Priva-Stream: Private Collaborative Live Streaming

Simon da Silva

Priva-Stream: Private Collaborative Live Streaming

Simon da Silva

Priva-Stream: Private Collaborative Live Streaming

Simon Da Silva
Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France
simon.da-silva@labri.fr

Abstract

Video streaming currently accounts for more than 75% of overall Internet traffic. However, Content Delivery Networks, which are commonly used to deliver video content, are costly to setup and do not scale well for live video streaming services.

We introduce Priva-Stream, an edge-assisted content delivery system allowing to aggregate video content from multiple edge peers in addition to existing delivery servers. Priva-Stream uses incentive mechanisms to reward participating users, and trusted execution environments such as Intel SGX to enforce privacy. Our experiments using a complete prototype show that Priva-Stream increases the quality of experience end-users can expect while at the same time providing strong privacy guarantees.

Keywords streaming, privacy, trusted execution environments

ACM Reference format:
Simon Da Silva. 2018. Priva-Stream: Private Collaborative Live Streaming. In *Proceedings of Middleware'18, Rennes, France, December 2018,* 2 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 Introduction and Background

Video streaming is already the first source of Internet traffic and is forecast to more than double by 2020. Content Delivery Networks (CDNs) are essential to the scalability and quality-of-experience of large-scale video delivery. The use of commercial CDNs comes however with significant costs. An alternative is caching video content on end-users devices and leverage direct connections for edge-assisted collaborative CDNs. This approach comes with two important challenges. End users must have an incentive to contribute their resources, or may not be interested in bearing the associated costs. Caching and serving previously-consumed video content is an important threat to user privacy, potentially allowing to map their personal interests.

We introduce Priva-Stream, an edge-assisted content delivery system allowing to aggregate video content from multiple edge peers in addition to existing delivery servers. Priva-Stream uses MS-Stream, a multi-source adaptive streaming solution, trusted execution environments such as Intel SGX to enforce privacy, and incentive mechanisms to reward participating users.

Adaptive Streaming Among the many existing on-demand and live video streaming architectures and techniques, the DASH standard [11] has emerged and is currently widely used by most big industry players such as Netflix or Youtube. The MS-Stream [2] solution was introduced. It is an extension of DASH wherein a client can simultaneously utilize multiple servers in order to aggregate bandwidth over multiple links while being resilient to network and server impairments. For each video segment, the client simultaneously instructs several servers to generate and deliver complementary sub-segments, and merges the received sub-segments so as to reconstruct a playable video segment with the highest possible visual quality over time.

Trusted Execution Environments Trusted execution environments (TEE) offer the guarantees of isolation, confidentiality and integrity of data and computations performed in untrusted environments by leveraging custom microprocessor zones. In the same line as ARM TrustZone [1], Intel designed Software Guard Extensions (SGX) [3] first introduced with the Skylake generation of processors, as an isolated execution space. SGX defines an *enclave* as an authenticated secure container that encapsulates private data and computations.

Privacy-preserving live streaming systems Most private content consumption systems rely on content and metadata encryption, often relying on Private Information Retrieval protocols. For instance, Popcorn [5] manages to enforce privacy for Netflix-like streaming systems. Other approaches consist in encrypting data at the CDN side [4] or introducing encrypted communication channels between anonymous participants [8]. However, all these solutions come with a high cost and performance overhead.

Rewarding for collaborative systems Without incentive to contribute, many users will selfishly consume resources (eg. content or bandwidth) without contributing back to the system. To try and solve this issue, some incentive and free-riding control mechanisms were designed. A few systems reward upload capacity or storage in P2P-VOD systems [12, 13]. But most mechanisms implement either direct [10] or indirect [6] reciprocity schemes. For instance, SVC-TChain [7] incentivizes good behavior in layered P2P video streaming through a triangular reciprocity scheme TChain [9]. However, in our case, the clients download video segments from other peers to obtain QoE improvements in addition to streaming from the public servers, and do not exclusively obtain data from peers. The incentive can thus not be based on reciprocity.

2 Priva-Stream

As shown on Figure 1, the Priva-Stream system is composed of four distinct entities. The **client** host an MS-Stream client. The **tracker** server fully runs inside an SGX enclave. The **MPD server** delivers manifest files. The **public server** delivers video segments.

The **tracker** hosts a key-value RAM database inside the SGX enclave. It contains the video contents identifiers (unique random ID and segment numbers), and keeps track of peers possessing the contents. When clients receive a new video segment, they post the information to the tracker which will then advertize the peer to other requesting clients. The **MPD server** delivers the manifest file, which contains technical details on the video contents, and the addresses of public servers hosting them. The **public server** is an

unmodified CDN server hosting unencrypted video contents. The peers and servers run behind an HTTP proxy inside an SGX enclave, and the tracker fully runs inside an SGX enclave, to intercept and encrypt all inbound and outbound requests.

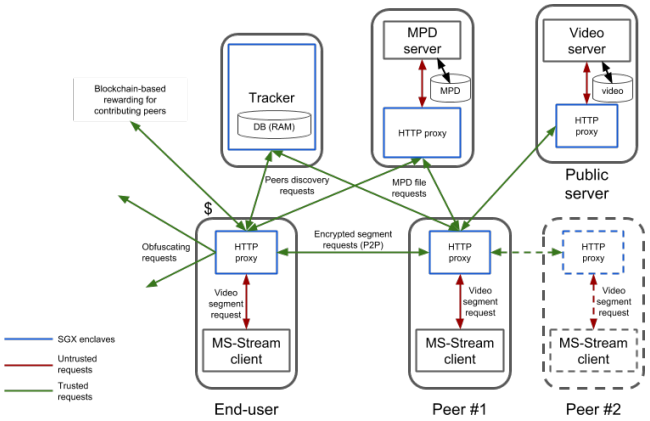

Figure 1. Priva-Stream architecture overview

2.1 System functioning

When an end-user is willing to watch a video content, he starts the Priva-Stream **client**. All communications are then intercepted by the HTTP proxy inside the SGX enclave, encrypted, and forwarded to the destination. The Priva-Stream **client** is only aware of encrypted addresses thanks to NAT performed inside the SGX enclave.

The **client** requests the **MPD server** for an MPD file, containing technical info on the video content along with public servers IP addresses. Then, the **client** asks the **tracker** for other peers which possess the video content. The **tracker** answers with a list of available peers encrypted addresses. Afterwards, the **client** sends multiple requests according to the MS-Stream specifications to a subset of peers, and to the public source server. All requests and responses are encrypted using AES-CTR mode.

3 Implementation and Evaluation Setups

We have a dozen of SGX-enabled Intel NUC which host the entities represented on Figure 1. Three NUC are setup with the servers: **tracker**, **MPD server**, **public server**. The other NUC host **peers**. To setup a realistic environment, we implement bandwidth limitations for the peers: 10 Mbps DL, 2 Mbps UL. We also implement a 10 Mbps UL limitation on all servers. Our experiment consist in a 5mn streaming session, where a single peer joins at the beginning, then a second peer after one minute, a third one after two minutes, and the others join simultaneously after three minutes. The experiment is replayed multiple times in various configurations detailed below, and aggregated to reduce distributions noise.

SGX impact In this setup we evaluate the impact of the SGX enclave on the QoE perceived by the user. To do so, we run the experiments with and without SGX enclaves, and plot QoE metrics: video bitrate, rebufferings per minute, quality changes per minute, startup delay.

Rewarding In this setup we evaluate the impact of the rewarding system on the QoE perceived by the users. We use Priva-Stream with a varying number of simultaneous sources available (i.e. number of tokens available for the users) and plot QoE metrics against the number of tokens.

Scalability In this setup we evaluate the scalability of the system in terms of number of simultaneous clients. To do so, we stress the content server, tracker server and peers with multiple simultaneous requests, and keep track of failures.

Obfuscating requests impact We analyze fake queries impact on privacy and QoE to quantify the eventual trade-off between quality and privacy.

4 Conclusion

Priva-Stream enables better scalability for live video streaming, while solving some of the main issues of edge-assisted systems, i.e. lack of privacy and free-riding. Priva-Stream uses Intel SGX to enforce privacy, and incentive mechanisms to reward participating users. Our experiments using a complete prototype show that Priva-Stream increases the quality of experience end-users can expect while at the same time providing strong privacy guarantees.

References

[1] Tiago Alves and Don Felton. 2004. TrustZone: Integrated Hardware and Software Security - Enabling Trusted Computing in Embedded Systems. (2004).

[2] J. Bruneau-Queyreix, M. Lacaud, D. Negru, J. Mongay Batalla, and E. Borcoci. 2018. Adding a New Dimension to HTTP Adaptive Streaming through Multiple-Source Capabilities. *IEEE MultiMedia* (2018).

[3] Victor Costan and Srinivas Devadas. 2016. Intel SGX Explained. *IACR Cryptology ePrint Archive* 2016 (2016), 86.

[4] S. Cui, M. R. Asghar, and G. Russello. 2017. Privacy-Preserving Content Delivery Networks. In *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*. 607–610. DOI: https://doi.org/10.1109/LCN.2017.27

[5] Trinabh Gupta, Natacha Crooks, Whitney Mulhern, Srinath Setty, Lorenzo Alvisi, and Michael Walfish. 2016. Scalable and Private Media Consumption with Popcorn. In *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation (NSDI'16)*. USENIX Association, Berkeley, CA, USA, 91–107. http://dl.acm.org/citation.cfm?id=2930611.2930618

[6] R. Landa, D. Griffin, R. G. Clegg, E. Mykoniati, and M. Rio. 2009. A Sybilproof Indirect Reciprocity Mechanism for Peer-to-Peer Networks. In *IEEE INFOCOM 2009*. 343–351. DOI: https://doi.org/10.1109/INFCOM.2009.5061938

[7] P. Rahimzadeh, C. Joe-Wong, K. Shin, Y. Im, J. Lee, and S. Ha. 2017. SVC-TChain: Incentivizing good behavior in layered P2P video streaming. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*. 1–9. DOI: https://doi.org/10.1109/INFOCOM.2017.8057140

[8] M. A. Rajan, A. Varghese, N. Narendra, M. Singh, V. L. Shivraj, G. Chandra, and B. P. 2016. Security and Privacy for Real Time Video Streaming Using Hierarchical Inner Product Encryption Based Publish-Subscribe Architecture. In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. 373–380. DOI: https://doi.org/10.1109/WAINA.2016.101

[9] K. Shin, C. Joe-Wong, S. Ha, Y. Yi, I. Rhee, and D. S. Reeves. 2017. T-Chain: A General Incentive Scheme for Cooperative Computing. *IEEE/ACM Transactions on Networking* 25, 4 (Aug 2017), 2122–2137. DOI: https://doi.org/10.1109/TNET.2017.2685560

[10] Kyuyong Shin, D. S. Reeves, and Injong Rhee. 2009. Treat-before-trick : Free-riding prevention for BitTorrent-like peer-to-peer networks. In *2009 IEEE International Symposium on Parallel Distributed Processing*. 1–12. DOI: https://doi.org/10.1109/IPDPS.2009.5161007

[11] Iraj Sodagar. 2011. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia* 18, 4 (Oct. 2011), 62–67.

[12] Weijie Wu, John C. S. Lui, and Richard T. B. Ma. 2012. Incentivizing Upload Capacity in P2P-VoD Systems: A Game Theoretic Analysis. In *Game Theory for Networks*, Rahul Jain and Rajgopal Kannan (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 337–352.

[13] W. Wu, R. T. B. Ma, and J. C. S. Lui. 2014. Distributed Caching via Rewarding: An Incentive Scheme Design in P2P-VoD Systems. *IEEE Transactions on Parallel and Distributed Systems* 25, 3 (March 2014), 612–621. DOI: https://doi.org/10.1109/TPDS.2013.94