AVT Working Group
Internet Draft
Intended status: Informational
Expires: April 2011

G. Feher
BME

October 19, 2010

Using approximate authentication with Secure Real-time Transport Protocol (SRTP)

draft-feher-avt-approx-auth-srtp-00.txt

Status of this Memo

Copyright Notice

Abstract

This document describes how to use an approximate authentication algorithm in the Secure Real-time Transport Protocol (SRTP) to provide integrity protection for the Real-time Transport Protocol (RTP) traffic.

Approximate authentication is a class of authentication algorithms where the authentication does not require the exact match of the source and received data, but a parameter given, certain amount of deviation is acceptable.

Table of Contents

1. Introduction

Message authentication is a protection tool to protect the transmitted data during the transmission. The protection can signal whether the transmitted message is the same at the sender and the receiver or not. Due to this technique no modifications by adversaries and no natural transmission errors remain undetectable by the receiver. Message authentication often utilizes keyed cryptographic hash functions.

In the case of wired data transmissions bit errors are rear, usually this channel is considered error free. In contrast, in the case of wireless channels bit errors during the transmission are common. The transmitted erroneous data frames are dropped by the receiver based on an integrity check. In order to avoid problems caused by these

drops, often an error correction protocol is applied at link level. The simplest error correction is to repeat the transmission when the receiver does not acknowledge the arrival of the transmitted data. This is used e.g. in WiFi unicast connections. However, there are certain scenarios where simple error correction is not applicable, and actually there is no error correction at all in this case. As an example, in the case of multicast WiFi transmissions, which are often used to transmit audio and video data, there is no protection against drops caused by natural bit errors.

Nowadays there are already available technologies that are able to cope with bit errors. Even if bit error correction is not possible, errors can be concealed. Depending on the amount of errors, the experienced results of bit error concealments could be better than dropping whole packets. Bit errors are even supported on transport layers with protocols, such as UDP-Lite [RFC3828]. This latter document also mentions radio technologies (e.g. [3GPP]) that permit partially damaged frames in the MAC layer.

Overall, there are real scenarios, where audio and video data are transmitted during error prone radio channels. Due to the error resilience and error concealment techniques in the decoders, users may benefit from the received, but not dropped damaged packets. However, when the user selects secure audio or video transmission, using SRTP [RFC3711] with a cryptographic hash based authentication, all the damaged packets should be dropped as they fail the authentication. For this reason, flows protected with current SRTP algorithms are not able to profit from the damaged packets.

Approximate authentication is a class of authentication algorithms where the authentication does not require the exact match of the source and received data [DONGVU][FEHER][GRAVISH]. The user is able to give a threshold for the number of mismatching bits, and if the number of errors is below of such threshold, then the packet is authenticated. Authenticating a modified packet is not necessarily a security problem. When the authentication threshold is low, then the adversary is not able to spoof the whole content. Assuming that the content is encrypted, the adversary cannot perform predictable content spoofing. No hijacking is possible either. As a maximum, the adversary may turn down the quality of the transmitted media by modifying some bits in the stream, but this impact should be small compared to the case, where damaged packets are dropped.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

2.1. Stream ciphers and block ciphers

In this draft ciphers are mentioned at several places. Basically ciphers could be block ciphers or stream ciphers. The differences between the two types of ciphers are:

- Block ciphers deal with larger blocks of information (i.e. usually 64 or 128 bit blocks), while stream ciphers work bit by bit.

- Stream ciphers have memory

It is possible to convert block ciphers to stream ciphers back and forth, so it might not be obvious when a certain cipher in a certain operation mode is a block cipher or a stream cipher. Further on in this document, the stream ciphers refer to cryptographic algorithms that use some function (i.e. mostly the XOR function) to combine a pseudorandom number generator (PRNG) with a plaintext/ciphertext stream. They are also called as combiner-type algorithms in some NSA documents. Block ciphers refer to algorithms that operate on groups of bits of a fixed length, termed blocks. An important property of a block cipher is the avalanche effect, in the meaning that a slight change in the ciphertext causes a significant change in the decoded plaintext.

According to the previous distinction, AES-ICM (i.e. AES block cipher in Integer Counter Mode) is considered as a stream cipher, while AES-CBC (i.e. AES block cipher in Cipher-block Chaining mode) is considered as block cipher.

3. SRTP message authentication with approximate authentication

During the approximate authentication procedure, at the source side, a message authentication tag will be calculated and inserted to the authentication part of the SRTP message. This authentication takes care of the message header and payload separately. For security reason the message authentication tag is encrypted.

At the receiver side the included message authentication tag is decrypted and compared to the code that is calculated over the received message, same way as it was on the sender side. When the difference is below the given threshold, then the message is authenticated, otherwise the error audit message "AUTHENTICATION FAILURE" MUST be returned.

The calculation of the message authentication tag:

Authentication tag =

E_k1(H_k2(SRTP header||ROC) + AA_k3(SRTP payload))),

where E is an encryption algorithm using a block cipher, H is a cryptographic hash algorithm and AA is the approximate authentication algorithm. Their keys are k1, k2 and k3 respectively. These subkeys are generated using the negotiated authentication key.

The acceptance threshold parameter extends SRTP context. The value of the parameter SHOULD be decided during the negotiation of other parameters.

3.1. SRTP headers and context

In the SRTP message the authentication covers the header and the payload. The header section of the SRTP packet is considered as highly error sensitive. Furthermore, a possible attack against sequence number and other fields makes the header sensitive from security point of view as well. Thus, the SRTP header part requires perfect authentication, so it MUST be authenticated using a cryptographic hash function (e.g. HMAC-SHA1 [RFC2104]).

To perform the header authentication, a cryptographic hash function MUST be used. When the size of the message authentication tag (n_tag) is shorter than the output of the hash function, then only the leftmost n_tag bit SHOULD be considered. Otherwise, when the message authentication tag is longer than the hash output, then the hash value SHOULD be repeated along the length of the message authentication tag.

According to RFC3711, in the case of SRTP, the authentication should cover the ROC value from the cryptographic context as well. Since ROC is a sensitive data in terms of security, it requires perfect authentication. For this reason the ROC is concatenated to the SRTP header and the cryptographic hash function covers them both.

3.2. SRTP payload encryption

The approximate authentication of the message allows the successful authentication of the packet regardless a small number of bit errors in the payload. In order to circumvent content spoofing attacks, it is RECOMMENDED to encrypt the content. When the content is encrypted, the encryption algorithm MUST be a stream cipher. In other cases, due to the significant change caused by the avalanche effect, using approximate authentication is meaningless.

Since the encrypted payload use a stream cipher for the encryption procedure, there is no need for padding. Due to the nature of approximate authentication, as padding length can be spoofed, padding is risky. In fact, RTP padding MUST be disabled, so the packet SHOULD NOT contain any RTP padding or RTP pad count fields.

3.3. SRTP payload approximate authentication

The approximate authentication algorithm creates a checksum like authentication code. Using the code of the transmitted payload and the code calculated at the sender side, it is possible to deduce to the amount of modifications in the payload. The approximate authentication is not necessarily an error detection code, therefore it may happen that the exact number of differences cannot be signaled. In this situation the code gives an approximate value only, hence the name is approximate authentication.

Calculating the approximate authentication code MAY require the need of ROC and SEQ from the cryptographic context. Together with these values and the key associated to the approximate authentication, it is possible to perform packet specific cryptographic operations safely.

3.4. Message authentication tag encryption

The encryption of the message authentication tag is necessary, since the approximate authentication field might be subject of spoofing attacks. The encryption procedure can provide protection against spoofing. Stream ciphers are inappropriate for this protection, since this way the attacker can perform predictable changes in the decrypted authentication tag. The encryption algorithm used here MUST be a block cipher.

It is possible that the length of the message authentication tag is not the multiple of the block size, which the block cipher defines. Block ciphers having a block size larger than the message authentication tag length MUST NOT be used. The output of the block

cipher cannot be cut, just like hash outputs. Otherwise, there are two options. When the length of the message authentication tag is exactly the same as the block size, then any block cipher can be used. When the length of the message authentication tag is greater than the block size, then the Ciphertext stealing (CTS) operation mode [CTS] MUST be used. Using the CTS method, the size of the message authentication tag should not be aligned to the block size.

For short message authentication tags, it is RECOMMENDED to select a block cipher that has 64 bit block size (e.g. BLOWFISH [BF]).

During the transmission the message authentication tag may suffer bit error damage. Due to the avalanche effect in block ciphers, the authentication will fail in this case.

3.5. Key derivation for the algorithms

The encryption algorithm, used to encrypt the message authentication tag, the cryptographic hash function, used to protect the SRTP header and the approximate authentication algorithm, used to protect the SRTP payload require keys. The encryption algorithm MUST have a key, while at the hash function and at the approximate authentication, keys are OPTIONAL only. Despite these algorithms may work without a key, keyed algorithms are RECOMMENDED increasing the security. These three keys are called subkeys and managed by algorithms described in SRTP.

The subkeys MUST be generated using the authentication key (the k_a parameter in RFC3711) via the key derivation method defined in RFC3711. The key derivation MUST use the AES-CM PRF, which is mandatory to implement in SRTP. There are new labels defined for the new keys. For the encryption key the <label> = 0x06, for the hash function <label> = 0x07 and for the approximate authentication <label> = 0x08.

The length of the subkeys should be defined together with the particular approximate authentication algorithm.

4. SRTCP message authentication

The approximate authentication is valid only for SRTP messages. SRTCP SHOULD NOT use approximate authentication. In the SRTCP messages there are no data that could be used with bit damages. For this reason approximate authentication is not applicable.

SRTCP is out of scope of this document.

5. Security Considerations

The security considerations in RFC3711 apply to this document as well.

Section 9.5 of RFC3711 considers weak authentication in terms of security. The approximate authentication can be considered as a weak authentication, since due to its nature, it authenticates modified messages. It is impossible to get know whether the source of the modification is an adversary or it is a natural error.

SRTP MAY be used with weak authentication, where it is an acceptable security risk, and it is impractical to provide strong message authentication. The risks associated with exercising the weak authentication need to be considered by a security audit prior to its use for a particular application or environment. Further details can be found in RFC3711.

6. IANA Considerations

SRTP uses cryptographic transforms which a key management protocol signals. It is the task of each such protocol to register the cryptographic transforms or suites of transforms with IANA. This draft defines no new cryptographic transforms or suites of transforms, therefore IANA considerations can be omitted.

IANA will register new crypto suites into the subregistry for SRTP crypto suites, when a particular approximate authentication algorithm will be proposed.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

[CTS] Schneier, Bruce, "Applied Cryptography", Second Edition, John Wiley and Sons, New York, 1996. Errata: on page 195, line 13, the reference number should be [402].

7.2. Informative References

[RFC2104] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.

[3GPP] "Technical Specification Group Services and System Aspects; Quality of Service (QoS) concept and architecture", TS 23.107 V5.9.0, Technical Specification 3rd Generation Partnership Project, June 2003.

[BF] B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)", Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993), Springer-Verlag, 1994, pp. 191-204.

[DONGVU] Dongvu Tonien, Reihaneh Safavi-Naini, Peter Nickolas, Yvo Desmedt, "Unconditionally Secure Approximate Message Authentication", IWCC, 2009, pp. 233-247

[FEHER] Gabor Feher, Istvan Olah, "Enhancing wireless video streaming using lightweight approximate authentication", Multimedia Systems Journal (MMS), 2008, Vol 14(3), pp. 167-177

[GRAVISH] Liehua Xie, Gonzalo R. Arce, R. F. Graveman, "Approximate image message authentication codes", IEEE Transactions on Multimedia (TMM), 2001, Vol. 3(2), pp. 242-252

8. Acknowledgments

This draft is based on the experience gained in the OPTIMIX research project. The research leading to these results has received funding from the European Union’s Seventh Framework Programme ([FP7/2007-2013]) under grant agreement no ICT-214625.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Gabor Feher
Budapest University of Technology and Economics
H-1117 Budapest, Magyar tudosok krt. 2., HUNGARY
Email: feher10@tmit.bme.hu