CS6262: Fingerprinting Attacks on QUIC Media Streaming

My Duyen Nguyen, Archit Sengupta, Aamay Mohan Puntambekar, Dipesh Rawat

July 25, 2025

Abstract

QUIC has come up as a modern transport protocol using HTTP/3 for video streaming platforms like YouTube. There is growing interest in understanding whether the encryption mechanisms are enough to prevent attacks by malicious actors. In this work we investigate the feasibility of predicting video resolution from encrypted QUIC traffic using different traffic analysis techniques. We study a controlled testbed environment based on Media over QUIC Transport (MoQT) as well as real-world traffic collected from YouTube. We analyzing features such as bitrate, packet burst size, and timing patterns and developed rules to classify video quality. Our results show that resolution prediction is accurate in a controlled MoQT testbed. However, YouTube's protocol-level defenses introduced noise that reduced classification performance. Our findings show that fingerprinting attacks exist for QUIC-based media streaming.

1 Introduction

The QUIC Protocol which has been standardized by the Internet Engineering Task Force(IETF), is a modern transport layer protocol. It is designed to improve the limitations of TCP. QUIC is built on top of UDP and introduces features like reduced connection establishment time, built-in encryption, and stream multiplexing to avoid head-of-line (HOL) blocking. QUIC uses multiple independent streams within a single connection and is able to integrate TLS at the transport layer. All of this means that it can achieve large reductions in latency and improve performance for a variety of applications like gaming, streaming, and web-browsing. The specification for QUIC was defined in RFC 9000 [1]. It defines the core protocols mechanisms for secure and reliable transport. Major platforms such as Google, Meta and Cloudflare have adopted QUIC to build several services like search, video streaming, and content delivery. QUIC has been cemented in its role as a foundational transport protocol for modern internet infrastructure.

Although QUIC encrypts most metadata at the transport layer, it cannot fully eliminate side-channel leakage through information that is observable such as packet sizes, timing information, and bitrate flows. Side channels remain a potential source of information leakage, and this is concerning given that video streaming constitutes a large portion of global internet traffic. The IETF has initiated the development Media over QUIC Transport(MoQT) in order to optimize media delivery over QUIC. It is a protocol aimed at low-latency reliable media streaming. However, despite these developments the possibility of inferring user information from QUIC-based media streams raises important privacy and security concerns.

All of this leads to an important question: despite QUIC's built-in encryption can attackers still infer/predict sensitive user information, such as the resolution of a streamed video by analyzing side-channel patterns in QUIC or MoQT traffic? If so, such vulnerabilities could be exploited for user surveillance, behavior profiling, or censorship. Different surveillance actors such as advertisers, ISPs, or government censors could exploit these patterns to profile user behavior. Therefore, understanding the extent to which encrypted QUIC-based media streaming leaks identifiable traffic patterns is crucial for evaluating the true privacy guarantees promised by next-generation transport protocols.

In the following work, we aim to investigate the feasibility of traffic analysis attacks on encrypted QUIC-based media streams. Specifically, we conduct experiments on both a controlled testbed using a MoQT implementation and on real-world traffic captured from various youtube streaming sessions. Our analysis deals with identifying video resolution and streaming characteristics through statistical information about the data like bitrate patterns without requiring access to encrypted payloads. We aim to assess how much information can be predicted/inferred from QUIC and MoQT traffic, despite their encryption and transport-layer protocols.

2 Related Work / Background

In the context of video streaming, media traffic fingerprinting is a well-studied area in network security. Foundational work in this space has focused on protocols like HTTP over TCP, where unencrypted headers and regular burst patterns made it relatively straightforward to infer user activity. Research by Andrew Reed and Michael Kranch [2] demonstrated that video resolution, codec type, and even content titles could be predicted using only TCP flow metadata such as packet size and timing.

As protocols evolved toward encrypted transport, the community's attention shifted to QUIC, a UDP-based protocol that encrypts nearly all header metadata. QUIC was standardized by the IETF in RFC 9000 [1] and now underpins HTTP/3. Because it prevents deep packet inspection and removes sequence/ack headers, QUIC was expected to neutralize traditional traffic analysis techniques.

In parallel with the evolution of QUIC, the IETF is actively developing the Media over QUIC (MoQ) protocol, designed for scalable, low-latency media delivery [6]. MoQ aims to replace traditional streaming architectures like DASH and HLS by using QUIC streams to transmit media objects such as video segments and audio frames. It introduces concepts like publish-subscribe models, object prioritization, and delivery via QUIC datagrams. While MoQ is still an Internet-Draft, early implementations exist, and its layered design opens the door for both efficient streaming and side-channel risks. To our knowledge, very little research has been conducted to evaluate fingerprinting risks specifically in MoQ-based streaming, motivating our study.

While most QUIC fingerprinting research targets web traffic, recent work by Zhao et al. [5] specifically investigates video streaming over QUIC. They propose a resolution fingerprinting method that accounts for segment-combined transmission, a defense mechanism used by platforms like YouTube. Their method uses combinatorial matching and segment-length correction to identify resolution with over 97% accuracy—even under randomized streaming conditions. This sets a strong precedent for real-time, passive attacks against encrypted video streams. Recent defenses have also explored how to harden QUIC traffic against fingerprinting. For instance, researchers at PETS 2024 proposed a tunneling approach where QUIC sessions are encapsulated within other QUIC streams to obfuscate metadata [4]. While their work focuses on website fingerprinting, it demonstrates growing interest in proactive transport-layer defenses. However, such defenses are not yet adopted in QUIC-based video streaming protocols like MoQT, leaving them susceptible to attacks such as ours.

Building on these insights, our project extends this line of inquiry to both controlled testbed environments using MoQT and real-world YouTube traffic, aiming to evaluate whether passive adversaries can reliably infer stream quality using only side-channel features such as bitrate and burst timing. Our methodology takes inspiration from prior TCP-based techniques but is redesigned to address the unique characteristics and challenges posed by QUIC's encryption and multiplexing.

3 Approach and Methodology

3.1 Attacker Threat Model

The attacker, Eve, in this scenario is a passive observer who monitors encrypted network traffic between Alice (the user) and the video streaming service. Eve does not have access to encryption keys or the actual content of the video stream. She relies on side-channel data, such as packet size, timing, and transmission rate to infer details about the video.

Eve's primary goal is to identify video resolution (e.g., 360p, 720p, or 1080p). However, she also attempts to identify other video characteristics such as the codec or bitrate by analyzing the traffic. QUIC is encrypted by default and this presents additional challenges for the attacker, as it ensures confidentiality. But she can still exploit consistent characteristics within the data across different video streams. She uses the statistical properties of the traffic to make educated guesses about the quality of the video being transmitted.

3.2 Local Testbed: MoQT Streaming Analysis

Testbed Setup

We set up a local test environment using a modified version of TugasAkhir-QUIC/public-moq-demo. It's a fork of Streaming-University's Media over QUIC Transport (MoQT) demo. This setup provided an accurate representation of the current draft of the MoQT protocol.

We could stream video content over QUIC using WebTransport (a browser protocol running HTTP/3) while maintaining full control over the environment. It also allowed repeatable experiments and packet capture for traffic analysis. We streamed content through and captured network traffic using Wireshark. We had no access to the actual video content or payload.

Video Preparation

We generated sample videos using ffmpeg, encoding each one in the H.264 video codec with AAC for audio. The videos were prepared as DASH playlists with a segment duration of 2 seconds. Shorter segment durations enabled more detailed observation of bitrate fluctuations over time.

The videos included a range of content types:

- **Low motion:** Static scenes, few edits, and slow pacing. We assumed these would present lower bitrates and data volume.
- **High motion:** Dynamic scenes with frequent cuts, camera motion, and fast pacing. These were expected to transfer data at much higher bitrates.

This range allowed us to study how content type affects streaming behavior and bitrate patterns.

Metrics Analyzed

From the captured QUIC traffic, we extracted and analyzed the following metrics:

- Average bitrate
- Burst size
- Total bytes per burst
- Timing of each burst

Among these, we picked average bitrate as the prediction criterion, since it was the most consistent and informative feature for the local testbed. It showed distinct and stable patterns across different resolutions, even when content type varied.

We analyzed our data to find the average bitrates for multiple streams. We then found consistent boundaries or value ranges associated with each resolution level. Using these observations, simple rules based on average bitrate thresholds were created. These rules were used to classify unknown streams by their resolution. This approach allowed us to check how reliable it is to infer resolution using only encrypted QUIC traffic and side-channel information.

3.3 YouTube Streaming Analysis

Background and Challenges

In the second phase of our study we wanted to analyze a real-world streaming platform, so we extended our analysis to YouTube. YouTube actively applies defenses against side-channel analysis unlike the local testbed. One such defense is the randomization of segment delivery, which aims to obscure traffic patterns and make fingerprinting more difficult, as referenced in [5]

Our goal in this phase was to detect and extract reliable patterns in network traffic even though these countermeasures exist. We began by capturing QUIC traffic from YouTube using Wireshark. However, we observed inconsistent results. The average bitrate patterns did not match the resolution levels reliably.

Protocol Switching and Capture Strategy

We had to diagnose the situation to understand what was going on. So we disabled QUIC in the browser and forced the traffic to use TCP. This approach improved consistency and we saw more predictable patterns and this made us believe there has to be a pattern in QUIC traffic as well.

We investigated further and realized that YouTube dynamically switches between QUIC and UDP during streaming. This behavior was making it difficult to isolate and analyze clean traffic flows for a single video session over QUIC itself. So, to cover all cases, we eventually switched to capturing both QUIC and UDP traffic simultaneously and analyzed them together. This gave us full visibility into the network behavior during streaming, regardless of protocol switches.

Feature Selection and Rule Creation

Once we had reliable traffic captures, we focused on burst-based features. Average bitrate wasn't reliable considering the platform sends a lot of different kinds of traffic and also randomizes media delivery.

We identified the top 5 bursts in each stream by total data volume and averaged their sizes. We found that these high volume bursts always represent delivery of media segments and they revealed important information about the resolution of the video.

We then analyzed these burst patterns across different known resolutions. Although traffic randomization introduced variability, we found that large bursts still followed consistent patterns that could be used for classification.Based on this, we created a new set of rules that mapped burst sizes and bitrates to video quality levels specifically for YouTube.

4 Findings/Results/Analysis

4.1 QUIC Media Streaming on Local Test Bed

Table 1: Training Data for Local Video Bitrates by Resolution (in kbps)

Video	360p (kbps)	720p (kbps)	1080p (kbps)
1	801	2944	4482
2	831	3098	4316
3	798	2941	4364
4	822	3101	4938
5	1050	3249	5082
6	1320	3987	5476
7	680	2964	3624

To analyze the performance of QUIC protocol in our local testbed environment, we first organized the raw bitrate measurements under a CSV. We collected data for 7 videos over 360p, 720p, and 1080p.

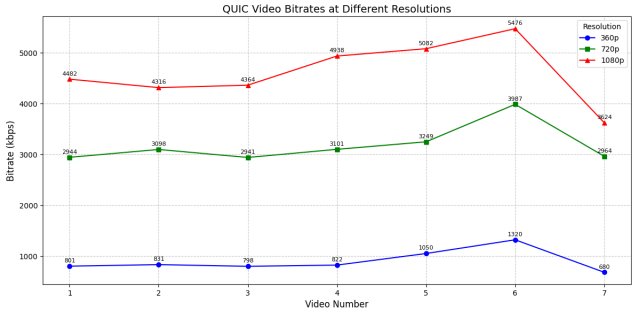
Table 1 shows a clear positive correlation between bitrate and video quality. At 360p, the graph shows that all bitrates for 360p were at least below 2000kbps. For 720p videos, the data collected for bitrates were all below 4000 kbps but above 2000 kbps. 1080p videos had bitrates larger than 3500 kbps. From analyzing these patterns in Table 1 and Figure 1, we were able to establish tighter lower and upper bounds for the three video qualities.

- 360p: bitrate $\leq$ 1500 kbps
- 720p: 2500 kbps $\leq$ bitrate $\leq$ 4000 kbps
- 360p: bitrate $\leq$ 1500 kbps
- 1080p: bitrate $\geq$ 4000 kbps

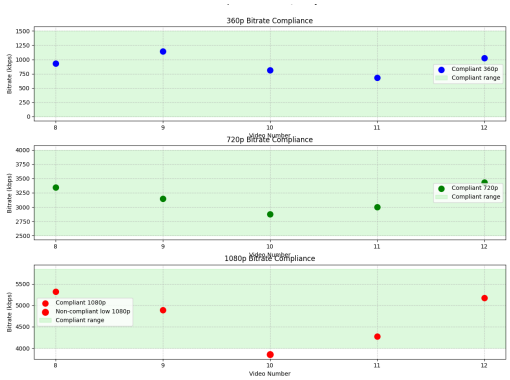
Thereafter, we collected 5 test videos to predict the video quality based on the bitrates.

Test No.	360p (kbps)	720p (kbps)	1080p (kbps)
1	929	3346	5318
2	1143	3147	4888
3	817	2878	3849
4	684	3002	4270
5	1029	3433	5169

(a) Video Quality Bitrates (kbps)



(b) Experimental testbed setup



(c) Local test data compliance

Figure 1: Local test bed video quality prediction results

Figure 1 shows each video in the test set matched the predictions developed with the training data except for test no. 3. The total accuracy achieved was $\frac{14}{15}$, 93%.

4.2 QUIC Media Streaming on Youtube

Table 2: Video Bitrate Statistics

Resolution	Avg_bitrate_for_bursts	Total_bytes_per_burst
144p	3013.9223	177 018.60
144p	2087.2530	56 830.00
144p	9645.4000	19 645.40
720p	1027.6080	60 490.80
720p	794.2983	13 922.00
720p	2486.5439	54 781.00
2160p	1052.9742	45 090.60
2160p	1443.6062	461 446.00
2160p	547.6874	22 210.80
2160p	29 281.3700	704 223.56

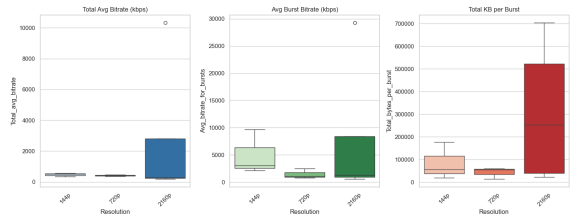


Figure 2: Box plot to show lower and upper bounds from training data.

Analyzing the dataset revealed patterns in bitrates that were inconsistent compared to the results from streaming QUIC on the local test bed, as demonstrated in Table 2 and Figure 2. There were no clear correlations between the video quality and bitrate metrics. For example, 2160p streams showed lower bitrate averages than 144p streams (182.11 kbps vs 558.02 kbps respectively), which contradicts expected bandwidth requirements. For 2160p media, the total average bitrate ranged from 182.11 to 10330.79 kbps, an extremely large range while in the same category. Burst patterns also show no consistent relationship with resolution. While Youtube does not disclose specifics on how it transmits media over QUIC, we theorize that Youtube unpredictably converts QUIC connections to plain UDP. This conversion appeared randomly throughout our sessions which explain the uncorrelated patterns in our data. This highlights the varying implementation of QUIC on a large-scale QUIC. To test this theory, we expanded our data collection method to capture both QUIC and UDP data on a diverse set of Youtube videos and identified more consistent patterns.

Table 3: Youtube QUIC Media Streaming Statistics

Resolution	Total_avg_bitrate	Avg_bitrate_for_bursts	Total_bytes_sent_per_burst
144p	4220	15 402.0000	534 212.00
144p	4894	12 463.0000	444 905.00
144p	994	25 902.0000	652 442.00
144p	582.6	15 531.0000	1 233 403.00
360p	1524.19	12 406.7317	611 093.20
360p	1984	44 777.0000	1 210 815.00
360p	6503	63 197.0000	913 640.00
360p	1840.21	16 203.5350	870 151.40
720p	4584	40 415.0000	2 520 136.00
720p	8260	94 583.0000	2 065 680.00
720p	10 165	18 034.0000	1 350 812.00
720p	10 136	25 068.0000	1 401 037.00
1080p	17 612.7	45 526.0000	2 688 331.00
1080p	5900	59 682.0000	4 284 151.00
1080p	4551.1	18 937.7318	1 682 338.00
1080p	7458.8	20 642.4499	2 795 837.20
4k	22 386.98	53 157.5039	21 835 676.00
4k	28 421	203 197.0000	19 879 211.00
4k	28 196	76 225.0000	20 682 550.00
4k	38 752	27 157.0000	38 566 643.00
4k	93 779	215 829.0000	20 659 215.00

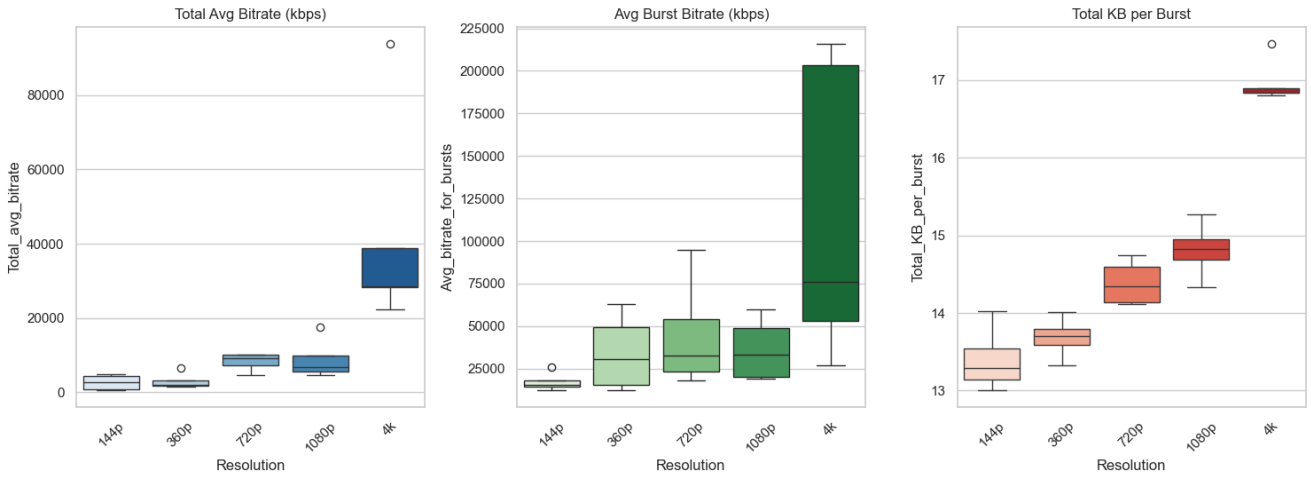


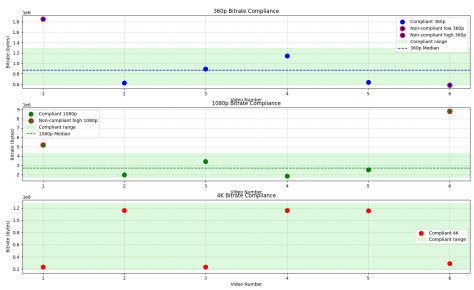
Figure 3: Box plot of the training dataset over three categories versus video qualities.

To test theory, we collected data from both media packets from UDP and QUIC on Youtube. The results are shown in Table 3. The approach, illustrated in Figure 3, revealed more consistent patterns in video quality, specifically in the correlation between Total KB per Burst. The data showed a general upward trend in bytes per burst as resolution increases. While overlap still exists between resolution categories, Figure 4 reveals patterns in the median values between quality levels. When Youtube converts between QUIC and UDP, the overall burst size remains more stable than instant bitrate measurements. Focusing on total bytes sent per burst, I established these bounds:

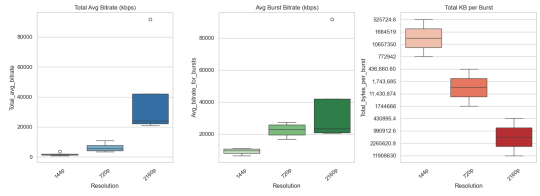
- 360p: 600,000 - 1,300,000 bytes (median: $\approx 870,000$ bytes)
- 1080p: 1,700,000 - 4,300,000 bytes (median: $\approx 2,700,000$ bytes)
- 4K: $\geq 19,000,000$ bytes (range: 19,000,000 - 39,000,000 bytes)

Video	360p (kbps)	1080p (kbps)	2160p (kbps)
1	1 855 910.8	5 183 852.0	23 132 488.2
2	621 879.6	1 972 271.8	116 194 554.0
3	895 933.2	3 412 489.4	23 376 867.0
4	1 146 426.0	1 843 087.4	116 117 461.0
5	638 467.0	2 524 853.6	115 372 214.0
6	579 178.0	8 815 917.4	29 103 760.0

(a) Test Videos for Bitrate Comparison Across Resolutions (kbps)



(b) QUIC/UDP bitrate compliance with predictions



(c) TCP packets collected over YouTube

Figure 4: (a) QUIC/UDP across resolutions; (b) QUIC/UDP compliance testing; (c) TCP packet analysis

The local testbed results and YouTube measurements show that QUIC media streaming protocol implementations vary in production environments. In our testbed, the results showed a consistent correlation between bitrate and video resolution where we were able to construct clear boundaries separating different media qualities (360p, 720p, 1080p). We were able to predict new test sets with 77% prediction accuracy (14/18). While the prediction accuracy was higher for QUIC in a controlled state, the reasonable accuracy achieved when examining QUIC in a

production environment shows that QUIC can still be fingerprinted, enabling potential video quality detection in more complex situations.

Our findings highlight the complexity of fingerprinting video quality with QUIC in real-world implementations. When collecting data on media streaming over Youtube, the inconsistencies observed suggest that large-scale deployments with QUIC may be implementing protocols differently from theoretical specifications. Youtube's dynamic switching from QUIC and plain UDP, potentially based on company protocols, disrupts clean patterns that were noticed in isolated environments. While the reason for Youtube's implementation remain unclear, our data of TCP media streaming patterns on Youtube (refer to Figure 4c) show more consistent patterns in the bitrates. These inconsistencies specifically affect QUIC streaming data while TCP remains consistent, thus easily fingerprintable.

5 Discussion

Our findings revealed that side-channel vulnerabilities still exist with QUIC and Media over QUIC Transport (MoQT). The protocol can be exploited to reveal sensitive information such as video resolution, which we tested and analyzed in this paper. The predictability of the video quality from QUIC media traffic and finger-printability implies weaknesses in its protocol in regards to user privacy. Given the use of streaming services and the adoption of QUIC across the internet infrastructure, Youtube as an implementation of QUIC has added an extra layer of security onto the scene. Whilst in the real world it may be harder to fingerprint video quality, our findings still reveal some privacy concerns in regards to QUIC for media streaming.

From a security perspective, this attribute of QUIC opens doors to allow attackers to passively infer video quality, which is a concern when it comes to adversarial profiling, censorship, and behavioral analytics. For example, ISPs could use such side channel techniques to infer content consumption patterns without decrypting traffic, raising concerns about digital surveillance.

Our local testbed results showed that resolution inference for QUIC media traffic is highly accurate (93%) using features such as bitrate and burst size. However, the inconsistencies observed in real-world traffic through Youtube (e.g.,switching of YouTube between QUIC and UDP) show how production deployments can affect fingerprinting success. These results show insight into applications can add an extra layer of protection with QUIC, but it still encourages further research into devising other security measures.

Potential defenses

- Traffic obfuscation, such as consistent bitrate padding or dummy traffic injection.
- Dynamic multiplexing/random segment ordering as seen partially in YouTube.
- QUIC over QUIC tunneling, which has recently been proposed for Web traffic defense, could be adapted for streaming content.
- Application-layer defenses, such as delaying segment delivery or introducing noise in burst sizes.

Future Work

- Investigating adaptive bitrate (ABR) streaming models and their fingerprintability.
- Generalization of fingerprinting methods across other QUIC-based streaming platforms (e.g., Netflix, Twitch).
- Evaluating the effectiveness of the proposed defenses under real-world deployment constraints.
- Exploring ML-based fingerprinting methods that can improve resolution classification despite countermeasures.

6 Conclusion

We investigated the feasibility of side-channel attacks on encrypted QUIC-based media streaming traffic. This was focusing on inferring video resolution. We implemented a passive adversary model and demonstrated high prediction accuracy (93%) in a controlled MoQT testbed. We used features such as average bitrate and burst sizes. Our extended analysis on YouTube showed practical challenges like dynamic protocol switching and delivery randomization, which reduced the accuracy of fingerprinting. These results show that encryption does stops direct content access but side-channel metadata is an important vector for privacy leakage. This shows that we need to design more privacy-preservation for the transport layer.

References

[1] J. Iyengar and M. Thomson. 2021. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000, IETF. Available: https://datatracker.ietf.org/doc/html/rfc9000

[2] A. Reed and M. Kranch. 2015. Identifying HTTPS-Protected Netflix Videos in Real-Time. Available: https://andrewreed.io/pubs/CODASPY2017_Reed_Kranch_Identifying_HTTPS_Netflix.pdf

[3] W. Barboza, S. Mattos, H. Brito, and A. Ziviani. 2022. Revisiting QUIC Attacks: A Comprehensive Review on QUIC Security and a Hands-on Study. ResearchGate Preprint. Available: https://www.researchgate.net/publication/365963028

[4] L. Raji, M. Kohls, P. Winter, and T. Pulls. 2024. Tunneling QUIC Through QUIC: On the Feasibility of QUIC-Based Website Fingerprinting Defenses. Available: https://petsymposium.org/popets/2024/popets-2024-0112.pdf

[5] Y. Zhao, H. Wu, L. Chen, S. Liu, G. Cheng, and X. Hu. 2024. Identifying Video Resolution from Encrypted QUIC Streams in Segment-combined Transmission Scenarios. *NOSSDAV '24*, ACM. Available: https://doi.org/10.1145/3651863.3651883

[6] C. Jennings, J. Finkel, and J. Corey. 2024. Media Over QUIC Transport. *IETF Internet-Draft*, draft-ietf-moq-transport-06. Available: https://datatracker.ietf.org/doc/draft-ietf-moq-transport/