Communications Link Layer Security

Dr. Kwei Tu, Communications System Consultant
Ktu1@houston.rr.com
Houston, Texas USA

Abstract

A highly robust secure protocol is proposed to provide communications link layer security including data source authentication, data integrity, data confidentiality, and replay attack resistance.

The protocol is based on shared secret data (SSD) and time codes between two communications entities. The Keyed Hash Message Authentication Code with Secure Hash Algorithm -2 (HMAC-SHA-2) is employed for data source authentication, data integrity and session key generation, while Advanced Encryption Standard (AES) provides the data confidentiality. The Crypto Sync which is already required for encryption/decryption of data in a secure link layer transfer frame operating in cipher feedback mode (CFB) as the initialization vector is also utilized as the Message Authentication Code (MAC) and session key generator and distributor. No additional overhead is required and the link channel bandwidth remains the same.

A link layer Consultative Committee on Space Data System (CCSDS) data frame structure is given as the example to illustrate the protocol.

The proposed protocol can be applied to any encrypted link layer transfer frame between any two communications entities.

1. Introduction

The main objectives of communications link security are to provide:

- Authentication
  Data is from an authorized source.
- Validation/Integrity
  Data that is intended for receiving system, is valid, and hasn't been tampered with by an unauthorized entity. Replay attack will be rejected.
- Confidentiality
  Data is only accessible by authorized user.

Data confidentiality is normally achieved by using symmetric secret key system, since they can operate in high speed. The Advanced Encryption Standard (AES) with 128-bit key length [1] has become the US encryption standard to replace the aging Data Encryption Standard (DES) since 2003. AES is an open encryption algorithm that has now been adopted as the encryption standard in the IEEE 802.11 wireless local area network (WLAN) [2] and is expected to be implemented in all the future commercial networks including Internet, mobile phones, satellite links, etc. AES operating in cipher feed-back mode (CFB) is particularly considered to be secure for data transmission [3].

Data source authentication and data integrity can be achieved by employing the keyed-hash message authentication code (HMAC) [4] due to its high speed operation.

Link layer protection starts with an authentication service and includes link layer encryption and integrity services. As a result, only authenticated users can actively use the link layer, and all data traffic on the link layer is encrypted and authenticated. Link layer sends blocks of data called "transfer frames" with the necessary frames identification, error control, and flow control.

The proposed protocol will be based on the shared secret data (SSD), time code, HMAC and AES. The SSD has been adopted by mobile communications industry for the cellular phone networks [5].

2. The Proposed Protocol

The protocol starts with formation of Message Authentication Code (MAC) for the data blocks in term of Crypto Sync, and session key generation from the Crypto Sync. It follows by data encryption and decryption processes operating in AES cipher feedback mode (CFB). Finally the protocol ends in MAC verification for data source authentication, and data block integrity check.

Before the proposed protocol is presented, the data format for the transfer frame at link layer will be discussed first.

2.1 Data Frame Format

To illustrate the proposed protocol, a data transfer frame between the two communications entities at the link layer will be assumed. In general, the data transfer frame is typically composed of a header, data zone and parity checks if block coding is employed. The whole data transfer frame may also be convolutionally encoded prior to being modulated on a radio frequency (RF) carrier at the physical layer.

A CCSDS (Consultative Committee on Space Data System) data format [6] is given as an example for this paper to illustrate the proposed protocol. The proposed protocol can be applied to any data frame structure. In this CCSDS data format, various data sources like video, voice, command and control, and data uploads can be accommodated dynamically. Figure 1 shows a typical CCSDS format with the following parameters as an example:

Frame length: 256 bytes (2048 bits)
Sync mark: 4 bytes
Header: 6 bytes Insert zone: 16 bytes (Crypto Sync) Data zone: 214 bytes Reed-Solomon parity checks: 16 bytes

A forward error correcting (FEC) Reed-Solomon (R-S) (255, 239) code shorten to R-S (252, 236) is given as the example. The insert zone is normally reserved for encryption/decryption synchronization and is called **Crypto Sync**. It serves as the encryption/decryption initialization vector in the CFB mode and the **Crypto Sync will play a vital role for the proposed protocol.**

2.2 Network Initialization

Before the network is operational, during the initialization phase, a unique Shared Secret Data (SSD) will be generated and distributed between two communications entities. A 128-bit SSD is assumed for this paper. Each entity will then be preloaded with this unique SSD. There are several ways to securely distribute SSDs in the initialization phase. It is beyond the scope of this paper to address the SSD distributions. In addition, the time code (GMT) must be synchronized in the initialization phase between these two entities and must be updated and checked periodically during the operation.

2.3 The Process

For each data frame in a communication session from one communications entity to the other:

1. The sending station generates a 128-bit Massage Authentication Code (MAC) or **Crypto Sync**. This crypto sync will be generated as the truncated output of the Keyed-Hash Massage Authentication Code (HMAC) [4] with Secure Hash Algorithm-2 (SHA-2)* [7] as the hashing function (HMAC-SHA-2). The input to the HMAC-SHA-2 is the plain text data blocks and header (total 220 bytes) with the SSD (16 bytes) and GMT time code (4 bytes) as the key.
2. This **Crypto Sync** and SSD will then be used to generate the data encryption key as the truncated 128-bit output of the HMAC-SHA-2 with the 128-bit Crypto Sync (MAC) as the input and the sending station 128-bit Shared Secret Data as the key.
3. The data blocks in the data zone will be AES encrypted with this key and the **Crypto Sync** will be used as the initialization vector for Cipher Feedback (CFB) encryption process.
4. The **Crypto Sync** in clear text will be filled in the insert zone and the encrypted data blocks in the data zone along with the clear text header will then be R-S encoded to form the transfer data frame by a data frame formatter.
5. The receiving station data frame deformatter will process the received data frame and recover the header, Crypto Sync and encrypted data blocks.
6. The receiving station will then use received **Crypto Sync** and receiving station SSD to regenerate the data frame session key as the truncated 128-bit output of the HMAC-SHA-2 and AES decrypts the data blocks in the data zone. It should be noted that the Crypto Sync carries the secret session key information and Crypto Sync will be different for each data frame due to different data contents and different time codes.
7. The receiving station will then regenerate a **Crypto Sync** based on the decrypted data blocks, GMT time code of receiving station, and SSD. The data integrity will then be verified against the received Crypto Sync. Data source authentication and data integrity, and confidentiality are assured if the crypto sync comparison is confirmed.

**Can be SHA-224, SHA-256, SHA-384, or SHA-512*

Figure 2 shows the whole process. It should be noted that each data frame is AES encrypted with a unique data frame session key.

3. Protocol Features

The Crypto Sync plays a vital role for the proposed protocol. It serves not only as the initialization vector for data CFB encryption/decryption process, but also as the session key generator and the digital signature for the data blocks. The Crypto Sync will be different for each data frame due to different data contents and time codes. **Since the Crypto Sync is required for CFB operation anyway, no additional overhead is required for providing additional authentication function and the link channel bandwidth remains the same.**

If the received Crypto Sync is matched with the regenerated Crypto Sync on the receiving side, the main objectives of link layer security will be achieved:

Authentication
Header and data is assured from an authorized source (Only sending station knows the SSD.)

Validation/Integrity
Header and data that is intended for receiving system, is valid, and hasn't been tampered with by an unauthorized entity (Only sending station knows the SSD and HMAC-SHA-2 provides necessary security.)

Confidentiality
Data is only accessible by authorized user (Only receiving stations knows SSD and session key and AES provides necessary security.)

The proposed protocol will also repel replay attack. If the adversary replays the intercepted transfer data frame, the timing codes will not be current and thus the Crypto Sync cannot be verified and data frame header and data blocks cannot be authenticated and will thus be rejected.

The security level for the encrypted data blocks is assured due to AES algorithm. It should be noted that the session key is changed and different for each data frame due to different Crypto Sync for each data frame.

If data blocks must be transmitted in clear under certain emergency situation, the proposed protocol based on HMAC-SHA-2 can still provide adequate protection for the data authentication and integrity

The security of the protocol is dependent on the secrecy of the SSD. SSD is equivalent to the master key and its crypto life can be longer. After the first communications session, the shared secret data (SSD) should be freshed or changed periodically through a secured session. The length of the SSD can also be extended to increase the security. For the satellite or spacecraft control, enough SSDs can also be stored onboard and SSD can be changed periodically.

4. Summary and Conclusions

The proposed protocol will provide robust security for the communications between the two communications entities at the link layer and will also provide a secure link even if the link is operated in the clear mode.

AES will provide a high security level of data confidentiality and HMAC-SHA-2 in conjunction with Shared Secret Data (SSD) and the time code will provide secure MAC for data frame header and data authentication and integrity.

The Crypto Sync field is the key for the proposed protocol. It plays triple roles: one as the initialization vector for encryption - decryption process and one as the MAC and the other as the session key generator and distributor.

No additional data bandwidth overhead is needed for the transfer data frame and thus the link message throughput is not affected and the data interfaces will also not be affected. The Shared Secret Data (SSD) must be kept secret and protected and should be changed periodically through the secure session after the initialization phase.

5. References

[1] Advanced Encryption Standard (AES), Federal Information Processing Standards Publications (FIPS PUBS) - 197, National Institute of Standards and Technology (NIST), November 2001.

[2] IEEE Std. 802.11i-2004, July 2004

[3] DES Mode of Operation, FIPS PUB 81

[4] The Keyed-Hash Message Authentication Code (HMAC), FIPS PUB 198, March 2002
[5] TIA IS-41C, Telecommunications Industry Association, 1995
[6] Recommendation for Space Data System Standards: Advanced Orbiting Systems, Networks and Data Links, CCSDS 701.0-B-2, Red Book, Issue 2, Washington, D. C. November 1992
[7] Secure Hash Standard, FIPS PUB 180-2, February 2004

Figure 1 A CCSDS Data Transfer Frame

<table>
<tr><td colspan="5">Channel Access Data Unit (CADU) - 256 bytes</td></tr>
<tr><td></td><td colspan="4">Coded Virtual Channel Data Unit (CVCDU) – 252 bytes</td></tr>
<tr><td>Sync Mark
4 bytes</td><td>Header
6 bytes</td><td>Insert Zone
(Crypto Sync)
16 bytes</td><td>Data Zone (Encrypted or Decrypted)
214 bytes</td><td>RS Parity
Check16 bytes</td></tr>
<tr><td></td><td colspan="3">Virtual Channel Data Unit (VCDU) - 236 bytes</td><td></td></tr>
</table>

Figure 2 Protocol Process