Securing Video Contribution Over The Internet

By Ciro A. Noronha
Cobalt Digital

Introduction

Video transport over the Internet has been a reality for many years. Advances in compression technology have greatly reduced the bit rate required for good quality video, and infrastructure improvements have increased the available bandwidth and reliability of the Internet. Broadcasters can use the Internet as a viable video contribution alternative to costly dedicated links.

Vendor-proprietary solutions to video contribution over the Internet have existed for a number of years. In 2017, the Video Services Forum (VSF) created the Reliable Internet Stream Transport (RIST) Activity Group to create a common interoperable specification. Since then, the RIST AG has produced three Technical Recommendations: Simple Profile[1], published in 2018 and updated in 2020, RIST Main Profile[2], published in 2020 and updated in 2021, and RIST Advanced Profile[3], published in 2021.

One of the issues related to video contribution over the Internet is security. There are two aspects to security, namely content protection in flight, and authentication. This article is an overview of the security aspects built into RIST Main Profile, and extended into RIST Advanced Profile.

Content Security Aspects

When video content is transmitted over the Internet, it leaves the broadcaster's internal corporate network and traverses multiple hops over one or more ISPs to reach its final destination. These hops are controlled by independent third parties, which has the following security implications:

- **Content Security:** An unauthorized party could potentially access the content and copy it as it flows through the network, without the broadcaster's knowledge.
- **Authentication:** An unauthorized party could potentially masquerade as either the content source or destination. For example, the broadcaster may believe it is delivering content to an affiliate, but in reality, the content is going to the unauthorized party. Another example would be an incoming feed, which the broadcaster believes is from a reporter in the field, but in reality, is from an unauthorized party.

RIST Main Profile has defined two independent security levels that provide different solutions to these two issues:

- **Pre-Shared Key (PSK):** This mode of operation is based on a pre-shared passphrase. All the link participants have a priori knowledge of a passphrase, which is used to encrypt the stream.
- **Datagram Transport Layer Security (DTLS):** This mode of operation uses the datagram version of TLS, which is used to secure web sites in the Internet. DTLS is specified in RFC 6347[4]. DTLS provides both encryption and authentication.

From a conceptual standpoint, PSK is built around encryption, and knowledge of the correct passphrase is sufficient for authentication. Its main advantage is that it supports one-to-many operation, and its main disadvantage is that, if the passphrase becomes compromised, it needs to be changed on all participants (although RIST provides a mechanism to do so on-the-fly). DTLS is built around authentication, and encryption is derived from that. Its main advantages are that it is very easy to drop a compromised node, and that it can provide encryption without authentication if so desired. Its main disadvantage is that it is strictly one-to-one.

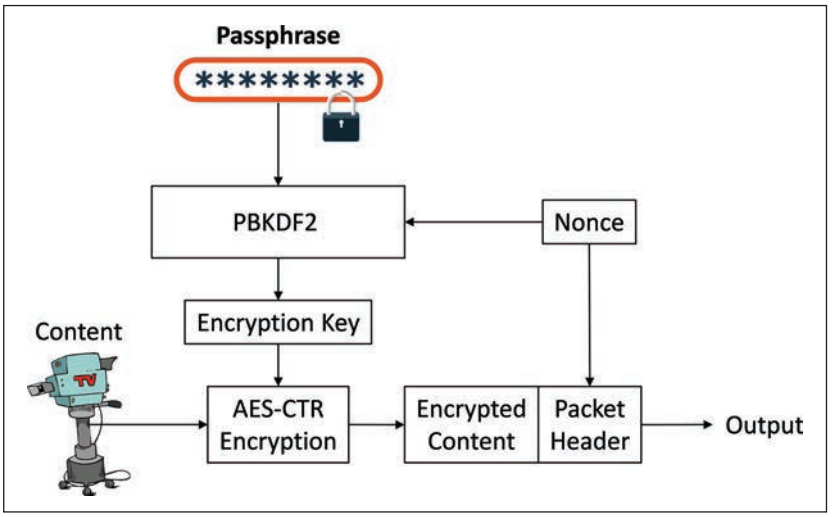

Figure 1.

In both cases, content protection is provided by encryption. In some locations in the world, there are legal restrictions on the maximum encryption key length; therefore, RIST supports both AES128 and AES256 operation in both PSK and DTLS modes.

RIST PSK Operation

In RIST PSK mode, the endpoints are pre-configured with

a secret passphrase. This passphrase is combined with a 32-bit Nonce to generate the actual encryption key, using the PBKDF2 algorithm[5] [5]. The Nonce is a random number generated by the sender, and included in the packet header. Since the receiver knows the passphrase as well, it can derive the key from the Nonce using the same method. This process is illustrated in Figure 1. The sender can rotate (change) the key at any time by simply changing the Nonce. For security purposes, this should be done periodically; using the same key for an extended period of time gives more data for an attacker to try and break it. Due to the use of AES-CTR and a 32-bit IV, the key needs to be rotated at least once every 2^{32} packets, but it should be rotated more often than this.

In PSK mode, knowledge of the passphrase is sufficient for authentication. If the passphrase becomes compromised, RIST has a mechanism to switch to a new passphrase (which needs to be also known apriori) without disturbing the stream.

RIST DTLS Operation

The core technology used in DTLS is asymmetric encryption Asymmetric encryption uses two mathematically related keys, a private key and a public key. Whatever is encrypted with the public key (which does not need be kept confidential) can only be decrypted by the private key. The main issue with asymmetric encryption is that it is computationally intense, so it is used to negotiate a symmetric encryption key for the actual communication. This way, two endpoints can establish a secure channel with no previous knowledge of each other. This solves the content protection issue, but not the authentication issue.

DTLS authentication is based on the concept of Key/Certificate pairs. A Key must be kept secret. A Certificate is derived from the Key and is public. A certificate allows secure communication, but only with the device that holds the corresponding key. A certificate may be signed by a third party called a Certificate Authority (CA). This is a third party that is trusted; if a device is prepared to trust the Certificate Authority, that trust extends to the certificates signed by it.

The certificate-based authentication process is illustrated in Figure 2. In this figure, Device B is deciding whether or not it trusts Device A. The same process can happen independently in the other direction. Device B has decided to trust certificates signed by a certain CA, so it has a copy of its CA Certificate, which was transferred to it through some secure means. Device B then receives a certificate from Device A. Unless that certificate matches the key stored in Device A, communication cannot even start. Device B can locally check the CA signature in the certificate coming from Device A against the CA Certificate it has. It that signature matches, and if Device B is prepared to trust the CA, then it will agree to communicate with Device A.

Security is maintained since:

- Even though the certificate from Device A is public, an unauthorized device cannot use it to establish communication because it does not have the corresponding key. Communication will not start.
- An unauthorized device may have a consistent certificate/key pair, but it will not be able to use it to start communication because the certificate is not signed by the CA trusted by Device B.

It is possible that a device goes "rogue"—in other words, it used to be authorized, but for whatever reason, it should not be accepted anymore. With certificates, it is simple to block a device using a field in the certificate called the Common Name—i.e., the name of the device. In a website, this is

continued on page 20

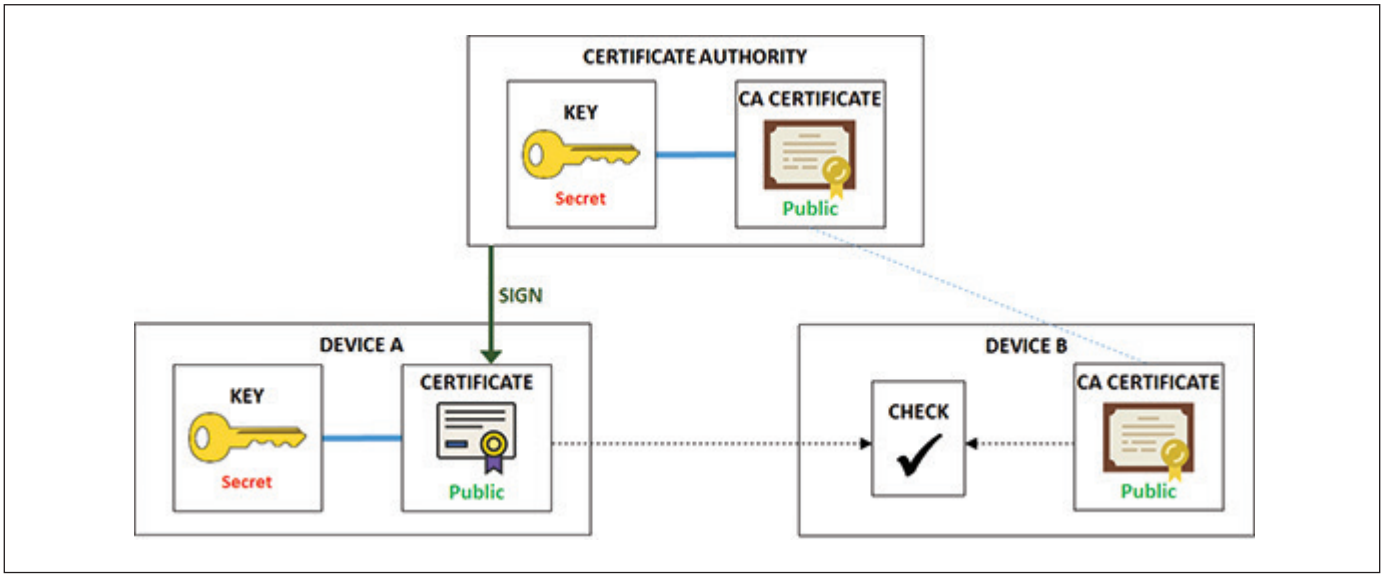

Figure 2.

Securing Video Contribution Over The Internet

continued from page 14

usually the address of the site, but it can be any text string. Referring back to Figure 2, Device B will execute an additional step after it verifies that the certificate from Device A is valid—it will check that the Common Name is not in a list of blocked devices. If it is, Device B will refuse to communicate, even though the certificate checks out.

Due to the nature of how certificates are generated, it is not possible to take a valid certificate and modify the Common Name (or any of the other fields encoded in it, including the validity date). Such an alteration will cause the certificate to become invalid (it is protected by a hash).

A broadcaster using RIST in DTLS mode will typically establish its own private CA in a secure system, and use this CA to generate certificates for each of the devices in their network. The CA does not need to be involved in the communication after that; the devices will check if the certificate is signed by the CA and will reject any endpoint that does not present such a certificate, making it impossible for an impostor to connect.

Conclusions

Video contribution over the Internet has been a reality for many years, but security cannot be an afterthought—it needs to be designed in from the beginning. Authentication is as important as content protection, especially for contribution feeds that go directly to air. This functionality is built into the RIST Specification from the Video Services Forum, allowing broadcasters to have secure communication over the Internet without being locked into a vendor-proprietary solution. Moreover, the fact that RIST is built upon vetted pre-existing technologies provides additional peace of mind.

About The Author

***Ciro Noronha**, executive vice president of Engineering, Cobalt Digital has been active in compressed video over IP since 1995, leading the development a number of commercial products. He holds a Ph.D. in electrical engineering from Stanford University where he served as a consulting professor. Noronha also holds six patents and has authored a number of papers.*

References

[1]Video Services Forum TR-06-1, "Reliable Internet Stream Transport (RIST) Protocol Specification–Simple Profile", 2020-06-25, available at *https://www.vsf.tv/download/technical_recommendations/VSF_TR-06-1_2020_06_25.pdf*

[2]Video Services Forum TR-06-2, "Reliable Internet Stream Transport (RIST) Protocol Specification–Main Profile", 2021-04-26, available at *https://www.vsf.tv/download/technical_recommendations/VSF_TR-06-2_2021-04-26.pdf*

[3]Video Services Forum TR-06-3, "Reliable Internet Stream Transport (RIST) Protocol Specification–Advanced Profile", 2021-10-19, available at *https://www.vsf.tv/download/technical_recommendations/VSF_TR-06-3_2021-10-19.pdf*

[4]Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, *https://www.rfc-editor.org/info/rfc6347*

[5]Moriarty, K., Ed., Kaliski, B., and A. Rusch, "PKCS #5: Password-Based Cryptography Specification Version 2.1", RFC 8018, DOI 10.17487/RFC8018, January 2017, *https://www.rfc-editor.org/info/rfc8018*