accommodate heterogeneous features in the sophisticated environments, and there are several deep-learning-based methods like the CNN-based solution [30, 31], and the LSTM-based solution [32]. However, deep-learning-based models usually need online training to adapt the latest features, and the computational cost is very high.

3. Traffic Data Analysis

In this section, we will introduce the video data analysis to illustrate the video bitrate and several types of traffic noise using the classic movie *Titanic*. In the following parts, *nginx* and *ffmpeg* is used to push the encrypted video traffic, *Chrome downloader* is used to provide the irrelevant traffic, and *wondershaper* is used to simulate the weak network environment with the random interference of bandwidth limitation, RTT, and packet loss.

3.1. Side Channel Attack on Video Traffic. VBR can bring the risk of privacy leakage through the bitrate fluctuation. Figure 1 shows the bitrate of a video which encode with constant bitrate (CBR) and VBR, and it can be seen that there are significantly different fluctuation trends between them.

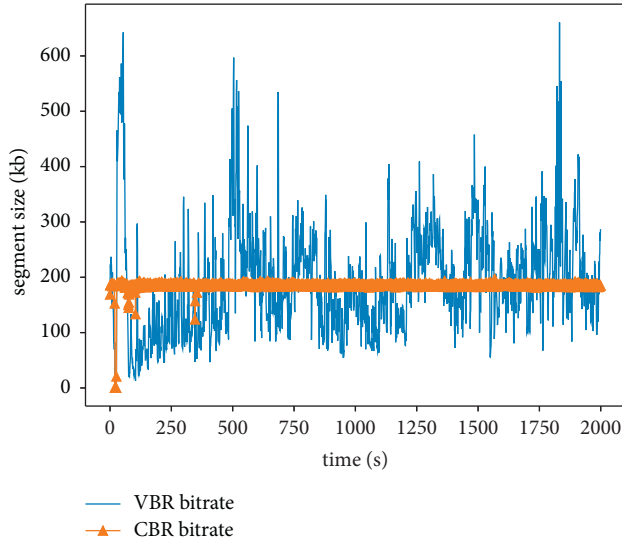


FIGURE 1: A comparison of CBR and VBR bitrate.

Additionally, TLS only encrypts the content, but leak the statistical features of the traffic. Figure 2 shows the correlation between the bitrate of VBR video and it is encrypted video streaming.

Obviously, the privacy of video viewers can be identified through the analysis of the video traffic even after encryption. When the attacker obtains a traffic fingerprint segment, the privacy may be leaked.

3.2. Bitrate Features with Irrelevant Traffic. When providing video streaming services for users, edge devices can also provide other multimedia services from different websites at the same time (such as encode offloading or download acceleration), resulting in the eavesdropped traffic containing multiple types of packets, which make it difficult to identify the video traffic. Figure 3 shows the traffic from a raspberry edge server, which contains only video stream and both video stream and download stream.

It can be seen that the video stream traffic is covered by mixed traffic, resulting in the disappearance of the video traffic features.

3.3. Bitrate Features in the Weak Network Condition. VBR features are usually easy to identify, which is more likely to lead to privacy leakage. However, such features are easily affected by noise or weak network condition, which reduces the accuracy of identification. Figure 4 shows the interference of bandwidth limitation and RTT on the traffic fingerprint of *Pirates of the Caribbean 5* from 1000 seconds to 1700 seconds. The video traffic is collected from raspberry edge server.

Since the beginning of traffic eavesdropping, the bandwidth limitation from 50 to 120 second and the burst RTT from 170 to 180 second lead to video playback jitter and corresponding backward drift of traffic features. Figure 5 adds the interference of 15% random packet loss to the traffic fingerprint of *Pirates of the Caribbean 5* from 2000 seconds

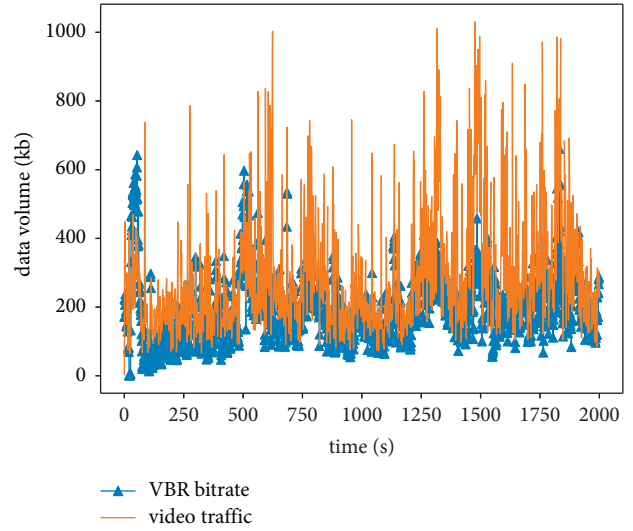


FIGURE 2: Video bitrate and traffic.

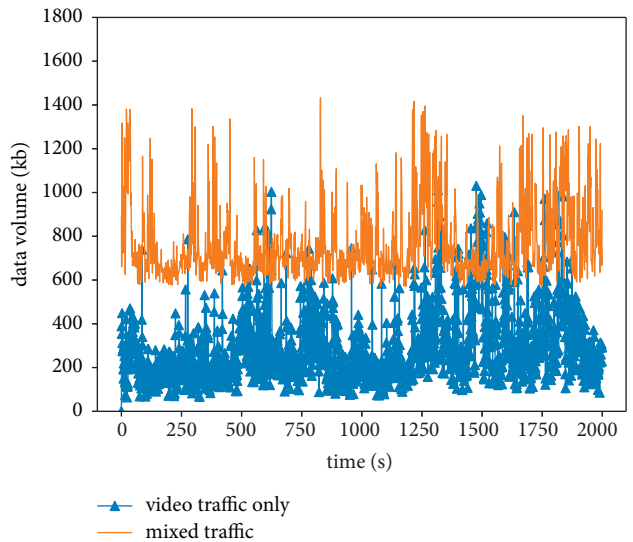


FIGURE 3: Video traffic and mixed traffic.

to 2200 seconds. Due to the packet retransmission function of TCP protocol, the interference of feature drift is reduced, but it still reduces the matching accuracy between bitrate fingerprint and traffic fingerprint. In a word, the bitrate-based video fingerprints raise stringent requirements on network conditions.

3.4. Bitrate Features with Intra-Application Interference. Even in the same application, the features will also be significantly affected by user operations, which usually cannot be predicted. Whether viewers explore the video list while watching or communicating through the intrasite chat system, it will have a destructive interference on the traffic features and seriously reduce the identification accuracy. Figure 6 shows the burst traffic by browsing the video list and the interference on the traffic features.

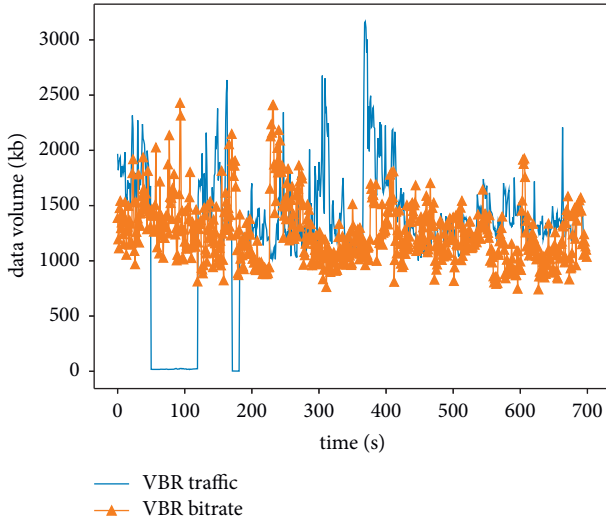


FIGURE 4: Traffic features under bandwidth limitation and RTT.

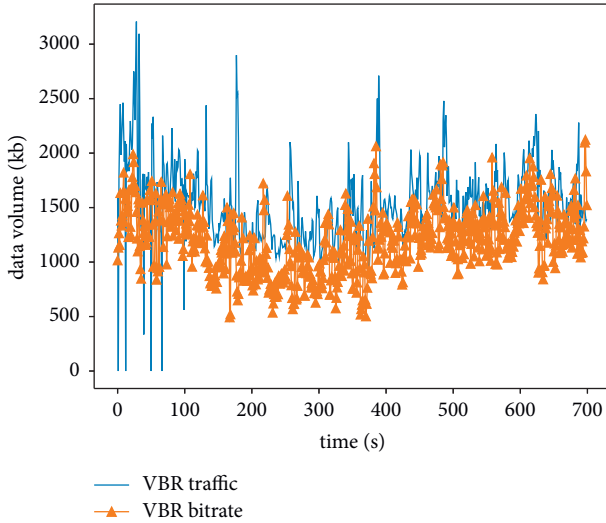


FIGURE 5: Traffic features under packet loss.

Obviously, the traffic generated by unpredictable behavior on 100 s to 110 s completely covers the original traffic features.

4. System Design

In this section, we will present the system design with the noise-resistant encrypted video traffic identification. The system structure is presented in Figure 7. The proposed system can be divided into following parts:

- (i) Interapplication traffic filter: A filter based on three labels including server name indicator (SNI) is proposed to filter out the traffic that from the multimedia edge server.
- (ii) LCS-based fingerprint matching: An LCS-based method is proposed for matching the traffic fingerprint and bitrate fingerprint under noise interferences.

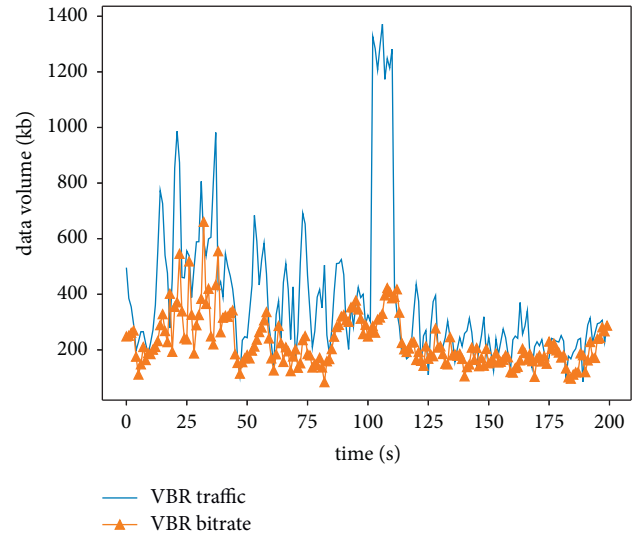


FIGURE 6: Burst traffic caused by user behavior.

The SNI tag is used to bring the domain name requested by the server through a plain text in the handshake stage of the TLS protocol. The attacker can easily obtain the target domain using SNI as an interapplication traffic filter, and further identify the whole TLS session through IP address or sequence number, and then obtain the video traffic flow completely without other interinterference due to the customization of the edge device. It should be noted that all the video providers need to transfer the video stream according to the protocol which specified by the edge multimedia framework, and the edge server will use the unified video protocol to send the video stream to users. As the popular edge multimedia frameworks such as *EasyNVR* or *Link Visual* all use TLS for video delivery, so our filter can be regarded as a general method for the existing video service. However, the traffic fingerprint will still affect by the burst traffic from unpredictable behaviors (such as exploring the video list), or the weak network condition, for example, low bandwidth and packet loss after filtering. So an LCS-based method is proposed to filter the intraapplication interference and identify the matched segments between traffic fingerprint and bitrate fingerprint.

5. Interapplication Traffic Filter

We will propose a traffic filter to eliminate irrelevant traffic from other applications in this section. Three labels are utilized to achieve the traffic filter: SNI, content type, and source IP address (srcaddr):

- (i) SNI is used to filter the video traffic which to be identified.
- (ii) ContentType is used to divide the TLS session.
- (iii) IP address is used to obtain the continuous TLS session.

The video content is sent in stream, but each video segment is encrypted in a TLS session, thus, the session is denoted as a video segment in a fixed length. ContentType is

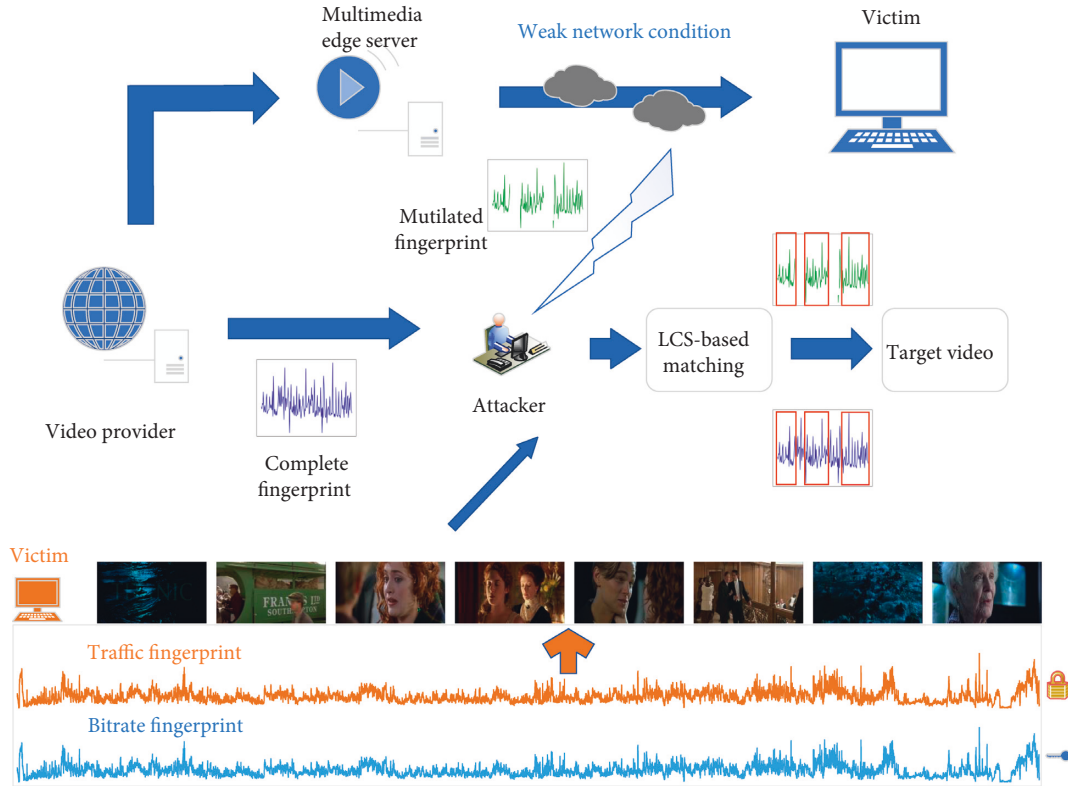


FIGURE 7: An overview of system structure.

used to check whether the packet is a TLS handshake packet (denote the start of a new TLS session). Since the SNI in the handshake packet holds the source domain name without encrypted, all video streaming TLS sessions can be identified. The filtering process is shown in Algorithm 1.

After filtering, we get a set $S = \{s_0, s_1 \dots s_j\}$ containing j packets in all TLS sections, where s_j is a two-tuple $\langle \text{length}_j, \text{time}_j \rangle$ for the j th packet with t_j as the arrival time and length_j as the packet length.

6. Noise-Resistant Fingerprint Matching

In the previous section, we obtain the packet sequence through filtering the TLS session. However, the intra-application interference still exists and seriously reduces the matching accuracy. In this section, we will propose a noise-resistant similarity matching method based on LCS model. Before performing the matching model between bitrate fingerprint and traffic fingerprint, we should discuss the feature drifting caused by weak network condition and intraapplication noise interference. The bandwidth fluctuation caused by weak network will limit the data obtained by viewers and then destroy the traffic fingerprint. For example, for the same video segment which bitrate fingerprint is (1, 2, 3, 4, 5), the traffic fingerprint eavesdropped from a viewer with stable network is (2, 3, 4, 5, 6), but eavesdropped from another viewer with weak network will become (2, 3, 0, 0, 4, 5, 6), which will seriously reduce the identification accuracy. Similarly, the intraapplication noise will also change the traffic features and reduce the accuracy. For example, the

traffic fingerprint eavesdropped from a viewer without interference is (2, 3, 4, 5, 6), but when there is a burst traffic caused by unpredictable behavior, the traffic fingerprint will cover by burst traffic interference and become (2, 7, 11, 8, 6). The two types of interference above refer to the drift between bitrate fingerprint and traffic fingerprint which violates the uniqueness in a fine granularity observation, even though the trend keeps consistent in the long-term observation. In order to perform the similarity matching method, we relocate the traffic fingerprint by second, as shown in Algorithm 2.

The algorithm recalculates the length of the packet in sequence S and matches the element in bitrate fingerprint with the timeline. Generally, weak networks and burst traffic are infrequent, it means that if most intervals of traffic fingerprint and bitrate fingerprint are matched in the long-term trend, we can ignore a few local mismatch caused by weak network or burst traffic. However, the common similarity matching method requires that all the elements in the sequence must be matched even if the fingerprint is under interference. Therefore, we propose a fingerprint matching method considering the traffic noise interference. We define $F(x_a, x_b)$ as the Euclidean distance between x_a and x_b . For a given x_a and x_b , if $F(x_a, x_b)$ is less than threshold ϵ , the x_a is considered to match x_b . Then, a noise-resistant model $N\text{-LCS}$ based on LCS model is proposed to adapt the fingerprint mismatches.

First, for the bitrate fingerprint T^B and traffic fingerprint T^F , the points in T^B can only match the points in T^F forward (e.g., T_5^B can only match $T_{5,6,7}^F$). This is because during the

video playback, the video player will not cache the played video contents. In addition, the matching strategy of LCS is too simple to adapt the weak network condition, so N-LCS optimize the matching strategy to adapt the noise interference. For $T^B = \{t_0^b, t_1^b, \dots, t_d^b \dots\}$ and $T^F = \{t_0^f, t_1^f, \dots, t_c^f \dots\}$:

- (i) if $t_c^f > t_d^b$ and $F(t_c^f, t_d^b) < \epsilon$, the point t_c^f and t_d^b are considered to be matched.
- (ii) if $t_c^f > t_d^b$ and $F(t_c^f, t_d^b) < \epsilon$, the point t_c^f and t_d^b are considered to be not matched, and the unmatched point t_d^b may have been caused by burst traffic.
- (iii) if $t_c^f < t_d^b$ and $F(t_c^f, t_d^b) < \epsilon$, the point t_c^f and t_d^b are considered to be not matched, and the unmatched point t_d^b may caused by limited bandwidth, RTT or packet loss. As the limited traffic will usually lead to the drift of traffic features, and the backtracking function should be added to LCS model in order to drop the redundant fingerprint at the trail of T^B to avoid false matching.



We use a two-dimensional matrix M with the size of $k * k$ to save the temporary matching result, where k is the length of bitrate and traffic fingerprint. The values of matrix M are calculated by the following formula:

$$M[i][j] = \begin{cases} M[i-1][j-1], & F(t_i^f, t_j^b) > \epsilon \text{ and } t_i^f > t_j^b, \\ M[i-1][j-1] + 1, & F(t_i^f, t_j^b) < \epsilon \text{ and } t_i^f > t_j^b, \\ \max(M[i][j-1], M[i-1][j]), & F(t_i^f, t_j^b) < \epsilon \text{ and } t_i^f < t_j^b, \\ 0, & i = 0 \text{ or } j = 0, \end{cases} \quad (1)$$

$0 < i, j < k,$

where ϵ is the threshold of F . In order to eliminate the interference of feature drift, N-LCS makes two rounds of backtracking at the end of the algorithm. The first round of backtracking determines the drift distance of the traffic fingerprint and drops the fingerprint at the tail of the bitrate fingerprint according to the drift distance. The second round of backtracking will use the bitrate fingerprint calculated in the first round to find the matching path in the matrix M and calculate the longest common subsequence between two fingerprints according to the new matching path using dynamic programming as the matching result. The calculating process is shown in Algorithm 3.

Figure 8 shows the partial match result between traffic fingerprint and bitrate fingerprint. The red line shows the match relation between bitrate and traffic fingerprint. It can

be seen that the LCS-based matching model can successfully ignore the invalid features caused by interference.

7. Implementation and Evaluation

7.1. Experimental Setup. In order to build the prototype system, we have an Amazon EC2 server as the video stream server, a raspberry pi as the edge server, and an Xiaomi 11 Ultra as the victim, respectively. The server configuration is listed in Table 1. *nginx* and *ffmpeg* is used to push the video streaming in RTMP protocol, and *Wireshark* is used to simulate Man-In-The Middle (MITM) attack to capture the encrypted TLS traffic of the victim. We use videos with several bitrates to generate the bitrate and traffic fingerprint and evaluate the effectiveness of N-LCS, and the configuration of video dataset is shown in Table 2 (The data set can be found at <https://1drv.ms/u/s!AnB84OgJQM04jkAYDlzO9fhchxeZ?e=fj4cY8>).

7.2. Effectiveness of the Traffic Filter. Then we test the effectiveness of the interapplication traffic filter proposed in Section 5. We use *Wireshark* to capture the video traffic encrypted by RTMP protocol, and A domain name registered from Tencent cloud is used to fill in the SNI tags. The output traffic from the edge device and the filtered input traffic from the victim are collected, respectively, as shown in Figure 9. The results show that the proposed traffic filter can identify all the target TLS sessions accurately.

7.3. Threshold Analysis. In this part, we will calculate the threshold ϵ of N-LCS model, which is used to identify the matched point in traffic fingerprint and bitrate fingerprint. A total of 300 groups of 50 seconds bitrate fingerprints and traffic fingerprints are used to calculate the similarity distance in the following cases, and the similarity distance is shown in Figure 10:

- (i) Fingerprints come from the same video.
- (ii) Fingerprints come from different videos, but the bitrate is similar.
- (iii) Fingerprints come from different videos, and the bitrate of different videos varies greatly.

Then, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) is used to define accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (2)$$

After that, we use the intersection of two false rate lines as the threshold to maximize the accuracy. As shown in Figure 11, 239 is the best threshold to reach the maximum accuracy of 0.766 (76.6% points in the fingerprints can be accurately matched).

```

Input:
  packet sequence  $P$ ;
Output:
  packet sequence  $S$ ;
(1) while packet  $[++i] \neq \text{NULL}$  do
(2)   if ContentType == HandShake and SNI == target domain then
(3)     Create a new sequence  $s$ 
(4)     Old_IP = packet  $[i].ip$ 
(5)   else if ContentType != HandShake and packet  $[i].ip == \text{Old\_IP}$  then
(6)     Add packet  $[i].length$  to sequence
(7)   end if
(8) end while

```

ALGORITHM 1: Video filter.

```

Input:
  packet sequence  $S$ ;
Output:
  traffic fingerprint  $T^f$ ;
(1) old_time = 0
(2) acc_len = 0
(3) while packet  $[++i] \neq \text{NULL}$  do
(4)   if packet  $[i].time - \text{old\_time} \geq 1$  then
(5)      $T^f.append(\text{acc\_time})$ 
(6)     acc_len = 0
(7)     old_time ++
(8)   else if packet  $[i].time - \text{old\_time} < 1$  then
(9)     acc_len += packet  $[i].length$ 
(10)  end if
(11) end while

```

ALGORITHM 2: Traffic fingerprint relocater.

Finally, we calculate the identification accuracy with 1–100 matching points as the threshold δ in above three cases, and the results are shown in Figure 12. When the bitrate of different videos varies greatly, there are less matched points between fingerprints, and the similarity distance between mismatched points is usually large, so only a small threshold is required to achieve high accuracy. When the bitrate is similar and the length of fingerprints is short, there are also many matched points though the fingerprints that come from different videos, result in the a lower accuracy compared with other cases. Since the identification accuracy of the threshold for matching points is not 100%, the identification accuracy will eventually decrease to 0 with the increase of threshold δ . Considering the difference between fingerprints, we use 0.43 as the threshold δ in following experiments.

7.4. The Effectiveness of N-LCS without Noise Interference. Figure 13 compares the N-LCS with two popular similarity-matching methods in a noise-free environment with different fingerprint lengths.

With the increase of fingerprint length, the proportion of matched segments in fingerprints gradually stabilizes, so the accuracy of all algorithms are increasing. However, the focus

of N-LCS is to identify and remove the noise interference in the traffic fingerprint, rather than improve the matching accuracy of fingerprints without noise interference; therefore, the accuracy of N-LCS is close to Pearson. It is worth noting that the fluctuation of traffic features lead to the poor performance of DTW algorithm based on global optimal distance, and the accuracy is significantly lower than Pearson and N-LCS.

7.5. The Effectiveness of N-LCS under Noise Interference.

In order to evaluate the effectiveness of N-LCS under the noise interference, we use the automatic script to randomly generate different levels of noise interference during video playback. The fingerprint with a length of 200 seconds is used to test the interference of bandwidth limitation, burst RTT, packet loss, and burst traffic on the identification accuracy of N-LCS under different noise levels. The results are shown in Table 3 then, the traffic captured with mixed noise (bandwidth limitation, packet loss and burst traffic account for 1/3 respectively) is used to compare the N-LCS, Pearson, and DTW algorithms. The results are shown in Figure 14.

With the increase in the proportion of noise interference, the identification accuracy of above algorithms

Input:
 bitrate fingerprint T^B ;
 traffic fingerprint T^F ;
Output:
 the length of subsequence Result

```

(1)  $k = \text{len}(T^B)$ ; define matrix  $M[k][k]$  and pre  $[k][k]$ 
(2) for iterate  $T^F$  and  $T^B$  do
(3)   if  $T^F[i] - T^B[j] < \epsilon$  then
(4)     if  $T^F[i] > T^B[j]$  then
(5)        $M[i][j] = M[i-1][j-1] + 1$ , mark  $i$  and  $j$  as matched points in matrix pre
(6)     else
(7)        $M[i][j] = M[i-1][j-1]$ 
(8)     end if
(9)   else if  $M[i-1][j] > M[i][j-1]$  then
(10)     $M[i][j] = M[i-1][j]$ 
(11)  else
(12)     $M[i][j] = M[i][j-1]$ , mark  $i$  and  $j$  as noise points in matrix pre
(13)  end if
(14) end for
(15)  $i = T^F.\text{length}$ ;  $j = T^B.\text{length}$ 
(16) while iterate similarity path in pre do
(17)   if pre  $[i][j]$  holds noise points then
(18)     tmp ++
(19)   end if
(20) end while
(21)  $i = T^F.\text{length} - \text{tmp}$ ;  $j = T^B.\text{length}$ 
(22) while iterate similarity path in pre do
(23)   if pre  $[i][j]$  holds matched points then
(24)     Result ++
(25)   end if
(26) end while

```

ALGORITHM 3: N-LCS solver.

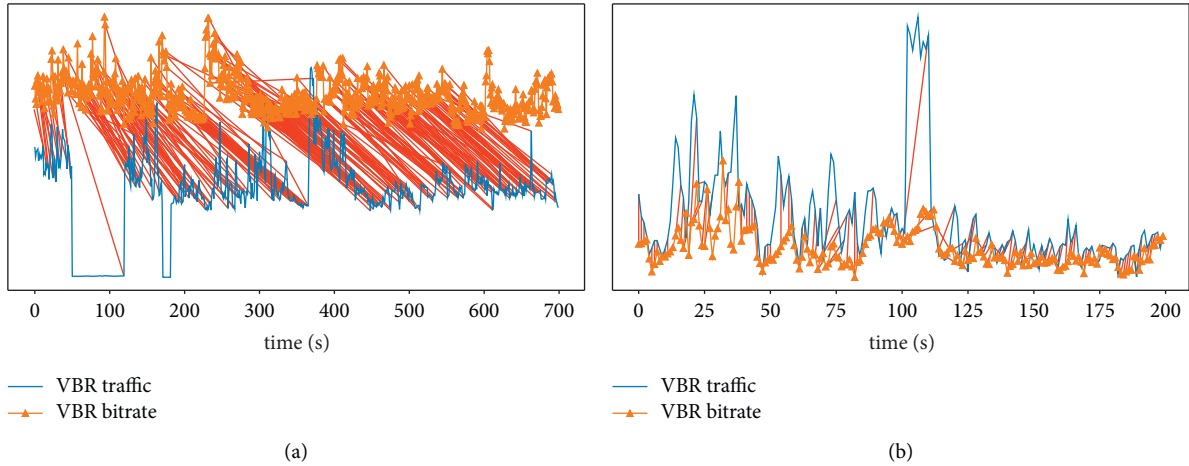


FIGURE 8: The comparison of matching result. (a) Video traffic in weak network conditions. (b) Video traffic with burst noise interference.

decreased in varying degrees, while the accuracy of DTW and Pearson decreased much faster than N-LCS. In addition, when the proportion of noise interference is less than 14%, the accuracy of N-LCS decreases slowly, while when the proportion exceeds 15%, the accuracy decreases significantly. This is because the N-LCS matching strategy reserves sufficient redundant for noise interference. The

average number of matching points between matched fingerprints is much higher than the identification threshold, and it will not have a great interference to the accuracy though there is a small amount of unmatched points. Then, we set the noise proportion to 14%, and compare N-LCS with three latest identification methods based on video fingerprint: beauty [5], p-dtw [24], and

TABLE 1: Sever configuration.

	ec2	raspberry pi
System	Windows server 2019	Ubuntu 18.04
Memory	1 GB	1 GB
Cpu	2.5 GHZ * 1	1.2 GHZ * 4
Hard disk	30 GB	16 GB
Network bandwidth	10 mbps	100 mbps

TABLE 2: Video dataset.

	Time	bitrate
Pirates of the Caribbean 5	02:48:30	10.1 mbps
Pirates of the Caribbean 5	02:48:30	8005 kbps
Pirates of the Caribbean 5	02:48:30	5991 kbps
Pirates of the Caribbean 5	02:48:30	4022 kbps
Pirates of the Caribbean 5	02:48:30	2074 kbps
Titanic	03:06:49	2607 kbps
Inception	02:28:21	1986 kbps
Avengers 3	02:29:33	2217 kbps
Trainspotting	01:34:16	1825 kbps

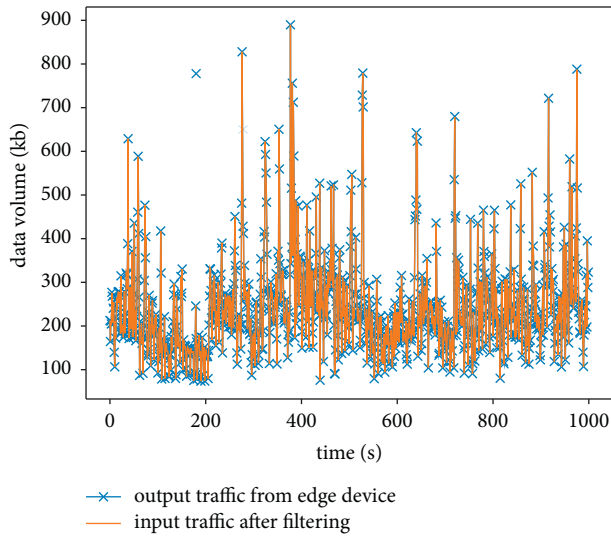


FIGURE 9: The comparison of output traffic from the edge devices and filtered traffic from the victim.

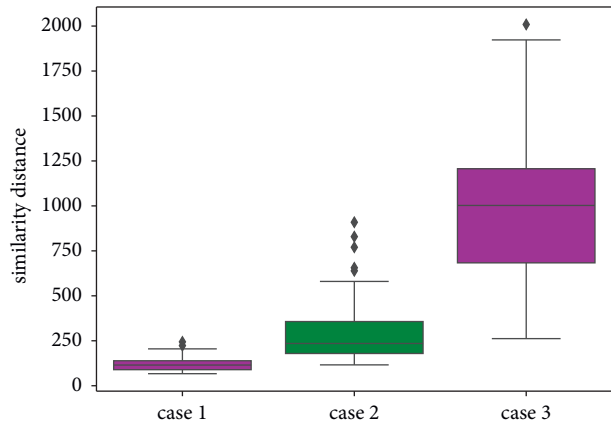


FIGURE 10: Similarity distance of our method.

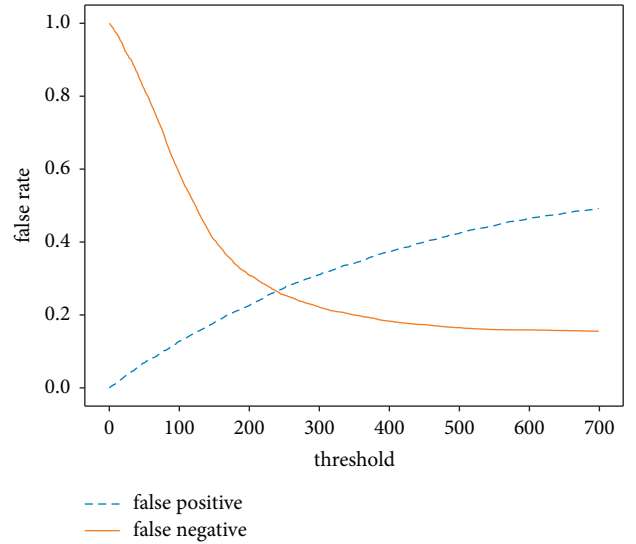


FIGURE 11: False rate of N-LCS.

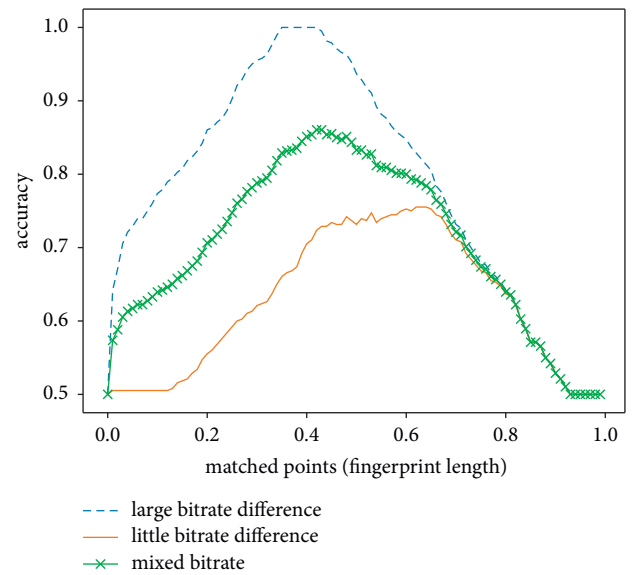


FIGURE 12: Accuracy in different bitrates.

leaky [33]. The test video clips were taken from the films *Titanic*, *Pirates of the Caribbean 5*, *Inception* and *Avengers 3*. The results are shown in Table 4. As the previous methods only focus on the accuracy of matching strategy, ignoring the noise interference from the real-world eavesdropping environments, result in the

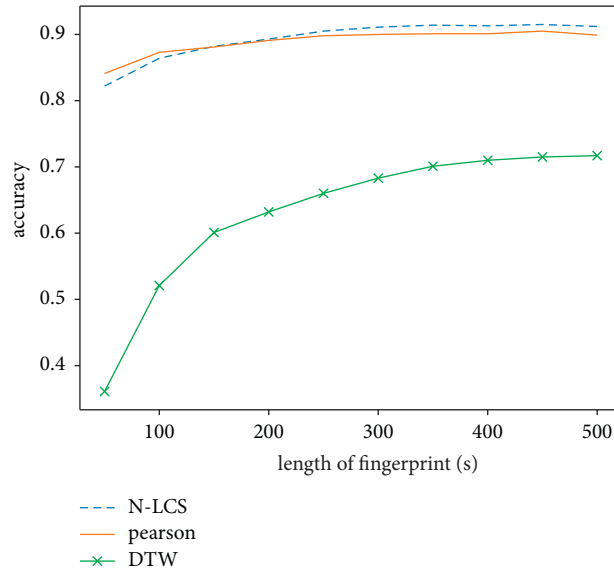


FIGURE 13: Accuracy of different methods without noise interference.

TABLE 3: Accuracy of N-LCS under different noise levels.

	2%	4%	6%	8%	10%	12%	14%	16%	18%	20%
Bandwidth limitation	0.870	0.861	0.859	0.855	0.845	0.837	0.821	0.781	0.733	0.679
Burst RTT	0.868	0.870	0.856	0.845	0.839	0.833	0.821	0.779	0.724	0.661
Packet loss	0.909	0.904	0.905	0.876	0.881	0.874	0.861	0.830	0.806	0.778
Burst traffic	0.883	0.877	0.850	0.849	0.840	0.827	0.815	0.767	0.718	0.650

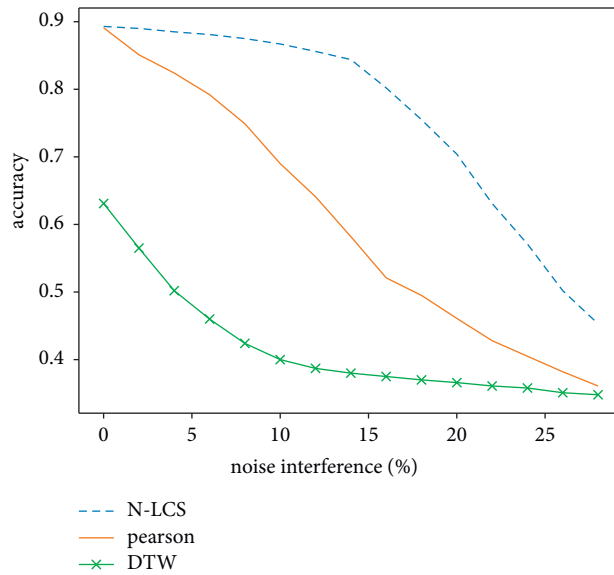


FIGURE 14: Accuracy of different methods under noise interference.

TABLE 4: Accuracy of different methods under noise interference.

	50 s	60 s	70 s	80 s	90 s	100 s	110 s	120 s	130 s	140 s
N-LCS	0.649	0.711	0.776	0.821	0.850	0.872	0.883	0.887	0.889	0.891
Beauty	0.369	0.461	0.545	0.618	0.682	0.717	0.751	0.772	0.791	0.807
P-DTW	0.420	0.491	0.553	0.605	0.649	0.677	0.701	0.721	0.737	0.752
Leaky	0.349	0.398	0.452	0.483	0.510	0.531	0.545	0.558	0.564	0.562

Here, the data unit is percentage. So, 0.649 means the accuracy is 64.9%. The bolded values represent the highest value for each column.

reduction of accuracy in the weak network condition and N-LCS can reach the highest accuracy even under noise interference.

8. Conclusion and Future Work

In this paper, we proposed a noise-resistant bitrate-based identification method for encrypted video traffic on the raspberry pi platform, which uses the LCS-based model to match the traffic and bitrate fingerprint. A real dataset using several famous movies captured from edge server and a prototype system was presented for performance evaluation. Through experiments, we proved that even the interference proportion can reach to 14%, and we can reach 89.1% accuracy after 140 seconds traffic eavesdropping.

With the prevalence of video streaming system, our work provides a new eavesdropping method that robust to interference. In the future work, we will optimize our model from the following two aspects. First, the identification accuracy will be optimized when the traffic fingerprints eavesdropped from victims are similar. Second, the proposed method only supports the popular protocols used in multimedia edge frameworks such, as RTMPS, and more protocols will be supported, for example, HLS and DASH in the future.

Data Availability

The bitrate fingerprint data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Zengkun Xie performed data collection, cleaning, and annotation, which are important parts of our work and greatly support our proposed machine learning methods.

Acknowledgments

This work is supported by National Natural Science Foundation of China 61602214.

References

- [1] "Statistic Social Media & User-Generated Content," 2021, <https://www.statista.com/statistics/1267892/tiktok-global-mau/>.
- [2] P. Becvar and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, pp. 1628–16563, Berlin, Germany, October 2017.
- [3] Y. Li, P. A. Frangoudis, Y. Hadjadj-Aoul, and P. Bertin, "A mobile edge computing-based architecture for improved adaptive http video delivery," in *Proceedings of the 2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, IEEE, 2016.
- [4] T. S. Saponas, J. Lester, C. Hartung, S. Agarwal, and T. Kohno, "Devices that tell on you: privacy trends in consumer ubiquitous computing," in *Proceedings of the 16th USENIX Security Symposium on USENIX Security Symposium, SS'07*, USENIX Association, Boston, MA, USA, August 2007.
- [5] R. Schuster, V. Shmatikov, and E. Tromer, "Beauty and the burst: remote identification of encrypted video streams," in *Proceedings of the 26th USENIX Security Symposium (USENIX Security 17)*, pp. 1357–1374, Vancouver, Canada, 2017.
- [6] H. Wu, Z. Yu, G. Cheng, and S. Guo, "Identification of encrypted video streaming based on differential fingerprints," in *Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, July 2020.
- [7] Q. Lu, S. Li, J. Zhang, and R. Jiang, "Pedr: exploiting phase error drift range to detect full-model rogue access point attacks," *Computers & Security*, vol. 114, Article ID 102581, 2022.
- [8] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao, "Dissecting video server selection strategies in the youtube cdn," in *Proceedings of the 2011 31st International Conference on Distributed Computing Systems*, IEEE, Minneapolis, MN, USA, June 2011.
- [9] J. Davidson, B. Liebald, J. Liu et al., "The youtube video recommendation system," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, Barcelona, Spain, September 2010.
- [10] D. Das, L. Sahoo, and S. Datta, "A survey on recommendation system," *International Journal of Computer Application*, vol. 160, no. 7, pp. 6–10, 2017.
- [11] Y. Zheng, C. Tian, H. Zhang, J. Yu, and F. Li, "Lattice-based weak-key analysis on single-server outsourcing protocols of modular exponentiations and basic countermeasures," *Journal of Computer and System Sciences*, vol. 121, pp. 18–33, 2021.
- [12] C. Tian, J. Yu, H. Zhang, H. Xue, C. Wang, and K. Ren, "Novel secure outsourcing of modular inversion for arbitrary and variable modulus," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 241–253, 2022.
- [13] D. Zhang, C.-Y. Chow, Q. Li, X. Zhang, and Y. Xu, "A spatial mashup service for efficient evaluation of concurrent k-nn queries," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2428–2442, 2015.
- [14] X. Cheng and X. Cheng, "A secure and lightweight data sharing scheme for internet of medical things," *IEEE Access*, vol. 8, pp. 5022–5030, 2020.
- [15] X. Lu, Z. Pan, and H. Xian, "An integrity verification scheme of cloud storage for internet-of-things mobile terminal devices," *Computers & Security*, vol. 92, Article ID 101686, 2020.
- [16] Z. Wang and D. Wang, "Achieving one-round password-based authenticated key exchange over lattices," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 308–321, 2022.
- [17] Z. Li, M. Alazab, S. Garg, and M. S. Hossain, "PriParkrec: privacy-preserving decentralized parking recommendation service," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4037–4050, 2021.
- [18] Z. Li, Z. Yang, P. Szalachowski, and J. Zhou, "Building low-interactivity multifactor authenticated key exchange for industrial internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 844–859, 2021.
- [19] J. Yu and R. Hao, "Comments on Sepdp: Secure and Efficient Privacy Preserving Provable Data Possession in Cloud Storage," *IEEE Transactions on Services Computing*, 2019.

- [20] Y. Liu, S. Zhang, J. Zhang, L. Tang, and Y. Bai, "Assessment and comparison of six machine learning models in estimating evapotranspiration over croplands using remote sensing and meteorological factors," *Remote Sensing*, vol. 13, no. 19, p. 3838, 2021.
- [21] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: survey and open issues," *IEEE Access*, vol. 6, Article ID 18209, 2018.
- [22] S. Taghavi and W. Shi, "Edgemask: an edge-based privacy preserving service for video data sharing," in *Proceedings of the 2020 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 382–387, IEEE, San Jose, CA, USA, November 2020.
- [23] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan, "A scalable and privacy-aware iot service for live video analytics," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, Taipei, Taiwan, June 2017.
- [24] J. Gu, J. Wang, Z. Yu, and K. Shen, "Traffic-based side-channel attack in video streaming," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 972–985, 2019.
- [25] T. Warren Liao, "Clustering of time series data-a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.
- [26] L. Ng and R. Ng, "On the marriage of lp-norms and edit distance," *Proceedings 2004 VLDB Conference*, vol. 30, pp. 792–803, 2004.
- [27] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, vol. 69–84, 2007.
- [28] T. Eiter and H. Mannila, "Computing Discrete Fréchet Distance," Technische Universität Wien, Vienna, Austria, CD-TR 94/64 tech. rep, 1994.
- [29] B. Su and J. Su, "One way distance: for shape based similarity search of moving object trajectories," *GeoInformatica*, vol. 12, no. 2, pp. 117–142, 2008.
- [30] X. Ding, K. Hao, X. Cai, X. S. Tang, L. Chen, and H. Zhang, "A novel similarity measurement and clustering framework for time series based on convolution neural networks," *IEEE Access*, vol. 8, Article ID 173158, 2020.
- [31] M. Wang, X. Tang, F. Chen, and Q. Lu, "Encrypted live streaming channel identification with time-sync comments," *IEEE Access*, vol. 10, Article ID 27630, 2022.
- [32] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018.
- [33] A. Reed and B. Klimkowski, "Leaky streams: identifying variable bitrate dash videos streamed over encrypted 802.11 n connections," in *Proceedings of the 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, January 2016.