ANNEX T STANAG 5066 TRANSEC CRYPTO SUBLAYER USING AES AND OTHER PROTOCOLS (Optional)

The TRANSEC Crypto Sublayer is an optional sublayer that provides TRANSEC. It defines a framework for using arbitrary stream cryptography to provide TRANSEC. It specifies how to use Advanced Encryption Standard (AES) in this framework.

The TRANSEC specified in Annex T can be used as an alternative to synchronous serial crypto devices connected to the DTS using STANAG 5066 Annex D.

This annex is new to Edition 4 of STANAG 5066.

T.1. Overview

This annex specifies protocol for supporting a Crypto Layer between STANAG 5066 and Modem, following the STANAG 5066 model. This protocol defines a generic framework for use with different encryption algorithms, with a specific mapping for AES encryption.

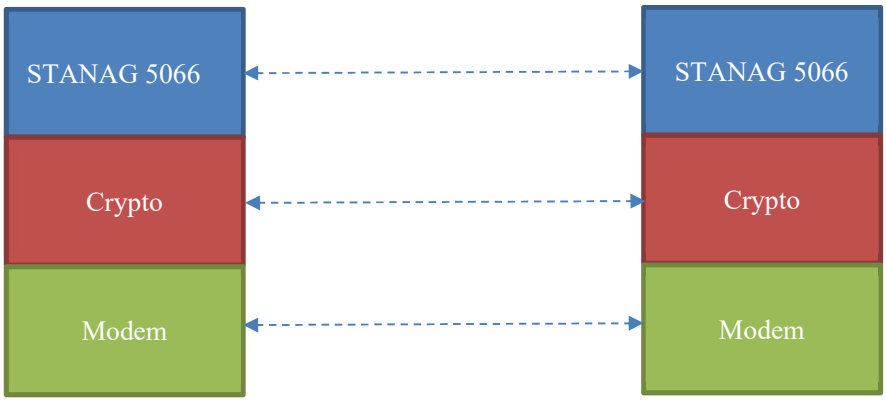

Figure T-1: STANAG 5066 TRANSEC Crypto Architecture

The STANAG 5066 architecture for use of TRANSEC Crypto is shown in Figure T-1. Crypto is used immediately above the modem, which is the lowest possible point that Crypto could be inserted. A stream crypto is used, so that modem data errors only impact that data. When crypto is synchronized, it does not introduce additional errors. This is an efficient approach.

HF traffic is easy to monitor and even the lowest layers of STANAG 5066 contain information that is of potential interest. Therefore, it makes good security sense to perform encryption at this lowest possible layer.

T.1.1. **Benefits of a Protocol Approach**

This specification introduces a protocol approach, which gives the benefits of the architecture and removes the overheads and issues associated with use of sync serial following Annex D. A number of implementation approaches are possible. An implementation approach that could be taken is:

1. Use the open TCP protocol interface specified in MIL-STD-188-110D Annex A to communicate between Crypto Layer and modem.
2. Implement the Crypto Layer framing as specified here, as part of the STANAG 5066 server. This can offer:
    a. Built in AES support.
    b. Plugin options to drive other Crypto.

T.2. Crypto Considerations

This section considers a number of crypto issues.

T.2.1. **AES**

AES (Advanced Encryption Standard) is a widely adopted US Government standard for encryption. It is widely used for commercial, government and military operation. Therefore, use of AES is specified in this annex.

T.2.2. **COMSEC and TRANSEC**

TRANSEC (Transmission Security) is the protection applied at the lowest communication level, of which Crypto between modem and STANAG 5066 is a good example. Technically, this annex specifies TRANSEC.

COMSEC (Communication Security) is the protection applied to user data. TRANSEC **may** be used to provide COMSEC. The model for use of Annex D is that COMSEC is provided by TRANSEC. This annex defines TRANSEC. It is a deployment decision as to whether this is also used for COMSEC.

T.3. Modem Service Specification

In order to understand how the Crypto Layer works, it is important consider the modem service interface.

T.3.1. **Modem Transmission**

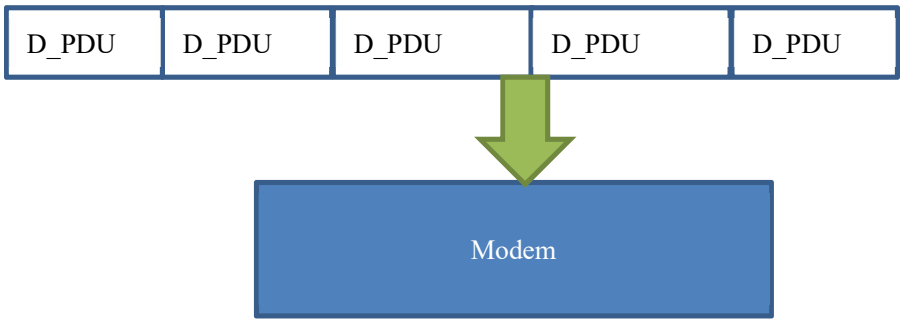

Figure T-2: Transmission of D_PDUs to a Modem

STANAG 5066 will transmit over the modem a “bounded stream” of D_PDUs as shown in Figure T-2. This is sent as a sequence of bytes, with request to start transmission implicit in the first byte. When the last byte of the last D_PDU is sent, it is marked as “end of stream”. The modem may pad after the last D_PDU in order to fill an exact number of blocks.

STANAG 5066 will also set parameters for the transmission, which will be fixed for the period of transmission and will deal with any errors reported. Parameters include: speed; interleaver; waveform; waveform-specific parameters; bandwidth.

For duplex and broadcast, transmissions may be of arbitrary length. In other cases, transmissions are limited to 127.5 seconds and each D_PDU is marked with remaining transmission time (EOT) in units of 0.5 seconds.

This information is transparent to the modem, but important for overall operation. STANAG 5066 will know the speed of modem in order to calculate data to send and to keep up with the modem. Transmission length is determined as start of transmission. However, it is desirable to defer choice of which data to send, to enable insertion of high priority data arriving after a transmission starts.

T.3.2. **Modem Reception**

Under good conditions, modem reception of data will be entirely symmetrical to transmission, and a bounded stream of D_PDUs will be provide to the receiving server. On reception with modern waveforms, most transmission parameters are determined from the transmission. Dealing with modem reception issues is key to Crypto Layer design.

T.3.3. **Determining Transmission End**

Data is often lost or corrupted during transmission. A key problem is to determine end of transmission. There are strict rules for modem transmission. A receiving modem

needs to apply heuristics to determine transmission end, particularly with poor HF conditions or aggressive choice of transmission speed.

There are two modem level mechanisms for determining end of transmission:

1. Modem Protocol. This is supported in STANAG 5069, but not older protocols. It definitively marks end of transmission.
2. EOM Marker. Two special bytes sent in the data stream. Care needs to be taken to handle the case where the "real data" includes this value. Heuristics to validate include ensuring that only "padding data" follows the EOM and that the RF signal falls off after the block is complete.

Both of these mechanisms can be "lost" due to fading or other data corruption. A modem will detect loss of RF signal. This can be an indication of transmission end or it could be a reception gap in a longer transmission. A modem will generally wait for a period before considering the transmission ended due to loss of RF. During this time, the modem will continue to send data to STANAG 5066 at "modem speed" and will maintain synchronization with the transmission (which may have ended).

The D_PDU EOT mechanism is helpful when the modem continues to receive in this way. A STANAG 5066 server can determine the end of transmission time from any single D_PDU. It will "know" that a transmission has finished, even when the modem continues to provide data (e.g., because Modem has not explicitly detected end of the RF transmission and is maintaining sync). The DTS will be able to switch to transmission, while the modem is still sending it (spurious) data.

T.3.4. **Modem Synchronizing During Transmission**

HF Waveforms start with a robust synchronization sequence, and so normal expectation is that modems synchronize at start of transmission, so that data bytes are correctly aligned over the transmission, even when there is loss and corruption.

Modem protocols can resynchronize during a transmission. This means that data can start to be received part way through a transmission. Also, a transmission can be "lost" and then a latter part of the transmission picked up as a new transmission.

Some waveforms, notably STANAG 4539 and STANAG 4285, synchronize very well during when the initial synchronization fails and it is desirable to allow for this on reception.

STANAG 5069 does not synchronize well during a transmission, if the initial synchronization fails. This means that when STANAG 5069 is used, it may be desirable to use longer (and more robust) pre-ambles and **may** be desirable to avoid overly long transmissions.

T.4. Crypto Layer

T.4.1. Service Interfaces

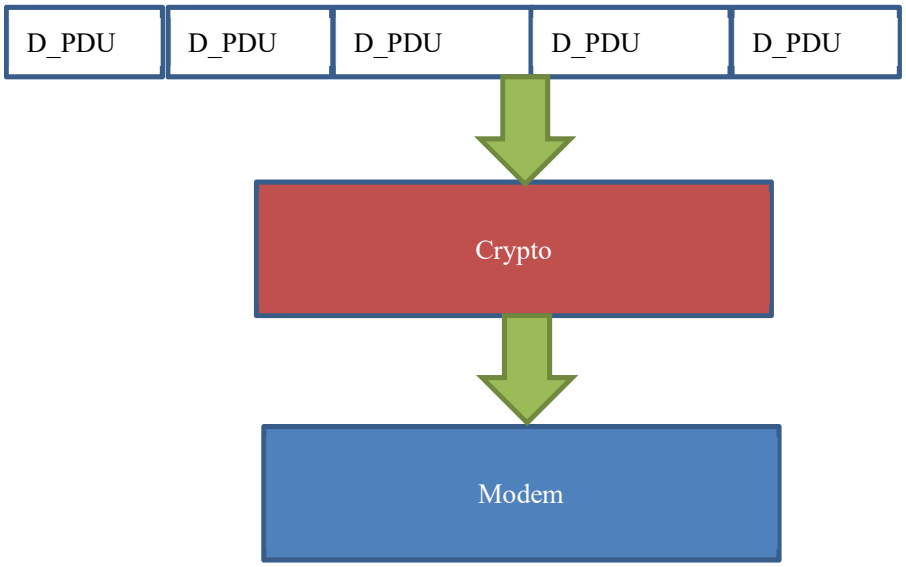

Figure T-3: STANAG 5066 Interface to Crypto

The basic service model of this protocol is that the data interface to and from the crypto layer is essentially the same and shown in Figure T-3. This is in line with STANAG 5066 Annex D.

STANAG 5066 needs to be aware of the protocol overheads of the Crypto Layer, so that it can correctly calculate transmission lengths. It **may** also choose to align D_PDU boundaries to modem block boundaries, which can be done precisely with this protocol stack.

T.4.2. Crypto Mapping & Counter Mode

The Crypto needs to operate as a stream crypto. Most modern encryption algorithms are block-oriented, which cannot be used directly. This is addressed with Counter (CTR) Mode, which is a mechanism to provide stream encryption from a block cipher. This is now widely recognized as a secure approach.

Counter mode works by initializing the cipher with an Initialization Vector (IV) and/or a Nonce. This then generates a crypto stream of bytes with as many bytes as needed. Sender and Receiver share the IV/Nonce and so can generate identical crypto streams.

The sender will XOR the data stream with the crypto stream to produce a transmission stream, which is sent between the modems. The receiver can then XOR the received

transmission stream with the crypto stream to restore the data stream sent by the peer STANAG 5066 server. This received stream may have data corruption due to modem level HF errors.

The strict synchronization of modem data transfer ensures that the streams remain aligned over the complete transmission.

T.4.3. Crypto Initialization

This section considers crypto initialization in more detail.

T.4.3.1. General Model

There is information which needs to be shared between sender and receiver which is configured prior to the transmission. This might be an external mechanism such as pre-placed keys, or other external mechanism.

Then there is information provided for each transmission, which will typically be an IV and/or Nonce, but there could be other information for encryption mechanisms other than AES.

T.4.3.2. AES Initialization

Each transmission will provide a 2 byte reference to the AES Key/Nonce pair. This reference enables:

- Different keys may be used for different pairs of 1:1 communication and multicast/broadcast groups.
- New keys can easily be used in the event of key compromise.
- Giving keys a limited lifetime, with migration to new keys.

Management of this reference and distribution of AES Keys and Nonces is not specified in this annex.

Each transmission will include an 8 byte IV that is unique for the AES Key/Nonce pair. Uniqueness can be ensured by simple incrementing or by Linear Feedback Shift Register. The last IV used should be recorded on permanent storage, so the unique IV will be ensured in the event of restart.

T.4.3.3. Resilience over HF

Transferring the initialization information needs to address potential corruption on transfer. HF errors tend to cluster, so the best approach to resilience is to repeat the information at intervals in the data stream.

The protocol also needs to address the possibility of the initial modem synchronization not happening. This is done by use of extended information that includes:

1. A Maury-Styles two byte pair, that enables synchronization.
2. A counter to enable the amount of preceding data to be calculated. This will enable to full crypto-stream to be determined, so that the received data can be XOR'd from the correct point.

T.5. Crypto Layer Protocol

The Crypto protocol defines two PDUs. The Crypto Sync PDU is specified in Figure T-4.

	MSB 7	6	5	4	3	2	1	LSB 0
0	Version=0				Algorithm			
1	Crypto Sync							
n								

Figure T-4: Crypto Sync PDU

For this version of the protocol, Version=0. The Algorithm identifies that Algorithm used. The length and semantics of the Crypto Sync bytes are determined by the Algorithm.

AES has Algorithm=0, and the encoding specified in Figure T-5 to give an 11 byte PDU.

	MSB 7	6	5	4	3	2	1	LSB 0
0	Version=0				Algorithm=0			
1	Key/Nonce Reference							
2								
3	Initialization Vector (IV)							
10								

Figure T-5: Crypto Sync PDU for AES

The encoding in Figure T-5 has the following elements:

1. The Key/Nonce Reference is two bytes and identifies the AES Key and Nonce pair to be used.
2. The Initialization Vector is an 8 byte IV.

The Extended Crypto Sync PDU is defined in Figure T-6:

	MSB 7	6	5	4	3	2	1	LSB 0
0		Maury-Styles						
1								
2	Version=0					Algorithm		
3		Sync Counter						
4								
5		Crypto Sync						
n+4								

Figure T-6: Extended Crypto Sync PDU

The encoding in in Figure T-6 has the following elements:

1. Maury-Styles is a two byte fixed header using the STANAG 5066 Maury-Styles value (90EB) using the same format as the one for D_DPUs defined in Annex C.

2. Sync Counter indicates the number of the Extended Crypto Sync PDU. The first one is numbered 1, the second 2, and so on. For AES, this PDU is 15 bytes long.

In order to protect against loss of Crypto Sync PDU, it is repeated in the transmission. Because transmission times can vary from less than a second to many minutes and speeds from 75bps to 240 kbps, there is a high variation of size of bounded stream. Repetition of Crypto Sync PDUs needs to be close enough to ensure repetition at the slowest speeds and to provide reasonable spacing at higher speeds to provide protection against corruption of full modem block.

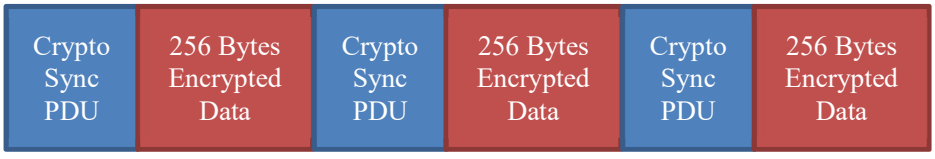

Figure T-7: Crypto Sync PDU repetition for first 4096 bytes of Data

For the first 4096 bytes of transmitted data, a Crypto Sync PDU shall inserted before every 256 bytes of encrypted data, as shown in Figure T-7. If less than 4096 bytes of encrypted data is being transmitted, the last block **may** be less than 256 bytes. Figure T-7shows the start of a block transmitted data. For AES, the Crypto Sync in this region has an overhead of 4.3%.

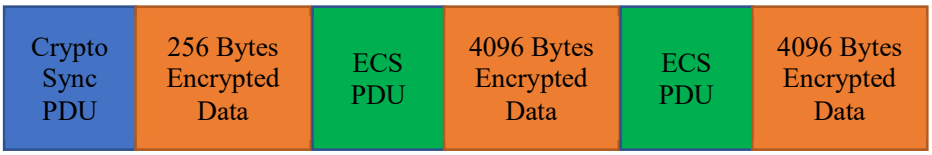

Figure T-8: Extended Crypto Sync PDU repetition

If more that 4096 bytes of encrypted data is transferred, the initial 4096 bytes are transferred as shown in Figure T-7. After this, data is sent in 4096 byte blocks, with each block preceded by an Extended Crypto Sync PDU. The last block **may** be less than 256 bytes.

Figure T-8 shows the last 256 bytes of the first 4096 bytes of data, followed by two blocks of 4096 bytes. For this extended region, AES Crypto Sync overhead is 0.4%.