An Overview of Communication Manager Transport and Storage Encryption Algorithms

Abstract

The following paper provides a description of the standard algorithms that are implemented within Avaya Communication Manager™ software to secure transport of IP Telephony communications or store data related to the IP Telephony applications. The document discuses capabilities of Communication Manger software version 2.0 as well as enhancements considered for future releases.

Table of Contents

1. Introduction

The purpose of this document is to provide detailed descriptions of cryptographic algorithms that are used within the Avaya Communication Manager software. Furthermore, this document provides a description of the IP Telephony protocols that are protected through the use of cryptographic functions of encryption and authentication. It also addresses the mechanisms that Avaya has employed to protect keys used by these algorithms.

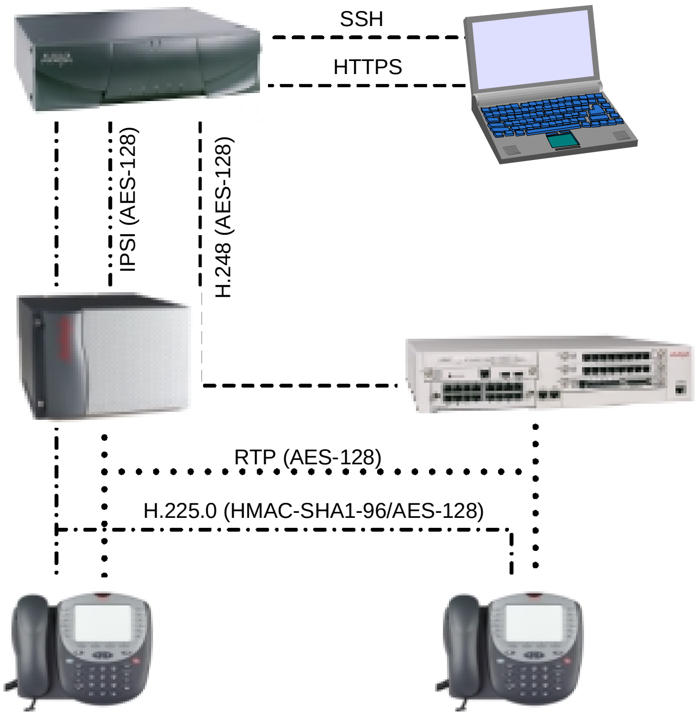

Figure 1: Multi-Connect

Communication Manager software implements cryptographic algorithms and methodologies that are generally accepted in the INFOSEC community. Furthermore, the selection of cryptographic functions is chosen based on their ability to be approved under a FIPS-140-2 or Common Criteria certification assessment.

2. Content

The remainder of this document will describe cryptographic algorithms and key management for the following data links:

- Internet Protocol Server Interface (ISPI) Link
- H.248 Link
- H.225.0 Registration, Admission, and Status (RAS)
- H.225.0 Call Signaling
- RTP
- Administrative Access
- Operational Data Storage of Account Information
- Backup of Translation Tables

Note that these descriptions apply to Avaya Communication manager version 2.1 or considered for later releases.

3. Link Security

3.1. IPSI Link Security

The IPSI link is defined as the link between the Internet Protocol Server Interface (IPSI) network interface board of the central gateway (e.g., G650) and the media server (e.g., S8700). This link relays control and signaling information between those two entities. As part of signaling, this link is also a conduit between the logical “gatekeeper” (resident in the media server) and the H.323 endpoint (through the central gateway) as depicted in Figure 1.

The IPSI link historically has been secured using 3DES, but is currently secured using the AES_128_CBC [AES] encryption algorithm, to prevent unauthorized access or modification. Inside the encrypted payload, the CRC_16 algorithm is used for error detection and to prevent unauthorized modification of the payload. Since the IPSI link is only between a specific interface card and the media server, the key that is used to secure that link only needs to be known by those two entities. As of the writing of this document, a key is pre-administered (which is stored in the IPSI flash memory and the Communication Manger software but is not accessible by administrators or users) to encrypt a dynamic 128-bit challenge for Diffie-Hellman (DH) [DH] encrypted key exchange [EKE]. Future releases may provide this pre-administered key to be assigned by the customer or dynamically negotiated.

3.2. H.248 Link Security

The H.248 link is the data link for control data between the media gateway controller (e.g. media server) and H.248 Media Gateways (e.g., branch gateways such as the G700) via the Gateway Control Protocol. The AES encryption algorithm protects data within this link and also includes a simple manipulation detection mechanism (arithmetic sum) inside the encrypted payload. The transport protocol is similar to TLS. The 128-bit symmetric key, which protects the data, is negotiated between the H.248 gateway and the media server using a DH key exchange. Each time an H.248 link is established, a new 128-bit symmetric key is negotiated using the DH key exchange.

Once the symmetric key is negotiated, it remains resident in the volatile memory of the media server and gateway, but is not accessible by users or administrators. Since the key is stored in volatile memory, it is destroyed whenever the H.248 link is re-created or whenever the media server or gateway is turned off.

3.3. H.225.0 Registration, Admission, and Status (RAS)

Before an H.323 IP endpoint can make a call, it must first register with a gatekeeper. Endpoints register and establish a signaling connection with Communication Manager using the H.323 registration and signaling standard, which is H.225.0. [ITUH2250].

The first portion of this handshake is the registration (or “RAS”) process between the endpoint with the gatekeeper.

With version 2.0 and earlier, authentication of the endpoints was achieved through a challenge-response mechanism (part of the H.225.0 registration handshake) and incorporated the use of the DES-56 encryption algorithm and derived a key based on the PIN of the extension to authenticate the registration of an endpoint.

However, a stronger registration and call signaling design is being considered.

With this new design, AES encryption and HMAC_SHA-1 authentication algorithms are implemented to secure registration of the endpoint without exposing any of the authentication credentials of the endpoint (e.g., the PIN of the endpoint) to offline attacks. This is achieved while providing registration authentication and replay protection.

Authentication keys are established through the use of an encrypted DH key exchange (1024 bits in length) [EKE] where a series of 128-bit symmetric encryption keys and 160-bit authentication keys are negotiated. Authentication of the endpoint is achieved through the use of HMAC_SHA1_96 [SHA1, HMAC]. This results in a 96-bit authentication element for the RAS messages (truncated from the 160-bits generated by SHA1).

This authentication process is part of the H.225.0 security profile in H.235 Annex H [ITUH2350H].

As part of this new registration process, the endpoint and gatekeeper negotiate multiple keys of significant size (128-bits or greater) that are used for authentication of the ongoing registration messages as well as encryption and authentication of the signaling messages.

In other words, the registration process is secure because it uses the HMAC plus SHA-1 authentication algorithms combined with an encrypted DH key exchange.

Since the keys are negotiated each time the endpoint registers, these keys are only retained in RAM of the endpoint and gatekeeper and are not accessible by users or administrators.

3.4. H.225.0 Call Signaling

Once the endpoint has successfully registered, a second H.225.0 link is established between the endpoint and the gatekeeper. This is the signaling link. It is used to transmit call-signaling messages between the gatekeeper and the endpoint. Examples include button presses, status indicators, and transmission of media encryption keys (when calls are established).

Under version 2.0 and earlier, only certain data was encrypted on this link, such as the transmission of media encryption keys. In those cases, the DES-56 encryption algorithm was used with a key derived from the PIN of the extension.

With the new registration and call signaling design, the signaling channel provides authentication of each packet using the same standard algorithm of HMAC_SHA1_96 as described above.

In addition, this new design provides data encryption. Packets with certain elements of data that would be considered sensitive will transmit those elements as ciphertext created using the AES encryption algorithm (AES-128-CTR, counter-mode). The key that is used for encrypting the data is 128-bits in length and is also derived from the master shared secrete key negotiated during the registration process.

As described above, the keys used to authenticate signaling packets and encrypt sensitive elements are dynamically negotiated each time the endpoint registers with the gatekeeper. As such, these keys are only stored in the RAM of the end devices and are not accessible by users or administrators. New session keys are created whenever the endpoints are re-registered.

Simply put, all messages sent on the signaling link are authenticated using HMAC_SHA_96 authentication algorithm. Those elements within the signaling link that need to be encrypted are secured using AES and a 128-bit key.

3.5. RTP Media Encryption

Avaya is proud to offer media encryption of RTP streams.

With the introduction of version 1.3.1, Communication Manager offered RTP encryption using the Avaya Encryption Algorithm [AEA]. AEA is an RC4-like encryption algorithm. It provides efficient, strong encryption of the media streams between endpoints or between endpoints and media processing boards. CM uses a 104-bit key for AEA.

With the introduction of version 2.0, Communication Manager also supports media encryption using the Advanced Encryption Algorithm [AEA] in counter mode using 128-bit keys.

The Secure Real Time Protocol [SRTP] is also being assessed for a future release. SRTP incorporates AES for encryption of RTP packets and HMAC_SHA1 for authentication of RTP packets and RTCP packets. The Internet Engineering Task Force (IETF) is currently considering adopting this for an RFC sometime in 2004.

In all of these media encryption solutions, the media encryption keys are dynamically created on a per-connection basis. The keys are created within the gatekeeper and transmitted to the endpoints and media processing boards via the aforementioned secure links. Additionally, separate keys are produced for the “transmit” and “receive” streams of each call. In the case of conference calls, a unique pair of keys is assigned for encrypting the payload of each endpoint (one or transmit and one for receive). With the introduction of SRTP, derivation of additional keys is performed for authentication of the RTP and RTCP [SRTP] messages.

Since all of these keys are dynamically created and assigned, they are only stored in RAM and are not accessible by administrators or users. RTP keys are not escrowed.

3.6. Administrative Access

Administrators are able to access administrative functions of the Linux-based media servers via two security protocols: SSH [SSHWG] and HTTP over SSL (HTTPS) [HTTPS]. For each of these protocols, in general, cipher suites specify which encryption algorithm, key size, and mode will be used, along with the key negotiation method.

For SSH connections, the AES cipher suite is specified (128-bit key) as the preferred algorithm. The SSH client software (provided by and located on the administrators PC), negotiates with the media server to determine which cipher suite will actually be used. As long as the SSH client supports AES and is listed as a preference, it will be used for encrypting the data. Details of the authentication are provided by standards defined by [SSHWG].

For the web interface, the media servers support HTTPS. Under HTTPS the HTTP data is encrypted authenticated within a SSL (or TLS) connection. Similar to SSH, the SSL cipher suite is negotiated with preferred and prioritized cipher suites provided by the client and server during the SSL handshake. The media servers list 3DES as a preferred encryption algorithm. However, newer versions of TLS will support AES. When this is implemented, AES will be listed as the preferred encryption algorithm. Authentication of the messages is provided as part of the TLS protocol [TLS]

Regardless of transport method (SSH or HTTPS), the session keys are negotiated each time a link is established. These session keys are discarded at the end of the session and are not retained in flash memory.

3.7. Operational Data Storage of Account Information

During normal operations, Communication Manager manages numerous extensions as well as administrative accounts. This information is considered sensitive because unauthorized access could provide an attacker the means to achieve access to or change the privileges of administrative accounts or extensions on the system.

During operations, the administrative credentials (most importantly the administrative passwords) are stored as an MD5 hash inside of a Linux shadow password file. For more detail on this file, see [RHSG].

However, if the administrative account is ASG-protected, the account does not use passwords for authentication. Instead, during the login process, the administrator must provide a correct response to a dynamic ASG challenge. The correct response is the encryption challenge using a unique ASG key. The ASG key is stored in an ASGFILE or LACFILE (for support accounts) and both are encrypted using 3DES. The 3DES key is pre-administered inside Communication Manager software.

The extension and PIN information is stored in Translation Tables of Communication Manager. Aside from imposing tight software controls on the access of the Translation Tables, the PIN information is encrypted using 3DES in the Translation Table files and a pre-administered key stored in Communication Manger software.

Future versions of Communication Manager will provide for greater support of standardized central authentication mechanisms such as RADIUS, LDAP, and Kerberos.

3.8. Backup of Translation Tables

Communication Manager provides the ability to backup Translation Tables for offsite storage. When these tables are backed-up, the customer has the option of providing a pass phrase and encrypting those backup files. When this is done, the GnuPGP [GPGP] utility is used to encrypt the backup file using the CAST5 encryption algorithm (default setting for this utility which uses a 128-bit key based on the pass phrase). Future version will standardize on the AES encryption algorithm (128-bit).

4. Conclusion

Within Communication Manager, communications are secured from end-to-end using standard encryption and authentication algorithms. Keys are dynamically generated and are stored in RAM where they are overwritten whenever the links disabled or re-created. Additionally, all links support the use of the AES algorithm for encryption using 128-bit keys. When authentication is used, the HMAC_SHA1_96 authentication algorithm is implemented.

The bottom line is that customers can have confidence in Avaya’s VoIP solutions because of the implementation of standard encryption and authentication algorithms, use of dynamic key negotiation, and incorporation of this capability as a fundamental part of the standard product offering of Avaya media servers, gateways, and endpoints.

5. Additional References

This section is optional. If you have other references such as product documentation that you wish to point out to the user you may add pointers here.

[AES] Advanced Encryption Standard, FIPS-197, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[DH] W. Diffie and M.E. Hellman, “New Directions in Cryptography,” IEEE Transactions in Information Theory, v. IT-22, n. 6, Nov 1976, pp. 664-654

[EKE] Bellovin and Merritt, U.S. Patent 5,241,599, August 31, 1993, assigned to Lucent Technologies AT&T Bell Laboratories.

[GNUPG] www.gnupg.org

[HMAC] H. Krawczyk, M. Bellare, and R. Canetti, HMAC: Keyed-Hashing for Message Authentication, IETF Informational RFC 2104, February 1997.

[HTTPS] E. Rescorla; “HTTP over TLS”; RFC 2818, http://www.ietf.org/rfc/rfc2818.txt

[ITUH2250] ITU-T Recommendation H.225.0, "Call signaling protocols and media stream packetization for packet-based multimedia communication systems"

[ITUH235H] ITU-T H.235 Amendment 1, "Security and encryption for H-series (H.323 and other H.245-based) multimedia terminals," Annex H

[RHSG] The Official Red Hat Security Guide, http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/pdf/rhl-sg-en-80.pdf

[SHA1] FIPS PUB 180-1, Secure Hash Standard, U.S. Department of Commerce, Technology Division, National Institute of Standards and Technology, April 17, 1995.

[SRTP] Baugher, Carrara, Naslund, Norrman; “SRTP: The Secure Real Time Transport Protocol,” IETF RFC Pending, http://www.ietf.org/internet-drafts/draft-ietf-avt-srtp-09.txt

[SSHWG] IETF Secure Shell Working Group (secsh), multiple IETF Internet Drafts, http://www.ietf.org/html.charters/secsh-charter.html

[TLS] T. Dierks, C. Allen; “The TLS Protocol,” IETF 2246, http://www.ietf.org/rfc/rfc2246.txt