WHITE PAPER
2019

Encrypted Video Streaming

IBM **Watson Media**

What to know about encrypted video streaming

Looking for ways to safeguard your video assets? When interviewed on the purchase decisions behind streaming technologies, 83 percent of corporate executives cited the importance of securing content from those not authorized to view it.[1] Of those who worked in companies that use live online video more than 100 times a year, 66 percent cited content security as either mandatory or very important to the purchase decision.

When it comes to securing assets, restricting access to content – such as through the use of specific permission levels – is important but another critical aspect may be encrypting content to protect it when it's transmitted over the Internet. In a sense, encryption is the key ingredient to help keep assets inaccessible to unauthorized users.

This paper looks at encryption and the different types of technologies that can be used to safeguard video assets. It includes considerations to make when choosing certain technologies and also examines different encryptions that are relevant at different states of video data, from content at rest to after playback. The paper also examines how IBM® Enterprise Video Streaming, an International Organization for Standardization (ISO) 27001 certified solution, helps encrypt video assets for organizations across these different states.

What is encryption?

Encryption is the process of encoding data and assets to keep them hidden or inaccessible by unauthorized users. If done correctly, unauthorized users will be unable to use these assets, even if they somehow gain access to them.

This process involves an encryption key that uses an algorithm to encode readable data (plaintext) into unreadable data (ciphertext). Decoding this requires a corresponding decryption key to revert the ciphertext back into readable data. The methods, types and complexity levels of encryption can vary drastically.

Video storage: encryption at rest

Video content encrypted at rest can be done through Advanced Encryption Standard (AES). AES comes in three different key sizes: 128, 192 and 256 bits. Basically, AES transforms the key and some data (plaintext) into something random, known as ciphertext. To draw meaning out of the ciphertext, AES and the same key used to transform it are required to convert it back into plaintext. The key itself is actually a number. The amount of possible numerical combinations depends on the size of the key used. An AES-128 bit key has 2^{128} possibilities. An AES-256 bit key has 2^{256} different number possibilities. That's a huge pool of potential numbers.

When it comes to comparing key sizes, more possibilities mean a lower likelihood that a brute-force attack could be successful. Now a brute-force attack is an automated trial-and-error method that generates a large number of consecutive guesses toward the desired data. The more potential numbers, the longer it would take for the average brute-force attack to be successful.

However, compromising even an AES-128 bit key could be very difficult. In 2016, it was projected that to crack just one AES-128 bit key would take 500,000,000,000[2] years. Cracking a new, different AES-128 bit key would likely take the same amount of time.

Given Moore's Law – which theorizes that computers get twice as fast every two years – this brute-force task does get easier as time progresses. The same report theorizes that with Moore's law in effect, it will be 78 years before a brute-force attack could crack an AES-128 key in a year. In 128 years, that number could go down to a second.

For most, those figures are outside of their lifetime, but the notion does showcase that technology will increase the effectiveness of brute force in time. Consequently, there is a shelf life, so to speak, for the encryption technology, something that people have known and taken action as a result. In 2015, the NSA stopped recommending AES-128 bit keys,[3] which placed a lot of attention on AES-256 bit keys. As a result, many regulations may now require the use of AES-256 bit keys, which are likely to have a much longer "shelf life" before brute-force attacks catch up.

With IBM Enterprise Video Streaming, video content is stored at rest through dm-crypt (a transparent disk encryption subsystem) using the Linux Unified Key Setup (LUKS) AES-256 bit key.

Delivery: encryption in transit

Similar to assets at rest, content in transit or during delivery can also be encrypted using AES at different key sizes. One way to do this is to use Secure Sockets Layer (SSL) or Transport Layer Security (TLS). While these two technologies are sometimes used interchangeably, TLS is the successor technology and was built from SSL 3.05. Versions of SSL prior to TLS 3.0 and lower have notable exploits that have been pointed out publicly by both Google and Mozilla. As a result, their use has been prohibited by the Internet Engineering Task Force (IETF) as of 2015 and blocked in modern browser versions of Google Chrome and Mozilla Firefox, among others.

As a result, make sure that AES and SSL are using a later TLS version. With IBM Enterprise Video Streaming, AES-256 is supported via TLS and functions through encapsulating communication between a client's or viewer's machine and the server using four protocol layers. These layers are:

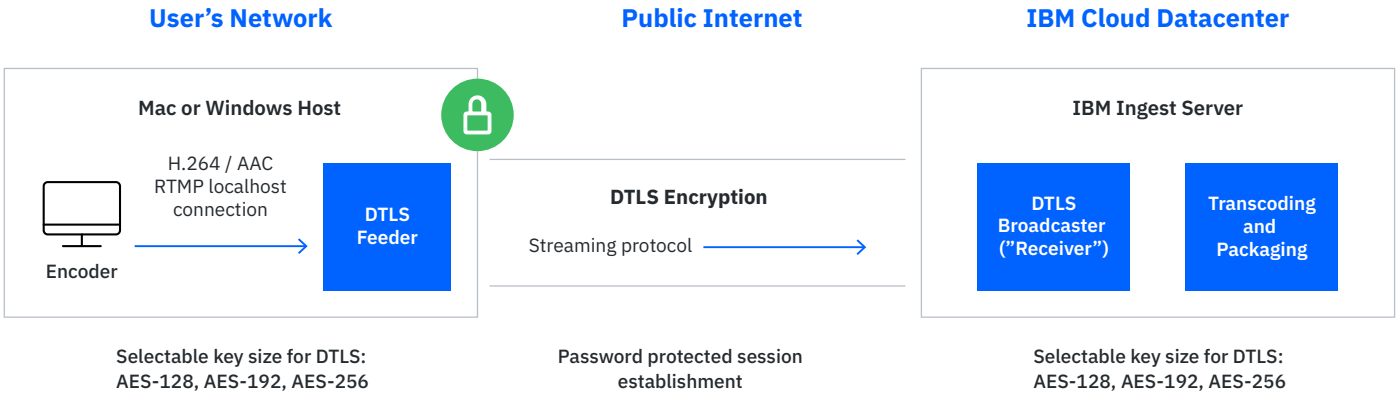
– Record Layer
– ChangeCipherSpec Protocol (signals the beginning of a secure communication)
– Alert Protocol (sends errors, problems or warnings about the connection)
– "Handshake" Protocol

Across these layers, a server presents its digital certificate to the client's machine using public key encryption to validate the certificate and confirm the server's identity claim. After a successful authentication, both the client and server establish a cipher setting and a shared key to encrypt data that is exchanged during the session.

In addition, a protocol can also be utilized for Datagram Transport Layer Security (DTLS) encryption from the encoder to ingest servers. This is a communications protocol that allows the exchange of data between appropriate systems while preventing eavesdropping and tampering.

Now DTLS is a modified version of TLS that functions over datagram transport. It reuses protocol elements of TLS with small modifications and improvements for it to work properly with User Datagram Protocol (UDP).

Encryption from the encoder to ingest servers using DTLS



Like TLS, DTLS data is carried in records and only processed once the entire record is available. A key difference, though, is that DTLS avoids fragmentation by requiring that records fit within a single datagram. There are a few benefits to this approach, such as the case datagrams carrying the remaining fragments are lost, the received ones cannot be processed. As a result, it's harder to try and circumvent the technology as DTLS also doesn't buffer partial records, meaning that host memory is used more efficiently and therefore may be less susceptible to a denial-of-service (DoS) attack.

Playback: restricting access at endpoint

Encryption of assets is only one part of the equation. Another critical piece is how content is restricted, which controls who can watch assets. There are a variety of methods for keeping content restricted, including:

Permission levels

For playback, specific permission levels can be set up to access content. This can range from simple password protection to single sign-on (SSO) restrictions. Essentially, this requires users to log in either with unique credentials or through login details that are shared with other activities from a corporate directory.

Geographic restrictions

Assets can also be restricted by geography. An example would be requiring that users can only watch the content if they are coming from a specific country, often done by checking their IP address. If the IP doesn't match, access will be blocked.

Domain control

If content is presented on websites, another method is to control which URLs are able to embed content. If the URL is different either on a selected domain level or on a specific URL level, playback is prevented.

Beyond methods to restrict content, sharing assets with other parties is another issue to consider. For example, assets might be handed over to an agency for captioning or translating, which creates the potential for them to be leaked to unintended audiences. One solution may be to keep these processes in-house as much as possible, such as generating captions yourself manually or through automated artificial intelligence (AI) solutions like IBM Watson® Captioning.

With IBM Enterprise Video Streaming, all of these restriction options are available, from setting up permission levels through SSO to creating geographic restrictions. For SSO providers, integration with services such as Okta, OneLogin, Ping Identity and more are offered. Once set up, content is restricted from use until a viewer authenticates. In addition, audit trails for administrators and managers who log in this way are also provided while specific viewer tracking is maintained for end users.

After playback: client-side storage

If the goal is to prevent content from being leaked, storing it locally on the viewer's machine can create challenges. This process – called progressive downloading – creates a temporary copy of the video on the viewer's computer (also called "caching"). This content can be watched while it's still downloading and is often done over HTTP. Since the video is stored locally, there are programs that can retrieve it from the cache and save a permanent copy.

An alternative to progressive downloading is streaming video content instead. This uses a streaming media server and streaming protocols like RTMP to serve the video content. In this scenario, the video is not stored on the client side. This can also provide an improved end user experience as viewers may skip to the end of an asset quickly without having to download the entire video first. In addition, adaptive bitrates – a process of having multiple quality versions of assets and serving the optimal version for a user's connection speed – is also supported.

Another method to protect content after playback is with digital restrictions management (DRM). This is a broad term that essentially means some sort of technological restriction is in place to limit what can be done with a media asset. Use cases can run the gamut from preventing content from playing offline to limiting the number of devices used to watch content. Since DRM may be used for different ends, it could be defined as a few simple technological limitations or could prevent the full and unrestricted use of an asset.

With IBM Streaming Manager for Enterprise, live and on-demand content is streamed. To ensure further protection, progressive downloading is not supported. Only moderators and administrators can download assets from the platform, helping to mitigate the risk of a viewer having a local, stored version of an asset.

Summary

Content security includes both restricting access and using encryption to keep unauthorized users from accessing content. Building a robust encryption strategy includes factoring in the different states' content will be in, from at rest to in transit – and also evaluating the importance of leak prevention. To achieve these goals, consider leaning toward streaming technology and finding ways to decrease the number of parties that content has to go through.

Many of these strategies and practices are already integrated into the IBM Enterprise Video Streaming offering.

Schedule a demo to learn more about how IBM Enterprise Video Streaming can provide an end-to-end solution that helps safeguard your video assets.

About IBM Watson Media

Created in January 2016, IBM Watson Media brings together innovations from IBM's R&D labs with the IBM Video Streaming platform capabilities of Clearleap® and Ustream®. Through the unit, IBM delivers a powerful portfolio of video services that spans open API development, digital and visual analytics, simplified management and consistent delivery across global industries. IBM Watson Media supports top media and enterprise companies with reliable video on-demand and streaming services.

For more information on IBM Watson Media, please visit www.video.ibm.com.

Footnotes

1 Vonder Haar, S., "Five Building Blocks for Enterprise Streaming Success," Wainhouse Research/ IBM Cloud™ Video, 2017

2 Gutteridge, L., "What's the deal with encryption strength—is 128 bit encryption enough or do you need more?," Medium, May 6, 2016

3 Information Assurance - Cryptography Today, NSA, August 19, 2015

4 Barnes, R., "The POODLE Attack and the End of SSL 3.0," Mozilla, October 14, 2014

5 Internet Engineering Task Force, June 2015

6 Kemmer, C., "Turn Off SSL 3.0 and TLS 1.0 in Your Browser," SSL.com, February 19, 2015

7 Modadugu, N., "The Design and Implementation of Datagram TLS," Stanford University, 2004

IBM Watson Media
San Francisco, CA 94108

Produced in the United States of America
March 2019

IBM **Watson Media**