Developing Ethernet Enhanced–Security System

B.J. Herbison
Secure Systems
Digital Equipment Corporation

Abstract

The Ethernet Enhanced–Security System (EESS) provides encryption of Ethernet frames using the DES algorithm with pairwise keys, and a centralized key distribution center (KDC) using a variation of the Needham and Schroeder key distribution protocol. This paper is a discussion of some practical problems that arose during the development of this system. Section 1 contains an overview of the system and section 2 provides more detail on the system architecture. The remaining sections discuss various problem that were considered during the development and how they were resolved.

1 Overview of the System

The Ethernet Enhanced–Security System (EESS) consists of Digital Ethernet Secure Network Controllers and VAX Key Distribution Center software. DESNC controllers are encryption devices that provide node authentication and data confidentiality and integrity on an Ethernet[1] (or IEEE 802.3) local area network (LAN). The VAX KDC software manages the DESNC controllers on a LAN and enforces a LAN access control policy.

DESNC controllers are store-and-forward communication devices that sit between nodes and the Ethernet. Each controller has four ports for nodes and one port that is connected to the LAN. Communication among these five ports is restricted by the controller according to the LAN access control policy.

When Ethernet frames are exchanged between two nodes that are connected to two different DESNC controllers, the frames are encrypted by one controller and decrypted by the other controller. This encryption occurs at the Data Link layer of the network and is transparent to higher network protocol layers. Nodes can use any network protocols that normally work over Ethernet (e.g., DECnet or

The following are trademarks of Digital Equipment Corporation:
DESNC, VAX KDC, DECnet, VAX, and VMS.

S. Goldwasser (Ed.): Advances in Cryptology - CRYPTO '88, LNCS 403, pp. 507-519, 1990.

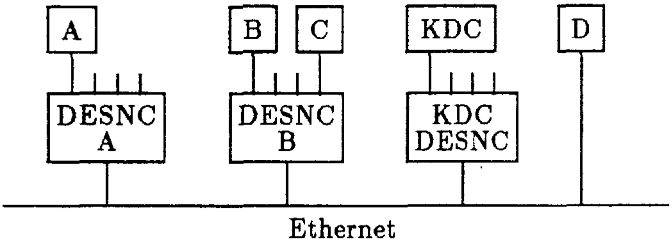

Figure 1: Sample Secure Ethernet

TCP/IP) without modification, and any device that conforms to the Ethernet or IEEE 802.3 standard can be attached to a DESNC controller.

DESNC controllers are managed by VAX KDC software running under VMS on specially designated KDC nodes on the Ethernet. Each KDC node must be attached to a DESNC controller that assists the KDC node; this controller is called a KDC controller. A KDC node and the attached KDC controller are collectively referred to as a KDC.

KDC nodes provide a user interface for the network security manager to control the security of the network. Through the interface the network security manager informs the KDC and the DESNC controllers of the configuration of the LAN, enters the LAN access control policy, determines the status of controllers on the LAN, and controls the network security auditing on the LAN.

It is possible to have multiple KDC nodes on one LAN. A large extended LAN can be supported with only a few KDC nodes, but having more than one KDC node improves the availability of the LAN by eliminating a single point of failure.

A sample secure Ethernet is shown in figure 1. The format used for encrypted node frames transmitted between DESNC controllers is shown in figure 2. Everything from the sequence number to the manipulation detection code (MDC) is encrypted. Most of the fields in the message are present as a result of the issues discussed in this paper. The section references indicate the locations in this paper where the fields are discussed.

2 System Architecture

2.1 Encryption Keys

Messages exchanged among DESNC controllers and KDCs are encrypted using the Data Encryption Standard (DES) encryption algorithm[2,3]. The messages are encrypted using the Cipher Block Chaining (CBC) mode of DES. When a message is encrypted, the encrypted portion of the message is padded to a multiple of the

Fields		Size	Section
Destination Address		6 bytes	
Source Address		6 bytes	
IEEE 802 Header		10 bytes	5.2
Message Type		2 bytes	3
Encryption Identifier		2 bytes	3
Original Header		10 bytes	5.3
Sequence Number	*	4 bytes	4
Message Type Copy	*	2 bytes	3
Original Header	*		
Original Data Field	*		
Padding	*	0-7 bytes	2.1
MDC	*	2 bytes	2.2
Ethernet FCS		4 bytes	

* marks the encrypted fields

Figure 2: Encrypted Frame Format

DES block length (8 bytes).

Several different types of DES encryption keys are used by controllers and the KDC software.

VAX KDC Master Key: This key is used to encrypt controller master keys that are stored on KDC nodes. This encryption prevents an intruder from compromising the security of a LAN by merely obtaining a copy of the information stored on the KDC node (for example, reading a BACKUP tape from the KDC node).

This key is only known by the network security manager and the KDC controllers.

Key Generation Key: This key is used as part of the process that generates encryption keys. This key is only known to the network security manager and a KDC controller.

Initialization Key: These keys, one per controller, are used to distribute the master and service keys for a controller, and are then discarded. Each initialization key is known only by the network security manager, the controller initialized with that key, and the KDC that initializes the controller.

Master Key and Service Key: These encryption keys are used to communicate between controllers and KDCs. A different pair of keys is used for each controller. The keys for a controller are only known by the controller and the KDCs, and they are only stored in encrypted form on KDC nodes. These keys are never handled in unencrypted form by any person.

Association Key: These keys are used to encrypt communication between nodes protected by controllers. A different association key is used for each pair of nodes that communicate. Association keys are distributed by KDCs when controllers request associations. Association keys are only known by the controllers involved and by a KDC controller. These keys are never stored on a KDC node or handled by any person.

KDC controllers will generate encryption keys as needed by the KDC nodes, or the network security manager can have the KDC nodes acquire the keys that they need from a user-supplied key source. A small amount of user programming is required to use a user-supplied key source, as well as a large supply of keys.

2.2 Modification Detection

When messages are encrypted, a manipulation detection code (MDC) is appended to the end of the message before encryption. The MDC field, produced by using a 16-bit CRC, is part of the encrypted portion of the message. When messages are decrypted, the MDC function of the message is calculated again and compared with the value sent with the message to determine if the message was modified as it was sent over the LAN.

2.3 Initializing Controllers

Before a DESNC controller can operate, it must be initialized. To initialize a controller, the following steps are required:

- The network security manager enters information about the controller into a KDC node. This information includes the Ethernet address of the controller, the Ethernet addresses of the nodes protected by the controller, and the access control policy for those nodes.

  The access control policy is specified by assigning an access class range to each node on the LAN. The access class ranges are from a Bell and LaPadula[4]/Biba[5] secrecy and integrity lattice, with 256 secrecy and integrity levels and 64 secrecy and integrity categories.

- On the request of the network security manager, the KDC node prints out an initialization key for the controller. The key is either generated by the KDC controller or taken from a supplied key source.

- The network security manager enters the initialization key in the controller through a keypad on the controller's front panel.

- The controller communicates with the KDC and receives its master key. The master key exchange is encrypted with the initialization key that was entered into the controller. After this step the initialization key is erased and not used again.
- The controller communicates with the KDC and receives the information that it needs to operate. This information includes:
  - The lifetime for association encryption keys.
  - The name of the firmware that the controller should be using, and a cryptographic checksum for the firmware image.
  - The addresses of the key distribution centers on the LAN.
  - The addresses of the nodes supported by the controller.
  - Information about the supported nodes.
  - A list of the events that the controller should audit.

  All of this information is encrypted under the controller's master key when it is distributed over the LAN.

After these steps, the controller is operational. The controller can now communicate with any KDC on the LAN. Once a controller is initialized it is not necessary to enter any additional information manually. If the distributed information needs to be changed, the changes can be made remotely from any KDC. DESNC controllers retain the distributed information during power-off and over power interruptions, but the information will be erased if a DESNC controller is opened.

Operational controllers request association keys from KDC nodes as necessary, and encrypt and decrypt Ethernet frames sent by nodes using those keys.

2.4 Downline Loading Controllers

The operational firmware image used by DESNC controllers is downline loaded over the Ethernet using the same mechanism employed by other Digital products. This allows the controllers to be downline loaded by the same downline load servers that load other products on the Ethernet. These images are not encrypted and the servers are not necessarily KDCs. The integrity of the images (and the security of the LAN) is protected in the following manner:

1. When a new firmware image is installed on a KDC and downline load servers, the KDC generates an encryption key and a cryptographic checksum for the image. The KDC generates a different key and checksum for each controller on the Ethernet.

2. During controller initialization, the KDC distributes the name of the firmware image and the appropriate checksum information to each controller. If a new image is installed after a controller is initialized, any KDC may distribute the new image name and checksum information to the controller.

3. When a controller needs to be downline loaded, it requests the appropriate image. After it receives the image from a downline load server, the controller calculates the checksum for the image and compares the value against the stored value. If the received image does not have the correct checksum then that image is ignored and a new image is requested.

2.5 Associations

When two nodes try to communicate by exchanging Ethernet frames over the LAN, controllers will not allow the communication unless the *association* is allowed by a KDC. If allowed, these associations are granted upon demand by KDCs.

There are three different types of associations:

- Associations between two nodes protected by different controllers. Frames sent under these associations are secured through encryption while they are on the Ethernet.

  An example of this type of association would be an association between node A and node B in the LAN shown in figure 1.

- Associations between two nodes protected by the same controller. Frames sent under these associations are never sent on the Ethernet so there is no need for them to be encrypted. The controller only sends the frame to the node port where the destination node is attached, so this type of association is secure.

  An example of this type of association would be an association between node B and node C in the LAN shown in figure 1.

- Associations between a node protected by a controller and a node not protected by a controller. Frames sent under these associations are not encrypted (because there is no second controller to decrypt the frame), but communication is not allowed unless approved by a KDC.

  An example of this type of association would be an association between node C and node D in the LAN shown in figure 1.

When communication occurs between two nodes not protected by DESNC controllers, controllers and KDCs are not involved in the communication.

2.5.1 Encrypted Association Set-Up

The protocol exchange used between DESNC controllers and KDCs to encrypted set-up associations is similar to the protocols described in Needham and Schroeder[6] and Voydock and Kent[7]. Here is an example of how an association would be established between two nodes that are both connected to DESNC controllers.

Consider the LAN shown in figure 1. Setting up an association between node A and node B involves the following steps:

1. Node A sends an Ethernet frame to node B.

2. Controller A receives the Ethernet frame and verifies the source address of the frame.

3. Controller A requests an association from the KDC.

4. The KDC checks the access control policy, determines that nodes A and B are allowed to communicate, and sends an Association Open message to controller A. The Association Open message is encrypted with the master key of controller A. The message contains an association key, either generated by the KDC controller or taken from a supplied key source.

5. Controller A sends an Association Forward message to controller B. The Association Forward message is encrypted with the master key of controller B. This message was generated by the KDC and included in the Association Open message sent to controller A.

6. Controllers A and B communicate and determine that they share a common association key.

7. Controller A encrypts the frame sent in step 1 with the association key and sends the encrypted frame to controller B.

8. Controller B receives the encrypted frame, decrypts the frame, checks the manipulation detection code, and transmits the frame to node B.

Once the association is established, no further interaction with the KDC is required and all communication between nodes A and B is encrypted with the association key until the association expires. If an association is active and approaching expiration, the controller that originally requested the association (controller A in this example) will request another association before the first association expires.

The duration of associations is determined by the network security manager, and the information is distributed to controllers when they are initialized.

2.5.2 Unencrypted Association Set-Up

Associations that involve only one DESNC controller (either because both nodes are attached to the same controller or because only one node is attached to a controller) do not require an encryption key and do not involve synchronization between two controllers. Setting up these associations is simpler than setting up encrypted associations, several steps can be omitted.

For example, if node B in figure 1 wants to communicate with node C or node D, the following steps are required:

1. Node B sends an Ethernet frame.

2. Controller B receives the Ethernet frame and verifies the source address of the frame.

3. Controller B requests an association from the KDC.

4. The KDC checks the access control policy, determines that the nodes are allowed to communicate, and sends an Association Open message to controller B. The Association Open message is encrypted with the master key of controller B. No encryption keys are included in the message.

5. Controller B sends the frame received in step 1 to the appropriate destination (either to the correct port or to the LAN).

As in the previous case, communication between the node pair continues without KDC intervention for the duration of the association.

2.6 Trust

With any security system, it is important to know which components must be trusted, and the degree of trust required. The EESS architecture was designed to limit the degree to which an individual DESNC controller needs to be trusted.

The compromise of a DESNC controller may compromise the nodes protected by the controller, but will not compromise any other controllers or nodes on the LAN. This means that a controller must be protected as well as any of the nodes protected by the controller.

If multiple nodes are connected to the same node port of a controller, the nodes can masquerade as each other. This means that those nodes must be mutually trusting. If this is level of trust is not appropriate, a site can use DESNC controllers with only one node attached to each of the four node ports.

If a KDC node or the controller that supports a KDC node is compromised, the security of the LAN can be compromised. This means that KDC nodes and KDC controllers must be protected as well as any node on the LAN. While KDC nodes can be used for multiple purposes, the security of the network is improved if the KDC nodes are limited to network management functions and access to the nodes is limited to trusted individuals.

3 Determining the Encryption Key

Unless a cryptographic system uses a single encryption key to encrypt all messages exchanged, it is necessary to determine whether a message is a control message or an encrypted node frame and which key should be used to decrypt a particular message. While, in many cases, it is possible to determine the correct key from the context and from the source and destination addresses, the choice is occasionally ambiguous. Placing the information in the message explicitly avoids any ambiguity and is also more efficient to handle. To prevent modification attacks on the protocol messages, it is necessary to guarantee that any modifications of this information are detected.

The messages exchanged between the components of an Ethernet Enhanced-Security System contain two fields that are used to identify the encryption key. Each message contains:

Message Type: This field identifies the type of the message and, in particular, whether the frame is a control frame or an encrypted node frame.

This field is protected against modification by including a duplicate copy of the field in the encrypted portion of the frame (protected by the manipulation detection code). These copies are compared after the frame is decrypted.

Encryption Identifier: Once the type of message is known, this field uniquely identifies the encryption key.

Rather than use an explicit check, this field is implicitly verified. If the field is modified, then the wrong encryption key will be used to decrypt the frame and the manipulation detection check will fail.

For encrypted node frames, DESNC controllers are designed to allow rapid determination of the association key from the encryption identifier.

4 Sequence Numbers

When Ethernet frames are encrypted, sequence numbers are used for two purposes:

- To prevent attacks that involve the replay or reflection (exchange of source and destination addresses) of encrypted Ethernet frames.
- To whiten messages to prevent intruders from inspecting two encrypted Ethernet frames and determining if the original frames (or an initial portion) were identical.

Sequence numbers are used to protect both encrypted node frames transmitted between DESNC controllers and the control frames exchanged among the controllers and between controllers and KDCs.

Sequence numbers are 4 bytes long and contain a 31 bit count value and a 1 bit direction.

4.1 Sequence Numbers Versus Timestamps

While timestamps are commonly used to detect replay attacks, the EESS architecture uses sequence numbers. Using sequence numbers avoids then problem of synchronizing clocks, and sequence numbers were found to be easier to generate and compare than timestamps. In particular, detecting replay attacks while allowing for out-of-order frames it is easier and requires less storage with sequential sequence numbers than with timestamps.

However, sequence numbers do have their own synchronization problems. A pair of components can loose synchronization if one component sends a large number of messages while the other component is not working or is otherwise out of communication. For example, this may occur if a KDC is unable to communicate with some controllers for several days.

The architecture provides a way to securely resynchronize sequence numbers when these problems occur. KDCs and controllers synchronize their sequence numbers when they exchange status information. This synchronization allows any sequence number mismatch to be corrected. To avoid any possible replay attacks, sequence numbers are only raised during synchronization, never lowered.

4.2 Sequence Number Use

The EESS architecture uses a separate sequence number stream for each encryption key used. The keys used to encrypt node frames are distributed upon demand by KDCs and are used for at most a few days. The encryption keys used for control messages are used for longer periods.

Each time a message is transmitted using a particular encryption key, the DESNC controller (or KDC node) transmitting the message increments the sequence number associated with that encryption key. When a message is received, the recipient controller checks the sequence number and rejects the message if the sequence number is significantly lower than the highest sequence number received, or if another message has been received with the same sequence number.

Messages are accepted out of order, and no attempt is made to reorder the messages or to guarantee that all messages are delivered. (These functions are not normally provided by the Data Link layer, and should be provided by higher protocol layers.)

5 Interoperability

Because encrypted Ethernet traffic will probably be using the same Ethernet cable as unencrypted traffic, an Ethernet security system must be a 'good neighbor' on the LAN. This implies that the system must:

- Follow the Ethernet physical standards, including the restriction on maximum frame length,
- Follow standard Ethernet packet formats by including valid frames headers on all transmitted frames, and
- Allow the LAN to be maintained, or at least not prevent standard LAN maintenance operations.

The implications of each of these restrictions are discussed below.

5.1 Frame Length

For several reasons, including the addition of sequence numbers to frames, it is necessary for the size of frames to be increased when the frames are encrypted by DESNC controllers. However, to satisfy the Ethernet standard, the Ethernet frames transmitted by DESNC controllers must not be more than 1518 bytes long. There are two possible resolutions to these two requirements: Either restrict controllers to only encrypting frames that are short enough to be encrypted without exceeding the length restriction, or fragment long Ethernet frames when they are encrypted.

We chose to fragment long Ethernet frames when they are encrypted by a controller, and to reassemble them transparently when they are decrypted by the recipient controller (before they are transmitted to the destination). When one long Ethernet frame is transmitted by a node, two separately encrypted Ethernet frames are sent from one DESNC controller to the other. The recipient controller checks each frame and uses the two frames to rebuild the original Ethernet frame. The frame received by the destination node is identical to the frame transmitted by the source node.

Fragmentation affects the performance for long frames because it is necessary to send twice as many frames. But it is possible for network users to voluntarily reduce the length of Ethernet frames they transmit. This reduction avoids the need for fragmentation for the applications that can handle a reduced maximum frame length, but the fragmentation allows any existing application to continue to work correctly even if it sends maximum length Ethernet frames.

5.2 Frame Header

When frames are encrypted, the source and destination addresses are left unencrypted and the rest of the original header is replaced by an IEEE 802 header with a protocol identifier that identifies the frame as an encrypted frame.
The reasons for this are:

- If the addresses were encrypted, then Ethernet bridges would no longer be useful in filtering network traffic and there would be a significant performance penalty because it would be necessary for every node to decrypt each frame to determine if it is the intended recipient.

  The addresses are authenticated by the encryption key used, so there is no loss of node authentication due to plaintext addresses.

- If the original header (other than the addresses) was encrypted, the header would no longer have have a valid format (i.e., it would probably have an unassigned Ethernet protocol type, or an incorrectly formatted IEEE 802 header), thereby confusing LAN monitoring tools.

- If the original header is left unchanged, the rest of the message would look malformed (for a message with that header) because it was encrypted. This would also confuse LAN monitoring tools. (Also, an integrity check would be necessary for the header.)

- A distinct header provides an easy way to determine if a message needs to be decrypted when it is received.

Therefore, even though the extra header increases the overhead of encrypting the Ethernet frames, the header is added because it simplifies the processing of the frames and prevents confusion over the contents of the frame.

5.3 Network Maintenance

While it is necessary to replace the headers of encrypted Ethernet frame with headers containing protocol identifiers that identify the frame as being encrypted, the original header is also useful to network management tools. Tools that can examine this header can determine how a LAN is being used.

For this reason, DESNC controllers include the original header of the frame (except for the addresses) in unencrypted form after the header that identifies the frame as being encrypted. The header is also included in the encrypted portion of the message so that attempted modifications to the message can be detected.

When the frame is encrypted the DESNC controller examines the start of the frame and determines the frame format and the size of the header fields (excluding the addresses). DESNC controllers distinguish between:

- Ethernet format frames (with only a 2 byte protocol type),

- IEEE 802 format frames (with 5 or 6 bytes of header fields), and
- IEEE 802 format frames with protocol identifier (with 10 bytes of header fields).

The original header fields are copied into a 10 byte field in the encrypted message. This field is zero-padded if the header fields are shorter than 10 bytes.

References

[1] *The Ethernet, A Local Area Network, Data Link Layer and Physical Layer*, Version 2.0, (Digital, Intel, and Xerox), November 1982.

[2] *Data Encryption Standard*, Federal Information Processing Standards Publication 46 (FIPS PUB 46), National Bureau of Standards, 15 January 1977.

[3] *DES Modes of Operations*, Federal Information Processing Standards Publication 81 (FIPS PUB 81), National Bureau of Standards, 2 December 1980.

[4] D.E. Bell and L.J. LaPadula, *Secure Computer Systems: Unified Exposition and Multics Interpretation*, ESD-TR-75-306, MITRE Corporation, March 1976.

[5] K.J. Biba, *Integrity Considerations for Secure Computer Systems*, ESD-TR-76-372, MITRE TR-3153, MITRE Corporation, April 1977.

[6] R.M. Needham and M.D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers", *Communications of the ACM*, December 1978.

[7] V.L. Voydock and S.T. Kent, *Security in Higher Level Protocols: Approaches, Alternatives and Recommendations*, Report No. ICST/HLNP-81-19, National Bureau of Standards, September 1981.