FLIGHT TERMINATION COMMAND AUTHENTICATION USING BLOCK ENCRYPTION

Dennis Arce
Bourne Technologies, Inc.

ABSTRACT

Next generation flight termination systems (FTSs) will use digital technologies to verify the authenticity of range safety commands by command receiver-decoders located on each vehicle. This paper will discuss the general principles behind simplex message authentication using a block encryption cipher, and presents examples for demonstration.

KEY WORDS

Range Safety, Flight Termination, Authentication, Encryption, Block Cipher

BACKGROUND

Flight termination systems (FTSs) have been used since the late 1950s to protect the general public from unmanned vehicles that leave their controlled airspace. Current FTSs support a variety of different commands including the traditional ARM/TERMINATE sequence, and non-standard vehicle controlling commands. The activation of any of these commands overrides the vehicle's primary command link (or command process, as applicable) in such a way as to provide for the general safety of the community in the surrounding geographic region in which the vehicle is operating.

Recent events have caused the range safety community to assess the security requirements for this data link and revise them as necessary[1]. The findings of this assessment were that all FTSs should provide a *reasonable* amount of assurance that the link cannot be "spoofed," or improperly commanded by an unauthorized entity. ***Authentication*** is the process of determining that the command was sent by an entity from which the receiver is authorized to accept.

There are currently two types of flight termination data links that are in wide use at United States Ranges: *Inter-Range Instrumentation Group* (*IRIG) Tone* systems and *Secure* systems. Each of these data links uses sequences of tones over a simplex FM communications link to form their

[1] Steven G. Cronk, Maria A. Tobin, and Robert D. Sakahara, "Enhanced Flight Termination Study- Overview and Status," Proceeding on International Telemetering Conference 2001, International Foundation for Telemetering, 2001, pp. 297-403.

commands. The *IRIG Tone s*ystems use a rudimentary tone sequencing capability to provide responsive control for up to five simultaneous vehicles. The *Secure* system uses a very large number of tone combinations such that it is very difficult to guess the pattern that the vehicle is expecting for any specific command. However, other attributes of the *Secure* data link (e.g. number of vehicles supported in flight) lead to the implementation of a data link that supports security. An optimum system would provide this security and the attributes of the existing *IRIG Tone* systems that allow it to support multiple vehicles missions.

A major attribute of existing FTS architectures is that they use a simplex communications link. The command is sent from the transmitter to the vehicle with a one-way communications link. While most vehicles provide informational feedback about this link via telemetry, this feedback is not required for flight termination commands to be carried out on the vehicle. This is done because Range Safety components are generally required to be designed and tested to environmental requirements exceeding those of the rest of the components on the vehicle. In this way, the flight termination system is expected to work when all else fails on the vehicle. However, this adds both a technical and economic burden on the manufacturers of these components. Thus, future flight termination systems will most likely use a simplex communications link.

This paper discusses the implementation of a digital command link with commercial-off-the-shelf (COTS) encryption algorithms to provide greater flexibility in commanding the vehicle, while still providing a large degree of command authentication.

BLOCK ENCRYPTION BASICS

Encryption can be seen as a method of scrambling information in order to hide it. The methods used to hide the information make it difficult for an unauthorized entity to gain access to it, and easy for an authorized user to decode the information into its original form. The encryption 'key' is the secret shared by all authorized entities; it allows them to properly encrypt and decrypt messages. Since authorized senders and receivers are the only ones that know the "key," decryption by an unauthorized entity is difficult. The same is true for unauthorized senders, or 'spoofers,' of the message.

Block encryption or block cipher[2] encryption is a type of encryption where there is a one-to-one mapping between an unencrypted group of bits and an encrypted group of bits. For any specific code-key a specific unencrypted message will always result in the same encrypted message. This is true during both the encryption process and during the decryption process. Furthermore, the algorithms presented in this paper contain the same amount of unencrypted message bits as encrypted message bits. Therefore, the encrypted and unencrypted data streams have the same bit rate. These attributes are depicted by an example in Figure 1.0. This also means that any input to the receiver will cause an output. Determining whether this output is a valid message is the primary purpose of the authentication mechanism discussed herein.

---

[2] Bruce Schneider, *Applied Cryptography, Second Edition: protocols, algorithms, and source code in C*, John Wiley & Sons, Inc, 1996, p.189.

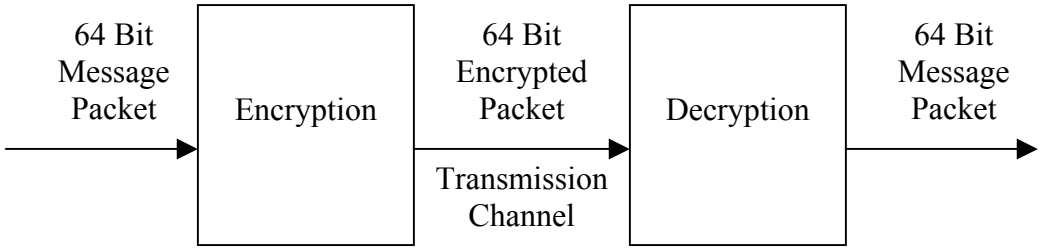

Figure 1.0 Basic Encryption and Decryption using a 64 Bit Block Cipher Algorithm

Another attribute of any practical form of block encryption is its ability to *diffuse*[3] the information. That is, the bits should appear to be random when in the encrypted state. Therefore, a change of one bit in the unencrypted state can cause the change in all the bits in the encrypted state, and vice-versa. On the average, a change in one unencrypted bit will cause a change in 50 percent of the encrypted bits.

MESSAGE AUTHENTICATION

Since a receiver with a decryption circuit will decrypt any block of bits to an unencrypted block, an authentication process will be employed to assure the transmitting authority of the message. This is done by fixing a number of bits to known or expected values in fields in the unencrypted message format. If the encrypted message contains those fields as expected, the probability that the message is in its original form can be determined through simple math. The number of fixed fields in the message is directly related to the certainty that the message is in its original form. However, since these bits must be fixed or expected, the amount of data sent in this packet is inversely related to the length of the expected bits. The derivation of this certainty will be described by example and extrapolation:

Given the following system:

1. Message is a 5-bit unencrypted word broken up into the following fields.

   Bit 0 is always 1
   Bit 1 is always 0
   Bits 3-2 are random data [4 values]
   Bit 4 is unused [don't care]

[3] H.X Mel and Doris Baker, *Cryptography Decrypted*, Addison-Wesley, 2000, p. 29.

2. Given a mapping of block encryption scheme as follows:

Note: The bits are aligned by position 43210.

Message – Encrypted	Message – Encrypted	Message – Encrypted	Message – Encrypted
00000 -10000	01000 -10110	10000 -01111	11000 -11010
00001 -00000	**01001 -00001**	**10001 -10111**	**11001 -11011**
00010 -00111	01010 -01000	10010 -01100	11010 -11110
00011 -10001	01011 -00010	10011 -00100	11011 -00101
00100 -10010	01100 -10101	10100 -11000	11100 -01101
00101 -10011	**01101 -01001**	**10101 -00011**	**11101 -11111**
00110 -10100	01110 -01010	10110 -01110	11110 -00110
00111 -11001	01111 -01011	10111 -11100	11111 -11101

3. Note that only eight different values would be sent, those with the suffix 01. Therefore, if any of the bits are changed in transit there is a probability that it would not be authenticated at the receiver. If the message 00001 is to be sent, it is encrypted to 00000. If a bit error occurs and it is received as 01000 the receiver will decrypt the value to 101**10**, which is not a valid message because the 01 suffix does not exist in it. That is why this type of authentication can be seen as a form of error detection.

4. Also note, that because the '4' position is a 'don't care' that there are eight values that can be accepted, while there are only four unique commands: Message 0**00**01 and 1**00**01 are authenticated as the same command.

5. Note also that if someone is trying to guess a correct value by placing random data into the receiver, there are only 8 of 32 values that are acceptable. This is because all values that have the 01 suffix are acceptable in this situation. This information can be extrapolated to the following formulae.

Given:
n = Message bit length [e.g., 5 bits]
k = fixed or expected bit length [e.g., 2]
m = command bit length [e.g., 2]
r = redundant bit length = n-(k+m) [e.g., 1]

$$\text{Total Valid Commands} = 2^{(\text{cmd len} + \text{rdt len})} = 2^{m+r} = 2^{m+n-k-m} = 2^{n-k} \quad (1)$$

The probability of guessing any acceptable command= P_{any}

$$\frac{\text{Total Valid Commands}}{\text{Total Messages}} = \frac{2^{n-k}}{2^{n}} = 2^{n-k-n} = \frac{1}{2^{k}} \quad (2)$$

The probability of guessing a specific command= $P_{specific}$

$$\frac{P_{any}}{\text{Unique Commands}} = \frac{\frac{1}{2^{k}}}{2^{m}} = \frac{1}{2^{k+m}} \quad (3)$$

For our example above:

- From (2), the Probability of guessing any command = $1 / 2^{2}$ = 1 chance in 4.
- From (3), the probability of guessing a specific command = $1 / 2^{2+2}$ = 1 chance in 16.

These probabilities are very high for our example, but as we increase the number of fixed or expected bits, the probabilities become quite low. A 64-bit encrypted packet containing 40 bits of fixed or expected values and a 6-bit command field would yield a probability of 1 chance in 2^{40} (= 1.1×10^{12}) tries that a randomly generated command would be authenticated as any message, and 1 chance in 2^{46} (= 7.0×10^{13}) to guess a specific command (e.g. ARM). In real world terms, this would equate to an average of 1 correctly guessed command every 697 years, for a system that sent 50 messages per second. Each specific command would be guessed every 44,600 years on average.

REDUNDANCY IN AUTHENTICATION FIELDS

There are situations where the fixed or authenticated bits are separated into fields. For cases where multiple values are acceptable for a field, the impact of the *fixed or expected bit length* variable will be diminished in the equations above. The amount that the variable will be diminished using the equation:

$$k' = k - \text{Log}_2(\text{Acceptable Values}) \quad (4)$$

Where
k' = effective k due to redundancy in authentication.

For example, assume the 64-bit system described above with 40 *fixed or expected bits* is divided into fields such as Vehicle ID and Transmitter ID. Assume that the receiver/decoder will accept only one Vehicle ID, but any of four unique Transmitter IDs. The effective k for this example is $40\text{-}\log_2(4) = 38$. A receiver decoder that accepted five unique Transmitter IDs would have an effective k of $40\text{-}\log_2(5) = 37.68$. The probability of randomly guessing any command for such a system would be:

From (2) above: $1/2^{37.68} = 1$ chance in 2.2×10^{11}

RECORD AND REPLAY ATTACK

Encrypting and authenticating do not mitigate the ability of an unauthorized entity from recording a command and replaying the command at an inopportune time. (For example, a "Terminate" command could be recorded during pre-flight checks, then played back during actual launch.) Therefore, many systems employ a message numbering scheme that synchronizes the sender and receiver. For this application, an appropriate message numbering scheme includes a command counter, and 'greater-than or equal to' authentication.

The process works as follows. The system uses a combination of both the *fixed or expected bits* and an additional field that contains a command counter value. Each time a different command is sent, the transmitter's command counter is incremented. The receiver authenticates the command based on the *fixed or expected bits*, and then determines if the command counter value received is 'equal to or greater than' the value of the last authenticated message. If the value is 'greater-than or equal to' the previous value, the command is authenticated. The receiver stores the last authenticated command counter value. To clarify this approach, the transmitter's counter is only incremented when a new command is sent, not on each subsequent frame. Thus, large counter values are not necessary unless a large number of unique commands will be sent.

CONCLUSIONS

Digital flight termination systems that use block encryption can authenticate the sender of the message thus providing mitigation to random attacks and errors in the transmission medium. The protection this approach provides can be calculated using simple math, and is related directly to the size of the authentication fields, while taking into account any intrinsic redundancies in those fields.