1. INTRODUCTION

This memo describes the IPv6 Encapsulating Security Payload (ESP). ESP seeks to provide integrity and confidentiality to IPv6 datagrams. It may also provide authentication, depending on which algorithm and algorithm mode are used. Non-repudiation and protection from traffic analysis are not provided by ESP. The IPv6 Authentication Header (AH) might provide non-repudiation if used with certain authentication algorithms. The IPv6 Authentication Header may be used in conjunction with ESP to provide authentication. Users desiring integrity and authentication without confidentiality should use the IPv6 Authentication Header (AH) instead of ESP. This document assumes that the reader is familiar with the related document "IPv6 Security Architecture", which defines the overall security architecture for IPv6 and provides important background for this specification.
1.1 OVERVIEW

The IPv6 Encapsulating Security Payload (ESP) seeks to provide confidentiality and integrity by encrypting data to be protected and placing the encrypted data in the data portion of the IPv6 Encapsulating Security Payload. Either a transport-layer (e.g. UDP or TCP) frame may be encrypted or an entire IPv6 datagram may be encrypted, depending on the user’s security requirements. Encapsulating the protected data is necessary to provide confidentiality for the entire original datagram, but can be very expensive to implement. Use of this specification will increase the IPv6 protocol processing costs in participating systems and will also increase the communications latency. The increased latency is primarily due to the encryption and decryption required for each IPv6 datagram containing an Encapsulating Security Payload.

In order for ESP to work properly without changing the entire Internet infrastructure (e.g. non-participating systems), the original IPv6 datagram is placed in the encrypted portion of the Encapsulating Security Payload and that entire ESP frame is placed within an datagram having unencrypted IPv6 headers. The information in the unencrypted IPv6 headers is used to route the secure datagram from origin to destination. An unencrypted IPv6 Routing Header might be included between the IPv6 Header and the Encapsulating Security Payload. An IPv6 Authentication Header may be present both as an header of the unencrypted IPv6 packet and also as a header within the encrypted IPv6 packet. In such a case, the unencrypted Authentication Header is primarily used to provide protection for the contents of the unencrypted IPv6 headers and the encrypted Authentication Header is used to provide authentication for the encrypted IPv6 packet.

The encapsulating security payload is structured a bit differently than other IPv6 payloads. The first component of the ESP payload consist of the unencrypted field(s) of the payload. The second component consists of encrypted data. The field(s) of the unencrypted ESP header inform the intended receiver how to properly decrypt and process the encrypted data. The encrypted data component includes protected fields for the security protocol and also the encrypted encapsulated IPv6 datagram.

2. KEY MANAGEMENT

Key management is an important part of the IPv6 security architecture. However, a specific key management protocol is not included in this specification because of a long history in the public literature of subtle flaws in key management algorithms and protocols. IPv6 tries to decouple the key management mechanisms from the security protocol mechanisms. The only coupling between the key management
protocol and the security protocol is with the Security Association Identifier (SAID), which is described in more detail below. This decoupling permits several different key management mechanisms to be used. More importantly, it permits the key management protocol to be changed or corrected without unduly impacting the security protocol implementations. Thus, a key management protocol for IPv6 is not specified within this draft. The IPv6 Security Architecture describes key management in more detail and specifies the key management requirements for IPv6. Those key management requirements are incorporated here by reference. [Atk95a]

The key management mechanism is used to negotiate a number of parameters for each security association, including not only the keys but other information (e.g. the cryptographic algorithms and modes, security classification level if any) used by the communicating parties. The key management protocol implementation usually creates and maintains a logical table containing the several parameters for each current security association. An ESP implementation normally needs to read that security parameter table to determine how to process each datagram containing an ESP (e.g. which algorithm/mode and key to use).

3. ENCAPSULATING SECURITY PAYLOAD SYNTAX

The Encapsulating Security Payload (ESP) may appear anywhere after the IPv6 header. The Internet Assigned Numbers Authority has assigned Protocol Number 50 to IPv6 ESP. Thus the header immediately preceding the ESP header will always contain the value 50 in its Next Header field. ESP consists of an unencrypted header followed by encrypted data. The encrypted data includes both the protected ESP header fields and the protected user data, which is either an entire IPv6 datagram or an upper-layer protocol frame (e.g. TCP or UDP). A high-level diagram of a secure IPv6 datagram follows.

```
|<--        Unencrypted              -->|<----    Encrypted   ------>|
+-------------+--------------------+------------+---------------------+
| IPv6 Header | Other IPv6 Headers | ESP Header | encrypted data      |
+-------------+--------------------+------------+---------------------+
```

A more detailed diagram of the ESP Header follows. In this diagram, the cleartext fields are detailed.

```
+--------------------------------------------+
| Security Association Identifier (SAID), 32  |
| bits                                       |
+--------------------------------------------+
| Syncronisation Data, variable length       |
+--------------------------------------------+
| Encrypted Data, variable length            |
+--------------------------------------------+
```
After decryption, the data transmitted in the above "Encrypted Data" field is formatted as follows:

```
+-------------+--------------------+------------+---------------------+
| Next Header |  Header Length     |  Reserved  |                     |
+-------------+--------------------+------------+---------------------+
| Protected data (either an entire IPv6 datagram |
|                  or a UDP frame or a TCP frame), variable length |
+-------------+--------------------+------------+---------------------+
```

3.1 CLEARTEXT FIELDS

The IPv6 Header is the conventional IPv6 Header defined by others in a separate Internet Draft. The ESP unencrypted field(s) are as follows:

SECURITY ASSOCIATION IDENTIFIER (SAID)

A 32-bit pseudo-random value identifying the security association for this datagram. If no security association has been established, the value of this field shall be 0x00000000. The set of SAID values in the range 0x00000001 though 0x000000FF are reserved for future use. A security association is normally one-way. An authenticated communications session between two hosts will normally have two SAIDs in use (one in each direction).

This receiver-orientation implies that, in the case of unicast traffic, the destination system will normally select the SAID value. By having the destination select the SAID value, there is no potential for manually configured Security Associations that conflict with automatically configured (e.g. via a key management protocol) Security Associations. For multicast traffic, there are multiple destination systems but a single destination multicast group, so some system or person will need to select SAIDs for that multicast group and then communicate the information to all of the legitimate members of that multicast group via mechanisms not defined here.

Senders to a multicast group may share a common SAID for all communications if all communications are authenticated using the same security configuration parameters (e.g. algorithm, key, security classification level, etc.). In this case, the receiver only knows that the message came from a host knowing the security association data for the group and cannot authenticate which member of the group sent the datagram. Multicast groups may also use a separate SAID for each sender. In any event, the combination of Destination Address and SAID is used to determine the correct security association data. If each sender is keyed separately and asymmetric algorithms are used, data origin authentication is also a provided service.
Each SAID value implies the key(s) used to encrypt and decrypt the encrypted portion of the ESP payload, the sensitivity level (e.g. Secret, Unclassified) of the user data in the ESP payload, the encryption algorithm being used, the block size (if any) of the encryption algorithm, the authentication algorithm being used (if separate from the encryption algorithm), the block size (if any) of the authentication algorithm, and the presence/absence and size of a cryptographic synchronisation or initialisation vector field at the start of the encrypted portion of the ESP (if no such field is present, then the size is of course zero).

The sending host uses the sending userid and destination host to select an appropriate Security Association (and hence SAID value). The receiving host uses the combination of SAID value and originating address to distinguish the correct association. Hence, an ESP implementation will always be able to use the SAID in combination with the 128-bit Destination Address to determine the security association and related security configuration data for all valid incoming packets.

**SYNCHRONISATION DATA**

This field is present only for algorithms which require a cryptographic synchronisation field for each packet. The value of this field is arbitrary. The length of this field is variable, though for any given algorithm it has a particular known length. It is considered to be plaintext in this document, though most users will not be able to make sense of its contents. Its presence or absence and its size are constant for all secure datagrams of any given SAID value. The ESP specification includes this field so that the payload specification will be independent of the cryptographic algorithm that is being used by the communicating systems. If present, the field contains cryptographic synchronisation data, such as a DES Initialisation Vector, for whichever algorithm is in use. [ISO92b] An ESP implementation will normally use the Security Association Identifier value for the payload being processed to determine whether this field is present and to determine the field’s size and use if present.

**3.2 ENCRYPTED FIELDS**
The ESP encrypted fields are as follows:

**NEXT HEADER**
This field contains the value indicating which header follows the ESP Encrypted Fields. For example, if an entire IP datagram follows, the field will contain the number 94, which has been allocated by IANA to indicate IPv4 has been encapsulated. [STD-2] If the field contains the number 41, it means that IPv6 has been encapsulated. [STD-2]
HEADER LENGTH

This field contains the length of the set of encrypted ESP fields (i.e., Next Header, Header Length, Reserved) minus the 8-byte minimum length. The minimum length is 64-bit aligned to comply with normal IPv6 alignment rules.

RESERVED

This field is currently used primarily for padding out to 64-bit alignment but might be used for other purposes in the future. In order to permit such future reuse without breaking previously deployed implementations, this field MUST be set to zero on transmission and MUST be ignored upon receipt. It is 16 bits long.

It is important that all routing headers and other data be included within the encrypted IPv6 datagram, even if the same data is in the unencrypted part of the IPv6 datagram. The receiving system shall ignore all routing information in the unencrypted portion of the received datagram and shall strictly rely on the routing information from the protected payload instead. If this rule is not strictly adhered to, then the system will be vulnerable to various kinds of attacks, including source routing attacks. [Bel89][CB94][CERT95]

The encrypted IPv6 datagram need not contain any explicit Security Label because the SAID indicates the sensitivity label for the encrypted IPv6 datagram. This is an improvement over the current practices with IPv4 where an explicit Security Label is normally used with Compartmented Mode Workstations and other systems requiring Security Labels. [Ken91][DIA]

If it is necessary to pad the protected data (e.g., to an integral block-size of the cryptographic algorithm in use), then the normal IPv6 Padding header is used to provide that padding. This means that ESP will not work with block-oriented algorithms whose block size is not an integral number of 8-bit bytes.

4. ENCAPSULATING SECURITY PROTOCOL PROCESSING

This section describes the steps taken when ESP is in use between two communicating parties. Multicast is different from unicast only in the area of key management (See the definition of the SAID, above, for more detail on this). There are two modes of use for ESP. The first mode, which is called "IP-mode", encapsulates an entire IP datagram inside ESP. The second mode, which is called "Transport-Mode", encapsulates a transport-layer (e.g., UDP or TCP) frame inside ESP. These terms should not be misconstrued as restrictive, for example an ICMP/IGMP message might be sent either using the "Transport-mode" or the "IP-mode" depending upon circumstance. This section describes protocol processing for each of these two modes.
4.1 ESP in IP-mode

The sender takes the original IPv6 datagram, encapsulates it into the ESP and then applies the encryption algorithm using the appropriate key for combination of sending userid and the receiving party. If no key has been established, then the key management mechanism is used to establish a encryption key for this communications session prior to the encryption. The (now encrypted) ESP is then encapsulated in a cleartext IPv6 datagram as the last payload. If strict red/black separation is being enforced, then the addressing and other information in the cleartext IPv6 headers and optional payloads might be different from the values contained in the (now encrypted and encapsulated) original datagram.

The receiver strips off the cleartext IPv6 header and cleartext optional IPv6 payloads (if any) and discards them. It then uses the combination of Destination Address and SAID value to locate the correct decryption key to use for this packet. Then, it decrypts the ESP using the session key that has been established for this traffic.

If no cryptographic key exists for this session, the encrypted ESP MUST be discarded and the failure MUST be recorded in the system or audit log. This audit log entry SHOULD include the cleartext values for the SAID, date/time, Sending Address, Destination Address, and the Flow ID.

If decryption succeeds, the original IPv6 datagram is then removed from the (now decrypted) ESP. This original IPv6 datagram is then processed as per the normal IPv6 protocol specification. In the case of a B1 or Comparted Mode Workstation, additional appropriate mandatory access controls SHOULD be applied based on the security level of the receiving process and the security level associated with this Security Association.

4.2 ESP in Transport-mode

The sender takes the original UDP or TCP or ICMP frame, encapsulates it into the ESP and then applies the encryption algorithm using the appropriate key for the combination of sending userid and receiving party. If no key has been established, then the key management mechanism is used to establish a encryption key for this communications session prior to the encryption. The (now encrypted) ESP is then encapsulated as the last payload of a cleartext IPv6 datagram.

The receiver processes the cleartext IPv6 header and cleartext optional IPv6 headers (if any) and temporarily stores pertinent information (e.g. source and destination addresses, Flow ID, Routing...
It then decrypts the ESP using the session key that has been established for this traffic, using the combination of the destination address and the packet’s Security Association Identifier (SAID) to locate the correct key.

If no key exists for this session, the encrypted ESP MUST be discarded and the failure MUST be recorded in the system or audit log. If such a failure occurs, the recorded log data SHOULD include the cleartext values for the SAID, date/time received, Sending Address, Destination Address, and the Flow ID. The log data may also include other information about the failed packet.

If decryption succeeds, the original UDP or TCP frame is removed from the (now decrypted) ESP. The information from the cleartext IPv6 header and the now decrypted UDP or TCP header is jointly used to determine which application the data should be sent to. The data is then sent along to the appropriate application as normally per IPv6 protocol specification. In the case of a B1 or Compartmented Mode Workstation, additional Mandatory Access Controls SHOULD be applied based on the security level of the receiving process and the security level of the received packet’s Security Association.

4.3 Combining ESP and the Authentication Header

This section describes how to combine the Encapsulating Security Protocol with Authentication Header. There are two different approaches, depending on which data is to be authenticated. The location of the IPv6 Authentication Header makes it clear which set of data is being authenticated.

In the first usage, the entire received datagram is authenticated, including both the encrypted and unencrypted portions, while only the data sent after the ESP Header is confidential. In this usage, the sender first applies ESP to the data being protected. Then the other plaintext IPv6 headers are prepended to the ESP header and its now encrypted data. Finally, the IPv6 Authentication Header is calculated over the resulting datagram according to the normal method. Upon receipt, the receiver first verifies the authenticity of the entire datagram using the normal IPv6 Authentication Header process. Then if authentication succeeds, decryption using the normal IPv6 ESP process occurs. If decryption is successful, then the resulting data is passed up to the upper layer.

If the authentication process were to be applied only to the data protected by IPv6 ESP and the protected data were an entire IPv6 datagram, then the IPv6 Authentication Header would be placed normally within that protected datagram. However, if the protected data were less than an entire IPv6 datagram, then the IPv6 Authentication Header
would be placed within the encrypted payload immediately after the ESP protected header and before any other header or the UDP or TCP frame.

If the Authentication Header is encapsulated within the ESP header, and both headers have specific security classification levels associated with them, and the two security classification levels are not identical, then an error has occurred. That error SHOULD be recorded in the system or audit log using the procedures described previously. It is not necessarily an error for an Authentication Header located outside of the ESP header to have a different security classification level than the ESP header’s classification level. This is true because the cleartext IP headers might have a different classification level when the data is encrypted using ESP.

6. TYPICAL USE

The ESP supports security between two or more hosts implementing ESP, between two or more gateways implementing ESP, and between a host or gateway implementing ESP and a set of hosts and/or gateways. A security gateway is a system which acts as the communications gateway between external untrusted systems and trusted hosts on their own subnetwork and provides security services for the trusted hosts when they communicate with external untrusted systems. A trusted subnetwork contains hosts and routers that trust each other not to engage in active or passive attacks and trust that the underlying communications channel (e.g. an Ethernet) isn’t being attacked. Trusted systems always should be trustworthy, but in practice they often are not trustworthy.

In the case where a security gateway is providing services on behalf of one or more hosts on a trusted subnet, the security gateway is responsible for establishing the security association on behalf of its trusted host and for providing security services between the security gateway and the external system(s). In this case, only the gateway need implement ESP, while all of the systems behind the gateway on the trusted subnet may take advantage of ESP services between the gateway and external systems. A gateway which receives a datagram containing a recognised sensitivity label from a trusted host should take that label’s value into consideration when creating/selecting an SAID for use with ESP between the gateway and the external destination. In such an environment, a gateway which receives a IPv6 packet containing the ESP should appropriately label the decrypted packet that it forwards to the trusted host that is the ultimate destination. The IPv6 Authentication Header should always be used on packets containing explicit sensitivity labels to ensure end-to-end label integrity.

If there are no security gateways present in the connection, then two end systems that implement ESP may also use it to encrypt only the
user data (e.g. TCP or UDP) being carried between the two systems. ESP is designed to provide maximum flexibility so that users may select and use only the security that they desire and need.

7. SECURITY CONSIDERATIONS

This entire draft discusses a security mechanism for use with IPv6. This mechanism is not a panacea, but it does provide an important component useful in creating a secure internetwork.

Users need to understand that the quality of the security provided by this specification depends completely on the strength of whichever encryption algorithm that has been implemented, the correctness of that algorithm’s implementation, upon the security of the key management mechanism and its implementation, the strength of the key [CN94] and upon the correctness of the ESP and IPv6 implementations in all of the participating systems. If any of these assumptions do not hold, then little or no real security will be provided to the user. Use of high assurance development techniques is recommended for the IPv6 Encapsulating Security Payload.

Users seeking protection from traffic analysis might consider the use of appropriate link encryption. Description and specification of link encryption is outside the scope of this note.

If user-to-user keying is not in use, then the algorithm in use should not be vulnerable to any kind of Chosen Plaintext attack. A Chosen Plaintext attack on DES is described in [BS93]. Use of user-to-user keying is recommended in order to preclude any sort of Chosen Plaintext attack and to generally make cryptanalysis more difficult. Implementations MUST support user-to-user keying as is described in the IPv6 Security Architecture. [Atk95a]

ACKNOWLEDGEMENTS

Many of the concepts here are derived from or were influenced by the US Government’s SP3 security protocol specification, the ISO/IEC’s NLSP specification, or from the proposed swIPE security protocol. [SDNS89, ISO92a, IB93, IBK93, ISO92b] The use of DES for confidentiality is closely modeled on the work done for the SNMPv2. [GM93] Steve Bellovin, Steve Deering, and Dave Mihelcic provided useful critiques of early versions of this draft.

REFERENCES


Federal Information Processing Standard (FIPS) Publication 81,
December 1980.

[NIST81] US National Bureau of Standards, "Guidelines for Implementing and
Using the Data Encryption Standard", Federal Information

Federal Information Processing Standard (FIPS) Publication 46-1,


[Sch94] Bruce Schneier, Applied Cryptography, John Wiley & Sons,

[SDNS89] SDNS Secure Data Network System, Security Protocol 3, SP3,
Document SDN.301, Revision 1.5, 15 May 1989, as published

DISCLAIMER

The views and specification here are those of the author and are not
necessarily those of his employer. The Naval Research Laboratory has
not passed judgement on the merits, if any, of this work. The author
and his employer specifically disclaim responsibility for any problems
arising from correct or incorrect implementation or use of this
specification.

AUTHOR INFORMATION

Randall Atkinson <atkinson@itd.nrl.navy.mil>
Information Technology Division
Naval Research Laboratory
Washington, DC 20375-5320
USA

Telephone: (DSN) 354-8590
Fax: (DSN) 354-7942
APPENDIX A: Use of CBC-Mode DES with IPv6 ESP

This appendix describes the application of the Cipher Block Chaining (CBC) mode of the US Data Encryption Standard (DES) algorithm to the IPv6 Encapsulating Security Payload. This mode of DES requires an Initialisation Vector that is 8 bytes long and requires that the encrypted data be a multiple of 8 bytes long. DES is described in several US Government publications. [NIST77, NIST80, NIST81, NIST88] A recent book also provides information on DES. [Sch94] That same reference indicates on page 231 that at least one hardware implementation of DES CBC can encrypt or decrypt at about 1 Gbps.

Each ESP header shall contain a 64-bit DES Initialisation Vector in the Cryptographic Synchronisation field when DES-CBC is in use. Each packet needs to contain its own different DES Initialisation Vector. Including the Initialisation Vector in each packet also ensures decryption of each received packet can be performed even though other packets might have been dropped or packets might have been misordered in transit. The method for selection of the Initialisation Vector value is implementation-defined.

The secret DES key shared between the communicating parties is 64 bits long, as per the DES specifications [NIST77, NIST80, NIST81, NIST88]. The 64-bit DES key consists of a 56-bit quantity used by the DES algorithm and 8 parity bits arranged such that one parity bit is the least significant bit of each octet.

The length of the octet sequence to be encrypted by the DES algorithm must be an integral multiple of 8. When encrypting, any needed padding shall be included by using a IPv6 hop-by-hop padding option. When the Transport-mode of ESP is used, such padding must appear between the encrypted ESP header fields and the start of the UDP or TCP data. If the length of the octet sequence to be decrypted is not an integral multiple of 8 octets, then processing shall be halted, the packet shall be discarded, and the event should be recorded in the system or audit log using implementation-specific methods.