NRC Publications Archive Archives des publications du CNRC

Characterizing the impacts of VPN security models on streaming video Park, Shihyon; Matthews, Bradley; D'Amours, Danny; McIver Jr., William J.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. / La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.
For the publisher's version, please access the DOI link below. / Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

Publisher's version / Version de l'éditeur:

<https://doi.org/10.1109/CNSR.2010.60>

Eighth Annual Communication Networks and Services Research Conference (CNSR) 2010, pp. 152-159, 2010-05-14

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=2e7655bc-c8a5-4a41-a504-1e1b2d96e703>
<https://publications-cnrc.canada.ca/fra/voir/objet/?id=2e7655bc-c8a5-4a41-a504-1e1b2d96e703>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
<https://nrc-publications.canada.ca/eng/copyright>
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
<https://publications-cnrc.canada.ca/fra/droits>
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at
PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

Characterizing the Impacts of VPN Security Models on Streaming Video

Shihyon Park, Bradley Matthews, Danny D'Amours, William J. McIver, Jr.
National Research Council Canada – Institute for Information Technology
{Shihyon.Park,Bill.McIver,Bradley.Matthews,Danny.Damours}@nrc-cnrc.gc.ca

Abstract

Video-based communications are becoming more important within domains, such as health care, that have stringent security and privacy requirements for data. VPN-based encryption is a now common part of organizational security architectures. It is expected that the time required to perform encryption at a VPN concentrator and decryption at the client side video may add significant overhead to the transmission of real-time video, which is already delay-sensitive. The real impacts of this overhead on Quality of user Experience are not well-understood. This paper describes experimental methods for measuring the network-level impacts of virtual private network (VPN) encryption on real-time video in wired and wireless networks. It also presents experimental results using these methods. This paper concludes that VPN-based encryption introduces significant latency to real-time video and that further analysis is warranted to characterize the transitive impacts of such latency on the quality of user experience.

1. Introduction

The research reported in this paper was conducted in the context of a broader project concerning the utilization of video applications within health care organizations. Web-based video has become an increasingly important communication medium within many types of organizations, including those in the health care sector [1].

The deployment of video applications within many organizations necessitates encryption for security and privacy. In some cases this requirement is legally-mandated. The health care sector is a prime example of where this requirement applies. The privacy of all patient-generated data and data about patients is paramount.

Layer 3 Virtual Private Network (VPN) concentrators provide a practical means for meeting security and

privacy requirements. They execute the Internet Protocol Security (IPsec) suite to authenticate, encrypt and decrypt IP packets, while facilitating trusted communications across firewalls to permit greater mobility for people working in those organizations.

A major concern is whether latency introduced by VPN-based encryption at the concentrator and decryption at corresponding VPN clients will be significant enough to impact the Quality of user Experience (QoE) to the already delay-sensitive medium of video. Evidence gathered in previous experiments [2] and [3] showed that the overhead introduced is significant.

This latency can be seen in Figure 1 for one video session from these previous experiments using the VLC video format and system. In this case, the video was 1:32 in length and is of a roller coaster ride containing segments with slow and fast motion.

Figure 1. VLC video packet travel times: VPN vs non-VPN sessions.

This evidence of latency induced by VPN-based encryption is, of course, not enough to determine the impacts of latency on QoE; however, previous research has shown that latency can create significant perceptual problems for viewers [4] [5]. This research did not look specifically at latency introduced by encryption, however, which is the focus of the research reported in this paper.

QoE is not simply a subjective concern. Certain QoE levels may be mandated within certain health care sub-

domains to ensure effective care through the best-possible presentation of video content to clinicians.

This paper represents results from an initial research phase devoted to characterizing the impacts of VPN-based encryption on QoE within synchronous video communication sessions. In this initial phase, measurement techniques were developed for characterizing the network-level impacts of VPN-based encryption on video streams. Synchronous video communication sessions include (1) live conferencing sessions, (2) webcasting, and (3) the streaming of recorded video to remote playback clients. Latency is not of concern in the context of video file transfer since such assets can be encrypted as a whole and then decrypted at the playback endpoint where network-related latency is assumed to be minimal or non-existent.

This paper reports specifically on the impacts 3DES encryption and decryption performed by a VPN concentrator on the transfer of different types of videos, each type differing significantly from the others in terms of proportion and quantities of frame types. We constructed a testbed within the NRC-IIT network consisting of a Flash Media Interactive Server (FMIS) and a VPN concentrator that implements the 3DES algorithm. We then investigated how FMIS servers manage video streams for the three video types. Results of previous experiments using the VLC video server are also presented.

The ultimate objective of this research is to develop approaches to encrypting video streams within health care applications that exceed mandated QoE levels. VPN-based security will be increasingly important within the health care sector as organizations seek to provide patients with access points via commodity Internet services.

Future work will examine the AES-256 encryption protocol, which is becoming prominent in the health care domain.

The remainder of this paper is organized as follows. Section 2 provides background discussions related to this study. Section 3 discusses the experimental methods. Section 4 discusses the results of the experiments. Section 5 provides conclusions and a discussion of future work.

2. Background

2.1. Virtual Private Networks

As the popularity of streaming videos has continued to grow, security of streaming video data has become a major concern for both users and administrators. VPNs are increasingly being used to secure real-time video such as in video conferencing. These multimedia applications are always highly sensitive to delay. Using VPNs for

multimedia applications requires that extra headers be added to video packets which leads potentially to more data fragmentation and more packets transmitted. This compromises quality of service (QoS) for multimedia transmissions. Therefore, it is very important to understand the behaviour of VPN-encrypted video streams.

2.2. VPN-based IPsec

VPNs provide secure data communications for selected clients within an existing public network or within a network to which a broader set of untrusted clients have access. Layer 3 VPNs provide such services at the Internet Protocol layer. IP Security (IPsec) defines a specific set of protocols for this layer. IPsec was designed by the IETF.

IPsec can be used, by itself, in a transport mode, where only the data part of an IP packet is encrypted. The IP header remains unencrypted in transport mode.

The use of IPsec in a VPN involves a tunnelling mode. A VPN device establishes a session or 'tunnel' between clients who request this service and some destination. IP packets are encrypted as a whole – header and data – using public key methods supported by the VPN device. These packets are then transmitted using new IP packet headers which allow them to be routed properly from the source and destination via a VPN device where they are authenticated and decrypted by the receiving VPN client. This client is typically a software application [6]. The 3DES encryption method has been in widespread use for IPsec [7]. A newer AES-256 encryption method [8] has been adopted by a number of industry and governmental sectors around the world, including health care.

2.3. VLC vs. Flash Media Interactive Server

VLC is an open source media framework developed by the VideoLAN project, which includes a client multimedia player application and a streaming video server. The current version of the VLC client is itself capable of streaming video and is now recommended by the VideoLAN project for streaming video over the VideoLAN Server or VLS. The VLC client is capable of streaming video in MPEG-1, 2, and 4 formats [9].

Flash Media Interactive Server is an Adobe product, which is capable of streaming live video or Flash Video (FLV) files using the Real Time Messaging Protocol (RTMP) [10].

VLC and FMIS show marked difference in their performance characteristics. Results in our previous experiments [2] showed that VLC sends video packets with different transmission rates depending on the content of video clips. Preliminary experiments for the results reported in this paper and using the roller coaster clip

discussed above (see Figure. 1) show significant differences for the single packet transmission rates between VLC and FMIS. We employed Tcpdump to capture information about video packets as they were streamed from the video server to the client. To determine how FMIS and VLC transmit video packets we needed to calculate the transmission rate for individual packets to understand what the nature of inter-packet times between FMIS and VLC. The transmission rate for a single packet, ptr , is defined with respect to packet size, ps ,

and inter-packet time, ipt , as $ptr = \frac{ps}{ipt}$.

Tables 1 and 2 show inter-packet delay times for the first 20 frames of the roller coaster clip. Very small inter-packet times can be seen for FMIS compared to all inter-packet times for VLC in this one clip.

Table 1. Inter-packet times for VLC

Packet #s	Time-stamps	Inter-packet times
63496	43.720858	
63497	43.734237	0.013379
63498	43.747404	0.013167
63499	43.761112	0.013708
63500	43.774859	0.013747
63501	43.788647	0.013788
63502	43.802292	0.013645
63503	43.816029	0.013737
63504	43.829749	0.01372
63505	43.843576	0.013827
63506	43.857235	0.013659
63507	43.870995	0.01376
63508	43.884733	0.013738
63509	43.898456	0.013723
63510	43.91218	0.013724
63511	43.926152	0.013972
63512	43.940347	0.014195
63513	43.954488	0.014141
63514	43.968598	0.01411
63515	43.982744	0.014146

Table 3 summarizes the single packet transmission rates over the data in Tables 1 and 2. These results show a marked difference between FMIS and VLC and suggest that small inter-packet delay times represent short transmission delays.

Table 2. Inter-packet times for FMIS

Packet #s	Time-stamps	Inter-packet times
115	10.467948	
116	10.468003	0.000055
117	10.468033	0.000030
119	10.468405	0.000372
120	10.468435	0.000030
121	10.468465	0.000030
123	10.468499	0.000034
124	10.468523	0.000024
126	10.468798	0.000275
130	10.471003	0.002205
131	10.471052	0.000049
132	10.471081	0.000029
133	10.471106	0.000025
134	10.471132	0.000026
135	10.471158	0.000026
137	10.471380	0.000222
138	10.471410	0.000030
139	10.471444	0.000034
140	10.471466	0.000022
142	10.471498	0.000032

Table 3. Comparison of single-packet transmission rates

	VLC transmission rate (bits/s)	FMS transmission rate (bits/s)
Minimum	782,694	4,970,521
Maximum	840,890	498,181,818
Average	845,578	61,573,033

3. Methods

Little public documentation from vendors existed with regard to the streaming rate, throttling, and buffer management policies used by FMIS and VLC to guide the design experimental methods presented here. Initial ideas were found in the work of Kuang and Williamson [11] who researched the characteristics of real-time streaming on a wireless LAN. The methodology presented here was further developed out of previous research [2] and [3] in examining the bandwidth required for real-time video to provide good QoS. VLC was used as a server/client to test the characteristics of real-time video traffic such as variable bit rate (VBR) transmission and burst traffic. It was observed in these experiments that burst traffic exists depending on the content of individual clips.

3.1. Testbed architecture

We made modifications to the NRC-IIT network architecture to construct two testbeds, one using a wired network and the other using a wireless (802.11) network. These testbeds are shown in Figures 1 and 2, respectively. Both testbeds are designed to measure the encryption time of video packets in the VPN concentrator and decryption time of video packets at client side. The testbeds consist of the following network elements:

- a server running Adobe® Flash® Media Interactive Server (FMIS) instance for transmitting video packets into the network to the client,
- a workstation running a video playback client,
- a network switch,
- a Cisco VPN 3000 concentrator attached to the network switch, which establishes session-based authentication and 3-DES encryption on IP packets between the client and video server; and
- a workstation attached to the switch running a packet analyser (or “packet sniffer”).

Figure 2. Wired VPN testbed architecture.

Figure 3. Wireless VPN testbed architecture.

3.2. Experimental Data

Three different types of video clips were used for taking measurements using the methods described below:

- an interview video,
- an entertainment video, and
- a sport video.

Each video type differs significantly from the others in terms of proportion and quantities of frame types: intra-coded pictures or I-frames, predicted pictures or P-frames, and bi-predictive picture or B-frames. Different distributions of frame types impose different performance requirements on the video server and client in terms of streaming and decoding. This can be seen in a comparison of inter-packet times for each clip type in Figure 4.

Figure 4. Comparison of inter-packet times for video clip types.

The original video clips are muxed MPEG-1 files, which were used in previous experiments using the VLC video server. Each of these videos was converted to the Flash Video (FLV) format for use with the FMIS server in these experiments. The characteristics of each video clip type, obtained using the video client applications and the video capture application VirtualDub, are given in Tables 4 and 5.

Table 4. Characteristics of the MPEG-1 video clips

	Interview video	Soccer video	Entertainment video
Resolution	320 x 240	320 x 240	320 x 240
Frame rate	25.000 fps	30.000 fps	29.970 fps
File size	3.92 MB	9.48 MB	10.8 MB
Duration	52.2s	60.5s	60.6s
I-frames	110	302	125
P-frames	328	302	500
B-frames	873	1205	1248

Table 5. Characteristics of the FLV video clips

	Interview video	Soccer video	Entertainment video
Resolution	320 x 240	320 x 240	320 x 240
Frame rate	15.000 fps	15.000 fps	15.000 fps
File size	1.91 MB	5.93 MB	4.00 MB
Duration	52.467s	60.333s	62.533s
I-frames	66	76	79
P-frames	722	833	860
B-frames	0	0	0

3.3. Data Collection Methods

A single experimental session consisted of streaming each video clip type – interview, soccer, and entertainment -- from the video server to the client as measurements described below were taken. For each testbed – wired and wireless -- ten sessions using VPN-based encryption were conducted for each video clip type. A set of ten control sessions were run for each video type without VPN encryption. Only FLV files and the FMIS were used in the results reported below.

Experiments reported here were not run on completely isolated testbeds. Organizational resources and schedules have not yet permitted this. To compensate as best as possible, experiments were run late in the evening when competing traffic was minimal. In addition, the multiple sessions were intended to characterize performance across potentially variable external factors such as network congestion and CPU load.

3.3.1. VPN-side Measurements: Encryption Time. To measure the time required to transmit VPN-encrypted packets, we connected a packet sniffer to the network switch to mirror the ports on the VPN concentrator to capture incoming and outgoing packets as they enter and exit the concentrator, where 3-DES encryption is applied to them. This is shown in Figures 2 and 3. The packets that are captured through interface *eth2* on the packet sniffer are non-encrypted video packets from the video server. The packets captured through interface *eth1* on the packet sniffer are VPN-encrypted video packets. The amount of time for a packet to travel through the VPN Concentrator is the single packet time difference: *eth1* - *eth2*.

3.3.2. Client-Side Measurements: Decryption Time. There is currently no mechanism exposed to us to capture the precise start and end time of the decryption process. The tcpdump packet analyser system was employed to capture the arrival of packets at the client side but they were still encrypted at this stage. Tcpdump is an open source command-line tool for monitoring (sniffing) network traffic. Tcpdump filters packets for real-time streaming video and prints out packet information: packet frame numbers, packet time-stamps, and diverse information [12].

Our general approach to this problem was to use tcpdump to observe the time difference between the arrival of the first encrypted video packet at the client side and the acknowledgement that is sent by the media player at the client side back to the video server. This method has faults, however. The number of encrypted video packets that arrive at the client side is not consistent for a given video. To compensate, we measure the time difference from the first segment of video packets that is received until an acknowledgement is returned. It was observed through preliminary experiments that this acknowledgement coincides with a gap of a few seconds between the decryption time of video packets arriving at the client and the encryption time of the acknowledgement packet that is returned to the video server.

Such video segments in this test are defined as the time from the arrival of the first video packet to the few seconds gap that occurs because the buffer has enough video at the client side to display video. Since the acknowledgement packet is very small, the encryption time in this case is almost negligible.

3.3.3. Inter-packet delay. Inter-packet delay time is the gap between adjacent packets in seconds. This metric was captured from tcpdump time-stamps made at the video sender and those made at the egress interface, *eth1*, for VLC and FMIS.

4. Results

Figures 5, 6, and 7 show the total measured propagation times for packets in each session of the three video clip types streamed from the server to video player within the wired testbed using VPN-based encryption and without encryption. These results show, as expected, that VPN encrypted video packets generally take significantly more time to travel from server to client than non-VPN encrypted video packets. Aggregate statistics from these sessions are shown in Table 6. In addition to increases in propagation time from approximately 23% to over 66%, the introduction of VPN-based encryption brought, with one exception, greater variability in propagation times.

Figure 5. Interview clip in the wired testbed.

Figure 6. Soccer clip in the wired testbed.

As posited above, a straightforward explanation for increases in propagation times is that is due to the time required for constructing the IPsec headers for VPN encrypted video packets and for decryption on the client

end. The data suggest some possibilities for the variance in propagation times. It was observed during these experiments that the size of a single video packet increased from 1370 bytes before VPN encryption to 1458 bytes after VPN encryption. This suggests that the larger video packets might induce the VPN concentrator to fragment packets due to size. This may also introduce conditions for greater variability in propagation times.

Figure 7. Entertainment clip in the wired testbed.

Table 6. Propagation times in the wired testbed.

	Interview video	Soccer video	Entertainment video
Average propagation time without VPN encryption	0.060680	0.164927	0.119588
Average propagation time with VPN encryption	0.101057	0.175179	0.147913
% Change in propagation time with VPN encryption	66.5413	62.163	23.6858
Variance in propagation times without VPN encryption	0.000020	0.000311	0.000076
Variance in propagation times with VPN encryption	0.002555	0.000031	0.000521

Figures 6, 7 and 8 show the total measured propagation times for packets in each session of the three

video clip types streamed from the server to video player within the wireless testbed using VPN-based encryption and without encryption. These results show, as expected, that VPN encrypted video packets generally take significantly more time to travel from server to client than non-VPN encrypted video packets. Aggregate statistics are shown in Table 7. Greater variability in the propagation times was seen in the wireless testbed; however, the increases in propagation times were not as great as in the wired testbed. In addition, the degree of variation in propagation times were more similar in the wireless testbed than in the wired testbed.

Figure 8. Interview clip in the wireless testbed.

Figure 9. Soccer clip in the wireless testbed.

Figure 10. Entertainment clip in the wireless testbed.

Table 7. Propagation times in the wired testbed.

	Interview video	Soccer video	Entertainment video
Average propagation time without VPN encryption	0.199467	0.374750	0.194228
Average propagation time with VPN encryption	0.242375	0.442725	0.319934
% Change in propagation time with VPN encryption	21.5113	18.1389	64.7210
Variance in propagation times without VPN encryption	0.024833	0.096947	0.018883
Variance in propagation times with VPN encryption	0.029557	0.067729	0.007732

5. Conclusion

This paper documented initial efforts to characterize the impacts of VPN-based encryption on streaming video. Methods for estimating decryption times within a testbed were described, including a packet monitoring approach that allows the estimation of client-side decryption time, which is otherwise not directly measurable in closed video play applications.

This experimental approach was evaluated by measuring the video packet propagation times from server to client using three types of video clips in the FLV format. It was observed that VPN encryption increased the propagation time of video packets significantly. Wireless environments appear to introduce factors for greater variability in propagation times.

More extensive experimentation is now planned with a fully-isolated network. Specific mechanisms that contribute to VPN encryption-induced propagation delay will be identified as well as opportunities for optimizing the streaming of VPN-encrypted video, as suggested in [13] among others. Parallel experiments are also planned for the newer AES 256 encryption. It is expected that most health care organizations will standardize of this encryption method. Ultimately, this research is intended to characterize all impacts of VPN-based encryption affects from the network level to the real-time video quality at the client side the quality of user experience.

References

- [1] K. Gibson & S. O'Donnell, Participatory Multi-Site Videoconferencing at River Valley Health, NRC Institute for Information Technology, National Research Council Canada, Technical Report NRCC#: 50410, 2008.
- [2] S. Park & J. DeDorek, Measurement of MPEG-2 Video Streams Through a Differentiated Service Domain, Proceedings of the 2009 Seventh Annual Communication Networks and Services Research Conference – Volume 00, 2009, Pages 41–46.
- [3] S. Park & J. DeDorek, Quality of Service (QoS) for Video Transmission, Ubiquitous and Future Networks, 2009. ICUFN 2009. First International Conference on, 2009, Pages 142-147.
- [4] H. Harlyn Baker, N. Bhatti, D. Tanguay, I. Sobel, D. Gelb, Michael E. Goss, W. Bruce Culbertson & T. Malzbender, Understanding performance in coliseum, an immersive videoconferencing system, ACM Trans. Multimedia Comput. Commun. Appl. - Volume 1, 2005, Pages 190 – 210.
- [5] Lester C. Loschky & Gary S. Wolverson, How late can you update gaze-contingent multiresolutional displays without detection?, ACM Trans. Multimedia Comput. Commun. Appl. - Volume 3, 2007, Pages 1511 – 6857.
- [6] P. Knight & C. Lewis, Layer 2 and 3 virtual private networks: taxonomy, technology, and standardization efforts, Communications Magazine, IEEE. - Volume 42, 2004, Pages 124 – 131.
- [7] National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, NIST Special Publication 800-67, Version 1.1, Revised 19 May 2008.
- [8] National Institute of Standards and Technology, Announcing the ADVANCED ENCRYPTION STANDARD (AES), Federal Information Processing Standards Publication 197, November 26, 2001.
- [9] VideoLAN.org, Overview of the VideoLAN streaming solution, 2006. <http://www.videolan.org/vlc/streaming.html>
- [10] Adobe Systems Incorporated, Adobe Flash Media Interactive Server 3.5, 2010. <http://www.adobe.com/products/flashmediainteractive/>
- [11] T. Kuang & C. Williamson, A Measurement Study of RealMedia Streaming Traffic, Proceedings of SPIE-The International Society for Optical Engineering Vol 4865, 2002, Pages 68–79.
- [12] Tcpdump, <http://www.tcpdump.org/tcpdump man.html>.
- [13] G. Cote, S. Wenger, & F. Kosentini, Standard-based system for robust video transmission over the internet, Canadian Conference on Electrical and Computer Engineering Vol 2, 1999, Pages.739–744.