International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT206492

Preventing the Video Leakages from The Traffic Streaming

Arunapriya R
Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamil Nadu, India

Article Info
Volume 6, Issue 4
Page Number: 509-516
Publication Issue :
July-August-2020

Article History
Accepted : 15 Aug 2020
Published : 22 Aug 2020

ABSTRACT

Video streaming takes up an increasing proportion of network traffic nowadays. Dynamic Adaptive Streaming over HTTP (DASH) becomes the defacto standard of video streaming and it is adopted by YouTube, Netflix, etc.Despite of the popularity, network traffic during video streaming shows identifiable pattern which brings threat to user privacy.In this paper, to proposea video identification method using network traffic while streaming. Though there is bitrate adaptation in DASH streaming, we observe that the video bit rate trend remains relatively stable because of the widely used Variable Bit-Rate(VBR) encoding. Accordingly, we design a robust video feature extraction method for eavesdropped video streaming traffic. Meanwhile, we design a VBR based video fingerprinting method for candidate video set which can be built using downloaded video files. Finally, to propose an efficient partial matching method for computing similarities between video fingerprints and streaming traces to derive video identities. To evaluate our attack method in different scenarios for various video content, segment lengths and quality levels. The experimental results show that the identification accuracy can reach up to 90%using only three minute continuous network traffic eavesdropping.

Keywords : DASH streaming, Video dataset with fingerprints, segment-based data transmission and VBRencoding, Adaptation Algorithm.

I. INTRODUCTION

Nowadays, online video streaming gets more and more popular. Ciscoreport shows that video streaming takes up a great proportion of Internet traffic and it is also in a rapid growth. The report predicts it will take up 82% ofall consumers Internet traffic by 2021. Adaptive Bitrate Streaming (ABS) based on HTTP gradually becomes the major market of video streaming due to itsadvantages of flexibility and infrastructure-friendly property. By splitting videosinto segments of multiple quality levels (bitrates), ABS enables a smart clientdriven bitrate adaptation. Dynamic Adaptive Streaming over HTTP (DASH) is arepresentative implementation of ABS. It has been an international standardsince 2011 and widely used by leading companies of video streaming, e.g.,YouTube and Netflix.

Despite of DASH's popularity, we find that its segmentbased datatransmission brings a risk of side-channel attack based on network traffic.Typically, a video in DASH is first encoded into multiple copies of different quality levels using Variable Bit-Rate (VBR) encoding. More specifically,different average bitrates, which indicate different quality levels, are used for each video copy, leading to different video size for different bitrates. Each video copy is then split into video Eavesdropping network traffic

Server
Stores video segments of multiple quality levels DASH streaming

Attacker
Video dataset with fingerprints

Client
Fetches video segments at intervals Fig. 1. DASH streaming shows distinctnetwork traffic pattern owing to its segment-based data transmission and VBRencoding. This can be used for video identification by attackers. Segments of a fixed length of playback time.

Due to video complexity variation along time, the segment size also varies along time for a video copy. Each time while streaming, a client requests a video segment in a certain quality level for playback. We find that such a mechanism in DASH results in distinct traffic pattern due to segment-based transmission and segment size variation of VBR. This can be used to identify videos while streaming, which we call side-channel video identification attack.Eavesdropping network traffic during video streaming, attackers can recognize certain pattern of the traffic. Meanwhile, a dataset of video fingerprintscan be built using downloaded video files. Attackers can then infer what videois currently streaming by comparing the traffic pattern and video fingerprints. Such traffic-based information leakage is quite serious due to the popularity of video streaming.

The design of such a traffic-based attack method faces the following challenges in practice. First, videos are encoded into multiple quality levels in DASH while these quality levels are neither prior known nor fixed. This brings challenges for generating stable and representative video fingerprints. Second,during video streaming, quality level is adaptively selected each time according to network conditions, e.g., bandwidth. Thus traffic traces of streaming the same video exhibit uncertain patterns. Last but not least, the eavesdropped traffic may not correspond to exactly a complete video, e.g., a user only watch part of the video thus the eavesdropped traffic only contain part of the video. Even the user watches the entire video, it is time-consuming to eavesdrop the entire video traffic for video identification.VBR encoding On server side, videos are encoded into multiple quality levels.

There are two common encoding schemes called Constant Bit-Rate (CBR) and Variable Bit-Rate (VBR). As the name suggests, CBR means that the rate at which the output data is consumed is constant. As opposed to CBR, VBR specifies an average bit rate constraint and varies the output data amount of video per time slot according to media complexity. Usually, the output of CBR has larger size than that of VBR. Considering streaming efficiency, VBR is adopted in most practical streaming services. To explain the effect of VBR encoding, we use a one-minute video clip as an example. First,we use 10-minute traffic trace of streaming a certain video in a fixed 6-second segment length. We randomly truncate sub traces representing different eavesdropping time for calculating dist to different video fingerprints. Red boxes show the distances between traffic traces and their matched video fingerprints. These distances are significantly below those unmatched ones. Besides, as eavesdropping time varies, distances of matched pairs keep stable. Second, we keep eavesdropping network traffic for 2 minutesin different segment lengths.

II. EXISTING METHODOLOGY

The study of popularity of YouTube videos based on meta-level features is a challenging problem given the diversity\ of users and content providers. Several models on characterizing the popularity of YouTube videos are parametric inform, where the view count time series is used to estimate the model parameters. The popularity of videos also depends on the social dynamics, i.e. the interactionof the content creators (or channels) with YouTube users. YouTube also has a social network layer on top of its media content to get popularity. Does not allow the classification of a videos view count dynamics which results from subscribers,migration, and exogenous events. By this popularity of YouTube channels willbe low and interaction of users is not good with the YouTube channels.

DISADVANTAGES

- Less interaction about YouTube channels to users.
- No optimization of Meta data after video is posted
- Less popularity of videos and channel.
- Does not allow the classification of a videos view count dynamics which results from subscribers, migration, and exogenous events.

III. PROPOSED METHODOLOGY

In the proposed system YouTube based on a large dataset. We investigatethe sensitivity of the videos meta-level features on the view counts of videos. It was found that the most important meta-level features include: first day view count, number of subscribers, contrast of the video thumbnail, Google hits,number of keywords, video category, title length, and number of upper-case letters in the title respectively. Additionally, optimizing the meta-data after the video is posted improves the popularity of the video. Morphing is a special effect in motion pictures and animations that changes (or morphs) one image or shape into another through a seamless transition. Most often it is used to depict one person turning into another through technological means or as part of a fantasy or surreal sequence. Traditionally such a depiction would be achieved through cross-fading techniques on film. Since the early 1990s, this has been replaced by computer software to create more realistic transitions.

ADVANTAGES

- Cannot Able To Upload The Duplicate
- Provide The Rating For The Video

Overview of the .NET Framework:

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet.

The .NET Framework is designed to fulfill the following objectives:

- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.

- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

Features of the Common Language Runtime

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime. With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registryaccess operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally featuring rich. The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and thirdparty language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety. The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers. Any compiler vendor who chooses to target the runtime can do so. Language compilers that target the .NET Framework make the features of the .NET Framework available to existing code written in that language, greatly easing the migration process for existing applications. While the runtime is designed for the software of the future, it also supports software of today and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs. The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

Internet Integration

The SQL Server 2000 database engine includes integrated XML support.It also has the scalability, availability, and security features required to operate as the data storage component of the largest Web sites. The SQL Server 2000 programming model is integrated with the Windows DNA architecture for developing Web applications, and SQL Server 2000 supports features such as English Query and the Microsoft Search Service to incorporate user-friendly queries and powerful search capabilities in Web applications.

Enterprise-Level Database Features

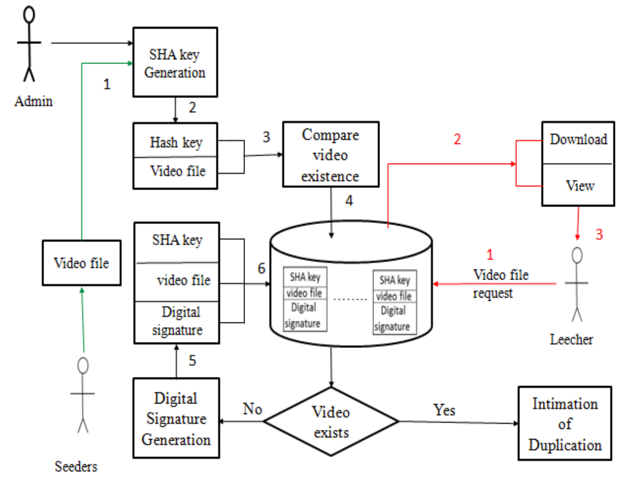
The SQL Server 2000 relational database engine supports the features required to support demanding data processing environments. The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the database. SQL Server

2000 distributed queries allow you to reference data from multiple sources as if it were a part of a SQL Server 2000 database, while at the same time, the distributed transaction support protects the integrity of any updates of the distributed data. Replication allows you to also maintain multiple copies of data, while ensuring that the separate copies remain synchronized. You can replicate a set of data to multiple, mobile, disconnected users, have them work autonomously, and then merge their modifications back to the publisher.

IV. RELATED WORK

Architecture both the process and the product of planning, designing, and constructing buildings and other physical structures. Architectural works, in

the material form of buildings, are often perceived as cultural symbols and as works of art. Historical civilizations are often identified with their surviving architectural achievements



VBR Encoding

On server side, videos are encoded into multiple quality levels. There are two common encoding schemes called Constant Bit-Rate (CBR) and Variable Bit-Rate (VBR). As the name suggests, CBR means that the rate at which the output data is consumed is constant. As opposed to CBR,VBR specifies an average bitrate constraint and varies the output data amount of video per time slot according to media complexity. Usually, the output of CBR has larger size than that of VBR. Considering streaming efficiency, VBR is adopted in most practical streaming services. To explain the effect of VBR encoding, we use a one-minute video clip as an example. It is encoded using H.264 and generates three different quality levels, i.e., 500, 1000 and 1500 kbps their respective data amount per second. Although the average bitrate is fixed, the data amount per second varies as a result of VBR. Furthermore, even in different quality levels, bitrate trend follows a specific pattern which implicitly indicates the video identity. This phenomenon inspires our idea of video fingerprinting.

Traffic of DASH

To investigate the traffic of DASH, we separately stream three different videos named Big Buck Bunny, Hockey Prodigy and Tears of Steal using DASH. For consistency, all of these three videos are encoded in an average bitrate of 1000 kbps and chunked in 6-second segment. Their respective traffic traces are shown in Figure 4. As explained previously, network traffic of DASH actually indicates periodic segment downloads. Further, DASH video streaming has three main characteristics which make it more vulnerable. First of all, streaming process shows traffic peaks because of its segment-based transmission. Thus, the traffic trace pattern is more distinct than continuous data transmission. Second, transmitted video segments are strictly in order. In other words, video segments arrive in a fixed sequence corresponding to the playback order.

Third, different from webpages, video streaming normally has longer life cycle and there is sufficient traffic data for completing attack during one single session.

VIDEO FINGERPRINTS AND TRAFFIC PATTERNS

As DASH adopts fixed length of video segment, we fingerprintvideos based on segmentation rules. Given a videoof n seconds, we calculate the data amount per second andget a sequence denoted as a = (a1; a2; : : : ; ai; : : :). However,by this naive means, different quality levels result in differentfingerprints. For this problem, we propose a differential-based method. For any adjacent data amount ai and ai□1, we use Equation 1 to calculate the differential between them. For consistency without loss of generality, we set r1 = 0 to represent no differential at the beginning. Thus, video fingerprints can be represented by r = (r1; r2; : : : ; ri; : : :). It eliminates the influence of multiple quality levels and emphasizes the bitrate
trend.ri = fdiff (ai; ai□1) = (ai □ ai□1)=ai□1 (1)
We denote transmitted data amount every second as bt attime t. Here, t is constrained to 1 _ t _ T and T is the whole eavesdropping period. In DASH, network trafficmostly consists of these two parts: client's request for newsegments and server's reply with video segment data. The former as HTTP request is negligible because it is very small
in comparison with the latter replied video data. In addition,DASH's specific MPD file also need to be transmitted beforestreaming. It also can be ignored because its amount is onlyseveral kilobytes. Thus, network traffic can be regarded asvideo data amount transmitted from server to client.

V. METHODOLOGY

- Server /Seeders Process
- Digital Signature
- Upload A Video
- Client/Leechers Process
- Apply Transformation
- Video Tracking

SERVER PROCESS (SEEDERS)

In computing, a server is a computer program or a device that provides functionality for other programs or devices, called "clients". This architecture is called the client–server model, and a single overall computation is distributedacross multiple processes or devices. Servers can provide various functionalities,often called "services", such as sharing data or resources among multiple clients,or performing computation for a client. A single server can serve multiple clients,and a single client can use multiple servers. A client process may run on the same device or may connect over a network to a server on a different device. Typical servers are database servers, file servers, mail servers, print servers, web servers, game servers, and application servers.

DIGITAL SIGNATURE CREATION

A digital signature is a mathematical scheme for demonstrating the authenticity of digital messages or documents. A valid digital signature gives a recipient reason to believe that the message was created by a known sender(authentication), that the sender cannot deny having sent the message (nonrepudiation),and that the message was not altered in transit (integrity). Digital signatures are a standard element of most cryptographic protocol suites, and arecommonly used for software distribution, financial transactions, contract management software, and in other cases where it is important to detect forgery or tampering.Digital signatures are equivalent to traditional handwritten signatures in many respects; properly implemented digital signatures are more difficult to copy than the handwritten type. Digital signature is implemented using cryptography.Digital signatures can also provide acknowledgement, meaning that the signer cannot successfully claim they did not sign a message, while also claiming their private key remains secret;

UPLOAD A VIDEO

Uploading videos to YouTube is a quick process from either your mobile device or from your computer. Follow this guide have your video online and attracting viewers in just a few minutes. Embed: If you want the video to actually show up on your Wikis paces wiki then copy this code. It will be enclosed in an iframe code. Use the Show More link to see a preview of the video. Using Show More you can also alter the size, whether or not to show suggested videos after the video finishes, the player controls, and the video title and player actions. You can also enable privacy-enhanced mode. If you change any of these, the code will update. Highlight all of the code and copy it.

CLIENT PROCESS (LEECHERS PROCESS)

A client is a computer program that, as part of its operation, relies on sending a request to another computer program (which may or may not be located on another computer). For example, web browsers are clients that connect to web servers and retrieve web pages for display. Email clients retrieve email from mail servers. Online chat uses a variety of clients, which vary depending on the chat protocol being used. Multiplayer video games or online video games may run as a client on each computer. The term "client" may also be applied to computers or devices that run the client software or users that use the client software. A client is part of a client–server model, which is still used today. Clients and servers may be computer programs run on the same machine and connect via inter-process communication techniques. Combined with Internet sockets, programs may connect to a service operating on a possibly remote system through the Internet protocol suite. Servers wait for potential clients to initiate connections that they may accept.

APPLY TRANSFORMATION (PIECE OF ATTACKS)

High quality conversion methods should also deal with many typical problems including: Translucent objects Reflections Fuzzy semitransparent object borders – such as hair, fur, foreground out-of-focus objects, thin objects Film grain (real or artificial) and similar noise effects Scenes with fast erratic motion Small particles – rain, snow, explosions and so on.

VIDEO TRACKING

Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are: human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging and video editing. Video tracking can be a time consuming process due to the amount of data that is contained in video. Adding further to the complexity is the possible need to use object recognition techniques for tracking, a challenging problem in its own right.

VI. CONCLUSION

Traffic-based attack in video streaming is a big threat to user privacy. In this paper, we propose a seamless and efficient attack method by eavesdropping network traffic while streaming. Relying on the invariant of video bit rate trend caused by VBR encoding, we design a robust video fingerprinting method. In various conditions, our identification accuracy can get up to 90% using 3-minute traffic traces. We plan to conduct our method on a larger dataset and explore its performance on online video services such as Youtube and Netflix. Meanwhile, as segment length has critical influence in our algorithm, an automatic detection method of video segment length or even streaming protocol is on our agenda. On the other hand, in face of such information leakage, countermeasures considering both network efficiency and streaming Quality of Experience (QoE) are also worth further studying.

VII. REFERENCES

[1]. C. V. networking Index, “Forecast and methodology, 2016-2021, white paper,” San Jose, CA, USA, vol. 1, 2016.

[2]. T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, “Effective attacks and provable defenses for website fingerprinting,” in Proceedings of the 123rd USENIX Conference on Security Symposium, ser. SEC’14. Berkeley, CA,USA: USENIX Association, 2014, pp. 143–157.

[3]. W. Wang and D. N. Cheng, “Skype traffic identification based on trendsaware protocol fingerprints,” in Vehicle, Mechatronics and Information Technologies II, ser. Applied Mechanics and Materials, vol. 543. Trans Tech Publications, Jun. 2014, pp. 2249–2254.

[4]. A. K. Das, P. H. Pathak, C. N. Chuah, and P. Mohapatra, “Contextual localization through network traffic analysis,” in IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, Apr. 2014, pp. 925– 933.

[5]. H. Li, Z. Xu, H. Zhu, D. Ma, S. Li, and K. Xing, “Demographics inference through wi-fi network traffic analysis,” in IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, Apr. 2016, pp. 1–9.

[6]. A. Reed and B. Klimkowski, “Leaky streams: Identifying variable bitrate dash videos streamed over encrypted 802.11n connections,” in 2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC), Jan.2016, pp. 1107–1112.

[7]. A. Reed and M. Kranch, “Identifying https-protected netflix videos in realtime,” in Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, ser. CODASPY ’17. New York, NY, USA:ACM, 2017, pp. 361–368.

[8]. R. Schuster, V. Shmatikov, and E. Tromer, “Beauty and the burst: Remote identification of encrypted video streams,” in 26th USENIX Security Symposium (USENIX Security 17). Vancouver, BC: USENIX Association, 2017, pp. 1357–1374.

[9]. P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli, “Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation,” Artificial intelligence in medicine, vol.45, no. 1, pp. 11–34, 2009.

[10]. L. J. Latecki, V. Megalooikonomou, Q. Wang, and D. Yu, “An elastic partial shape matching technique,” Pattern Recogn., vol. 40, no. 11, pp. 3069–3080,Nov. 2007.

Cite this article as :

Arunapriya R, "Preventing the Video Leakages from The Traffic Streaming", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 6, Issue 4, pp.509-516, July-August-2020. Available at doi : https://doi.org/10.32628/CSEIT206492
Journal URL : http://ijsrcseit.com/CSEIT206492