SICS/T-90/9004

State of the Art in Network Security

Bengt Ahlgren Per Lindgren Teet Sirotkin

March 1989

SICS technical report
T90004
ISSN 1100-3154

SICS/T-90/9004

State of the Art in Network Security

Bengt Ahlgren Per Lindgren Teet Sirotkin

March 1989

SICS technical report T90004
ISSN 1100-3154
This report was first printed on September 7, 1988. Minor corrections have been made.

Swedish Institute of Computer Science
Box 1263
S-164 28 KISTA
Sweden

Abstract

This report is an effort to describe the state-of-the-art in computer network security focusing on the OSI Security architecture. Other sources of information include the NCSC "Trusted Network Interpretation of the TCSEC".

The report describes the security threats imposed on networks and the countermeasures available. It gives a detailed description of the security services defined in the OSI Security architecture and the mechanisms proposed for realizing these services. An overview of security management with emphasis on key management is also included. The report contains numerous references to books and articles in the field of network security.

Contents

Preface

Our intention with this paper is to provide an overview of the state-of-the-art in network security, with emphasis on the OSI Security Architecture [ISO88]. It is mainly a summary of articles we regard as good. The document is structured with respect to the specific topics, not with respect to the different articles. This means that references to an article addressing many topics are scattered over several sections.

The reader should have some basic knowledge in computer networks, computer security and some familiarity with the OSI-model.

The first section is an introduction to and overview of network security. The second section is about standardization work. Then five sections follow which treat the security services defined in the OSI Security Architecture in greater detail. Section 8 relates security services to the OSI model; in what layer in the model can the services be placed? A secure network needs management to support the security services. Section 9 is mainly about key management, which is an important part of network security management. Section 10 of the paper is a summary of four articles describing more or less complete security protocols. The protocols contain solutions to some of the security issues discussed in the preceding sections. There are also two appendices containing definitions and recommended further reading.

For the purpose of readability we sometimes use different notations than the article cited.

Because the field of network security is relatively new, it is heterogeneous and rapidly changing. In the last few years the state of the art has begun to stabilize. The NCSC "Red book" has become well known, the OSI Security Architecture has became a standard, and a number of implementations have seen daylight. However, it is still difficult to get a good overview of the whole area of network security. This is what prompted us to write this paper.

We welcome any kind of response to our paper. We can be reached at the address: SICS, Box 1263, S-164 28 Kista, Sweden. It is also possible to send electronic mail to `postmaster@sics.se`.

BENGT AHLGREN
PER LINDGREN
TEET SIROTKIN

1 Security Threats and Countermeasures

> "ISO has identified a need to generate a series of standards for security enhancement within the OSI Architecture. The motivation for this work stems from:
>
> (a) the increasing dependence of society on computers which are accessed by, or linked by, data communication and which require protection against various threats, real or apprehended..." [ISO88]

Most literature on security in computer networks starts with defining the following security services:

- Protection of data against unauthorized disclosure (confidentiality).
- Protection of data against undetected loss/repetition (integrity).
- Protection of data against unauthorized modification (integrity).
- Assurance of the correct sender/receiver of data (authentication and non-repudiation).

These are the goals stated by Branstad [Bra87] as a "minimum set of security goals" in a LAN environment.

Other security goals include labeling of data according to its sensitivity, nondisclosure of the amount of data exchanged between peer entities [VK84] and recovery from breaches in the security mechanisms (e.g., disclosure of passwords etc.) [Den83]. Nonalgorithmic issues like key granulation and key distribution are discussed by Branstad [BDHR87], Denning [Den83] and Voydock & Kent [VK85].

1.1 Security policies

It is important to develop a security policy for the network [ISO88]. The policy chosen affects the choice of mechanisms to implement a certain service, what is to be protected, etc.

As an example of a policy decision, one can take the choice between the use of *end-to-end* or *link-by-link* security [VK85]. End-to-end security means that the data is 'safe' until it reaches the 'end user', which in OSI terms means that the security measures are applied at layer 4 or higher. Link-by-link security means, for example, that the datastream is encrypted just before it is sent to the physical distribution device (an ethernet cable for instance). That implies that the data must be decrypted at every node or link (packet switch, gateway, ...) between the sender and the recipient.

The primary advantage of a link-by-link policy, without an overlaying end-to-end security service, is that of transparency; the protocols used need almost no modifications, whereas if end-to-end security is utilized, extensive changes are needed [VK85], and the efforts to standardize security protocols has only recently begun [ISO88].

It is important to remember that the scope of the above mentioned goals is not to give a perfectly secure environment. Almost all our sources recognize the need for physical protection of the equipment (locked doors etc.) as a complement to the protection given by the protocol. This applies specially to those who propose centralized resources for storage of secret information (passwords, encryption keys, etc.). They must have an absolute trust in those resources; if the resource is compromised the whole net is rendered unsafe. Denning [Den83] will not commit herself to this. She uses one (or more) centralized resource(s), but not for storage and management of any secret information. There is also the aspect of vulnerability whenever a single resource is paramount for the correct functionality of a whole network.

1.2 Countermeasures

What can be done to meet the security goals stated by the articles? The bases of all countermeasures are encryption and (cryptographic) checksums, used independently or in combination [ISO88].

1.3 Confidentiality

Encryption is used to secure the confidentiality of data. The precise algorithm chosen is determined by such things as

- how long the cipher must withstand unauthorized decryption, and
- how fast the encryption must be.

Since the most desirable algorithms are quite slow, encryption is preferably implemented in hardware. [VK84]

1.4 Integrity

Unauthorized modification of data, although it cannot be prevented, can readily be detected [Bra87]. Methods used here are checksum computation together with encryption. The sender computes a checksum over the (possibly encrypted) data it is about to send and then encrypts the checksum. The encrypted checksum is then passed with the data being sent to the receiver. Note that it is not necessary to encrypt the data but to encrypt the checksum. If the data is changed, the checksum computed by the receiver will not be the same as the checksum received.

Loss of data, i.e., data being hindered to reach its destination, can equally easy be detected, but not prevented [Bra87]. That is to say if we are dealing with connection oriented communication. To be able to detect breaks or very long delays in a data stream, the data packets can be sequentially numbered. If this is combined with an exchange of data at regular intervals, even when the end users are not communicating, a broken connection can be detected at both ends. This makes it possible to detect an attempt to break the communication after the connection has been established, but before (or in) any stream of data is being sent. There are limitations on what types of communicating entities that can use this scheme. [VK85].

1.5 Authentication

In order to establish a secure connection between two entities, one-way or mutual authentication is desirable. The simplest solution (in some sense) is for the communicating entities to have a common secret key with which they encrypt and decrypt all data sent between each other. Different approaches can be used by the entities in order to establish a key:

- Use the same key all the time
- Fabricate a new key at every connection establishment
- Request a key from a trusted third party

An issue at hand here is key disclosure. How to prevent and how to limit the effects of such a loss is discussed at length by Denning [Den83].

1.6 Non-repudiation

It is sometimes important to be able to legally bind a sender or receiver to the sending or reception of data. A non-repudiation service provides this. Methods employed here are digital signatures and notarization. One or more (see [Den83]) resources on the network can act as notary public, to correctly and safely save records of all communication using this security service. If a dispute arises, the records can be consulted, and the dispute settled. It is natural that these resources must be well protected, not because they hold secret information, but because they are 'controlling' the truth. Change the record and you change the truth. Luckily, Denning [Den83] sees no problem to prevent this, nor to detect and take appropriate countermeasures if it is.

1.7 Traffic analysis

Voydock and Kent [VK84, VK85] pinpoints another threat; traffic analysis. If end-to-end security is used, at least the receiving hostname must be sent in cleartext in each PDU. This allows the determination of which hosts that are communicating, and how much they communicate. A link-by-link policy could prevent both issues, the former by encryption and the latter by maintaining a continuous stream of data (possibly meaningless such) between nodes. But [VK85] thinks that a combination of link-by-link and end-to-end is a waste of bandwidth in the net computing power in the nodes. According to [VK85] the cost of limiting traffic analysis more than is possible with the use of end-to-end methods is too high (and probably not really necessary). Abrams and Jeng [AJ87] feel they must issue a warning against the belief that an end-to-end policy can solve all problems (see the article for further references).

1.8 Use of old connections

Another 'mischief' that can be attempted by a perpetrator (called C) is to replay a previously correct establishment of a connection between two

entities (A and B). If C can fool A that B wants a connection by resending an old connection request from B, C can (if he succeeds) act to A as if he were B. To prevent this a number of different schemes can be employed:

- Timestamping the connection request [MPH88, Kar85, VK85].
- Use of unique session identifiers [Kar85, VK84, VK85].
- Use of a trusted third party to correctly identify (time-wise and identity-wise) A and B [Den83].

1.9 Access control

Access to various types of network resources also need to be controlled. The resources may be other hosts, software on other hosts, files on other hosts, CPU-time on other hosts, etc. To control access to all these resources, various types of capability lists and other kinds of security management information tables are needed [ISO88, Kar85].

1.10 Insider attacks

Since all computer and network equipment requires humans to operate it, the risk of someone doing something unauthorized with, or to, the equipment must be considered. In fact, most known computer crimes has involved insider attacks. Among the few precautions that can be taken to prevent these insider attacks are:

- Selecting the staff carefully [ISO88].
- Only use software and hardware that do not have any known deficiencies that can be used to break security [Sch88, ISO88].
- Keeping logs of all actions taken with or to the equipment.

Trojan horses and trapdoors are other mechanisms that can be introduced by a perpetrator (primarily in software) in order to help circumvent security barriers.

1.11 Conclusions

Most of the topics mentioned are discussed in many of our references. They also conform to the same general solutions to a very high extent. All references in this paper therefore mostly refer to papers that we consider as 'good' references rather than unique ones. The differences become clearer at a more detailed level, but this section only concerns itself with the broader issues of security threats and its countermeasures.

2 Standardization Work

2.1 Trusted Computer System Evaluation Criteria

In 1983 the US Department of Defense published a document called "Department of Defense Trusted Computer System Evaluation Criteria" [DoD83] also known as "the Orange book". The document defines seven **evaluation classes**, into which computer systems can be sorted according to their security properties.

The classes are as follows (in increasing security order):

D — Minimal protection Systems that are evaluated, but fail to meet the requirements for the higher classes.

C1 — Discretionary security protection Systems that allow users to protect data from accidental or deliberate reading or destroying. It is meant for users processing data at the same level of sensitivity.

C2 — Controlled access protection Systems with more finely grained access control, making users accountable for their actions through login procedures and logging of security relevant events.

B1 — Labeled security protection Same as C2, but an informal statement of the security policy model, data labeling, and mandatory access control over named subjects (users) and objects (resources) must be present.

B2 — Structured protection In this class, the security mechanisms are based on a clearly defined and documented formal security policy model. All subjects and objects are under discretionary and mandatory access control. Covert channels are protected. Authentication mechanisms are strengthened and trusted management is provided in the form of support for administrator and operator functions. Stringent configuration management controls are imposed.

B3 — Security domains In this class, the TCB[1] must mediate all accesses, be tamperfree and non-complex, to allow for analysis and tests. A security administrator is supported, audit mechanisms are expanded to signal security relevant events, and system recovery procedures are required.

A1 — Verified design Functionally equivalent to B3, but the TCB is constructed with formal specification and verification techniques.

To be sorted in these classes, each computer system must meet the following basic requirements:

- There must be an explicit and well-defined security-policy.
- Access control labels must be associated with each object.

[1] Trusted Computing Base; The totality of all security mechanisms, see the definitions.

- Individual subjects must be identified.
- Audit information must be kept and protected so that actions affecting security can be traced to the responsible party.
- The system must contain mechanisms that can provide assurance that the system enforces the basic requirements above.
- The mechanisms that enforce the above basic requirements must be continuosly protected against unauthorized changes.

2.2 Trusted Network Interpretation

The Orange book was followed by a "Red book", issued in 1987 by The National Computer Security Center (NCSC). It has the title "Trusted Network Interpretation" [NCSC87] and extends the evaluation classes to network systems. It also describes a number of additional security services (e.g., communication integrity, denial of service, and transmission security), that are needed in conjunction with networks.

2.3 OSI Security Architecture

The OSI Security Architecture [ISO88] became an international standard, ISO 7498-2, in early 1988. It provides general descriptions of security services and related mechanisms, which may be provided by the OSI-model. It also defines the positions within the model, where the services and mechanisms may be located. The standard is meant to provide a common framework for network security and be a basis for more detailed specifications; it is not detailed enough for use as an implementation specification.

2.4 Comparison

There are some differences between the Red book and the OSI Security Architecture, although they cover the same area. For example, the distinction between the security services and the mechanisms that implement the services is stronger in the latter. Another difference is that the Red book emphasizes the importance of assurance (i.e., continuous control that the mechanisms are functioning adequately), a subject that is rarely addressed in the OSI Security Architecture. Security within the end-systems, installations and organization are also topics outside the framework of the OSI Security Architecture.

2.5 Cryptographic algorithms

The work within ISO concerning the standardization of algorithms for data encryption (in particular the DES and RSA algorithms) has been abandoned.

The main reason is to discourage an overconfidence in the algorithms subject to the standardization work. Instead, a register over possible algorithms will be maintained, giving a choice as to which algorithm to use. In the proposed register, both published and unpublished algorithms can be entered. The register will only contain a minimum of information about the algorithms (for example, name, supplier(s), block size and key domain).

This is to be seen in parallel to the work within the USA on replacement algorithms for the DES. The 'new' algorithms, when they are ready, will not be published, and thus preventing any form of (international) standardization. [Pri87]

One reason for not standardizing RSA may be the fact that "the RSA cryptosystem is patented, by MIT." (Ronald Rivest, USENET, 1985)

"The patent covers *communication systems* using their algorithms", not the algorithm itself (algorithms cannot be patented). "The patent also covers a lot of subsidiary claims, such as using RSA only for key exchange." (Steven Bellovin, USENET, 1988)

RSA is as far as we know only patented in the USA (US patent number 4 405 829), which means that it is legal to implement the algorithm without licence anywhere else.

3 Confidentiality

> "Confidentiality: The property that information is not made available or disclosed to unauthorized individuals, entities or processes." [ISO88]

3.1 Services

ISO defines the following services for data confidentiality:

- Connection confidentiality—confidentiality of all user data on a connection.
- Connectionless confidentiality—confidentiality in all single connectionless data packets.
- Selective field confidentiality—only some part of the data is subject to confidentiality measures.
- Traffic flow confidentiality—to disguise *how much* communication is actually taking place.

3.2 Mechanisms

Data can be protected against unauthorized disclosure by means of **encryption**. The result of encryption is called **ciphertext** (or cryptogram). The encryption algorithm uses a **key**. To produce identical ciphertext from identical cleartext the same key must be used (if the encryption algorithm is 1-1 and onto). The reverse of encryption is called **decryption**. To decrypt a message correctly, only one key should be possible to use, the one intended for the decryption.

There are two types of encryption algorithms:

Symmetric key: Encryption and decryption is done with the same secret key.

Asymmetric key: Different keys are used for encryption and decryption. In some asymmetric key systems knowledge of one of the keys gives no information as to the identity of the other key. These kinds of cryptographic systems are called **public key systems**.

Note also that the use of encipherment implies the use of a key management mechanism.

Selective field confidentiality is a higher-level service which takes advantage of the characteristics of the data that will be protected. The actual service (encryption) may lie in a lower layer than where it is used.

Traffic flow confidentiality can be achieved by **traffic padding**. Traffic padding is the generation of spurious traffic and padding of protocol data units (PDUs) to a constant size. The spurious traffic must amount to the highest expected level of 'real' traffic. The content of the padding PDUs

must be indistinguishable from the content of the 'real' PDUs; thus traffic padding is useful only together with data confidentiality.

3.3 The Data Encryption Standard

Voydock and Kent [VK85] uses the same mechanisms as ISO [ISO88] for data and traffic flow confidentiality. They want to place the encryption for the end-to-end security service in layer 4, since that is the lowest layer where any additional protection of data can be given compared with placing it in a higher layer. The link-by-link scheme is to them not necessary, unless it is for military applications.

The article first defines two different ways to use the *DES* algorithm.

- ECB – "Electronic Code Book" – mode (unmodified DES).
- CBC – "Cipher Block Chaining" – mode.

3.3.1 ECB – Electronic Code Book mode

The ECB mode of DES is a block cipher over 64 bit blocks with a 56 bit key. Each block of cleartext is fed to the algorithm together with the key. This means that each block is encrypted independently. How sensitive is the DES to transmission errors (or message stream modification)? If 1 bit in the key/cleartext or key/ciphertext pairs is changed, there is a 50% chance of change in every bit in the ciphertext or cleartext. The change, however, is limited to the current block.

Identical cleartext block encrypted under the same key produces identical ciphertext blocks. This can be an exposure of block-sized data patterns that fall on the boundaries of the data. If padding in user data is done with the same value, the padding and its extent can be detected. Clearly this unmodified version of DES isn't secure enough.

3.3.2 CBC – Cipher Block Chaining mode

In CBC mode the DES algorithm need what is called an **initialization vector** (IV) of the same size as the message blocks (64 bits). The IV is used in the encryption of the first block (first 64 bits of the message). In all subsequent encryptions the ciphertext of the previous encryption is used. If we denote the message blocks with M_i for $i = 1, 2, \ldots, n$, the resulting ciphertext with C_i and the key with K, then we have:

$$\begin{aligned} C_1 &:= E_K(IV \oplus M_1) \\ C_i &:= E_K(C_{i-1} \oplus M_i) \qquad 2 \leq i \leq n \end{aligned}$$

where E_K is the DES encryption under the key K. $\oplus$ is the bitwise XOR operation. Similarly, when we decrypt we have:

$$\begin{aligned} M_1 &:= D_K(C_1) \oplus IV \\ M_i &:= D_K(C_i) \oplus C_{i-1} \qquad 2 \leq i \leq n \end{aligned}$$

where D_K is the inverse of E_K (i.e., the decryption).

In case of transmission errors or stream modification the error propagates into the next block only, assuming that the next block of ciphertext is received correct.

Now, since all encrypted blocks (except the first one) depend on the previous one, identical ciphertext is possible only when the cleartext have identical prefixes (preceding 'cleartext' block). Thus, all depends on the IV, and we can formulate the following requirements:

- Use a different IV for each association.
- The IV must be pseudorandomly chosen.
- IV's must be kept secret (during the time of an association) or be different for each data packet.

3.3.3 Conclusions

All privacy measures are based on data encryption. It prevents passive attacks by preventing the intruder from observing data in the clear. The most interesting encryption algorithm is the DES algorithm with some sort of modification (e.g., CBC). Data patterns can be masked by using unique keys for each association and by careful selection of IV's. (Work has begun in the USA to find replacement algorithms for the DES, since it is thought not to be safe so much longer [Pri87].) It is also interesting to notice, that ISO has abandoned the work on making DES an international standard, see section 2.5.

Where shall the encryption be placed? It is of no use to place an encryption mechanism below the transport layer (V&K assumes end-to-end security), and if placed above layer 4, additional protocol information is released. So the most logical position would be in the transport layer.

3.4 Conformity

The confidentiality service is not a subject for debate. Very few differences exist between the different papers. In fact, most papers don't even bother to describe the encryption methods proposed for usage (mainly the DES). The only subject where different opinions can be seen is on how to create/ store the keys. More on this subject can be found in section 9, Security Management.

4 Integrity

> "The property that data has not been altered or destroyed in an unauthorized manner..." [ISO88]

Integrity violation is an active threat against the packets sent on a connection. Most authors identify the following integrity threats:

- Modification
- Insertion of a synthesized packet
- Deletion
- Replay
- Reordering

Integrity attacks can be detected and recovered from, but not prevented.

4.1 Services

The following integrity services are suggested by [ISO88]:

- Connection integrity with recovery—Detection of integrity errors with attempted recovery.
- Connection integrity without recovery—Detection of integrity errors with no attemted recovery.
- Selective field connection integrity—Detection of integrity errors in selected fields within a data unit.
- Connectionless integrity—Detection of integrity errors in a single connectionless data unit.
- Selective field connectionless integrity—Detection of integrity errors in selected fields within a connectionless data unit.

4.2 Mechanisms

Different mechanisms are used to protect single data units and to protect data streams, although protection of streams makes it easy to protect single data units as well.

4.2.1 Modification

The obvious way to provide integrity for a data packet, according to [ISO88], is to append a checkvalue, which is a function of the packet including its header. The receiving entity computes the same checkvalue, and compares it to the one received. If the checkvalue is incorrect, then the packet was modified, or the wrong encryption key was used at some end. This mechanism relies on the fact that the checksum is encrypted. Some algorithms though, use a key as a seed for calculating the checkvalue, which makes the encryption redundant.

4.2.2 Insertion of a synthesized data unit

The mechanism above also protects against insertion of synthesized data units. Even if an intruder could calculate the correct checksum for the spurious message, he could not encrypt it with the correct key.

4.2.3 Reordering

Reordering attacks are discussed by Voydock and Kent [VK85]. This kind of attacks involve:

- Deletion
- Reordering
- Replay

Connection oriented communication requires a sequence number to be added to each data packet to detect reordering. Integrity mechanisms ensure that any attempt to change this number will be detected. Authentication mechanisms ensure, that the packets belong to the connection. To protect against replay, the same sequence number should never be reused in the same connection.

Another alternative is to use cryptographic chaining. But Voydock and Kent points out a weakness with this approach: "Restoring cryptographic synchrony after integrity violation can be difficult and expensive if the ability to decrypt a PDU depends on having successfully decrypted all previous sent PDUs. For this reason, protocols that implement countermeasures should encrypt each PDU independently." (Note: PDU stands for protocol data unit.)

Insertion of a synthesized packet or replaying of a packet sent on another connection will be detected, if a unique ID or session key is used for each connection. To detect replaying of packets sent in the opposite direction on the same connection, either different IDs (or keys) should be used in different directions, or a field indicating the direction has to be added.

For connectionless data transmission, time stamping may be used to provide a limited form of protection of the integrity of a sequence of data units.

4.3 Checksums

"Data integrity is often achieved by calculating a checksum. These checksums are associated with the data to be guarded, but are separate and distinct from checksums computed for data origin authentication or other cryptographic transformations for confidentiality. Such checksums are sometimes called manipulation detection codes." [ISO88]

Branstad [Bra87] describes an implementation with integrity mechanisms: "The PDU integrity protocol specifies how an electronic data integrity seal, called a Message Authentication Code (MAC), is computed for

each PDU. The seal covers either the user data only for simple data integrity or both the user data and the header (including sequence numbers) for data stream integrity. The seal is typically a 32-bit number that is computed using cryptographic functions on the parts of the PDU that are to be sealed so that its integrity can be verified when it is received at the corresponding security perimeter (layer 4 peer entity). ... If the value is not correct, the suspected PDU is discarded and a retransmission is requested."

"Corruption detection techniques, normally associated with detection of bit errors, block errors and sequencing errors introduced by communications links and networks, can also be used to detect message stream modification. However, if protocol headers and trailers are not protected by integrity mechanisms, an informed intruder can successfully bypass these checks." [ISO88]

5 Authentication

> "These services provide for the authentication of a communicating peer entity and the source of data..." [ISO88]

5.1 Services

The OSI Security Architecture defines two types of authentication services:

Peer entity authentication — authentication of two communicating peer entities.

Data origin authentication — authentication of the source of a single connectionless data unit.

5.2 Mechanisms

Some of the mechanisms that can be used to provide the authentication service are:

- Authentication information (e.g., passwords)
- Cryptographic techniques

Cryptographic techniques can be combined with a 'handshaking' protocol to protect against replay (i.e., to ensure the 'liveness' of the connection). The authentication mechanisms will often be used in conjunction with:

- Timestamps and synchronized clocks
- Two- and three-way handshakes
- Non-repudiation services (e.g., digital signatures, logs, etc.)

5.3 Data origin authentication

This section discusses an article by Francois Law Min *et al.* [MPH88]

They use data origin authentication together with connectionless data transfer. Mechanisms used are:

- A key distribution center (KDC)
- A public key cryptosystem
- A one-way (public) function, f, to be used on messages, M.
- Public key certificates (PKC) issued by the KDC.

Each user and the KDC have a secret and a public key to be used with the public key encipherment algorithm. Let the encryption with a public key K be called E_K and the decryption with a secret key K be called D_K. A public key belonging to user A is called E_A and a secret key is called D_A.

Now, if A wishes to encrypt a message M with its public key we write this as $E_A(M)$ and the decryption of M with the secret key as $D_A(M)$.

The irreversible property of f must be guaranteed, as must the computational infeasibility to find two messages, M_1 and M_2, such that $f(M_1) = f(M_2)$. Min *et al.* uses the DES algorithm to implement f. The function f itself takes two arguments, a message M and a (randomly chosen) value I.

5.3.1 Signing a message

If user A wishes to sign a message M he first computes $X := f(M, I)$. He then encrypts X and I with his *private* key D_A. The receiver B can check the validity and origin of the message by computing $f(M, I)$ and comparing the result with the received X-value. Because of the properties of f, a replay of an old signature with a new message should have very little chance of succeeding.

5.3.2 Proof of integrity

A public key certificate, *PKC*, is a proof of integrity of a public key. The *PKC* for A's public key is called PKC_A. *PKC*s are 'made' by the *KDC*, who keeps a record of all valid public keys. A *PKC* contains a timestamp, the unique identifier of A and A's public key, all encrypted under the private key of the *KDC*.

5.3.3 The protocol

So, if user A wishes to send a message M to user B in a way that B can be sure M really came from A, the following protocol is used:

$$\begin{array}{rcll} A & \longrightarrow & KDC: & Id_A \text{ and a request for } PKC_A \\ KDC & \longrightarrow & A: & PKC_A := D_{KDC}(\text{Timestamp}, Id_A, E_A) \\ A & \longrightarrow & B: & PKC_A,\ D_A(X, I) \text{ and } M \end{array}$$

A can check the validity of the *KDC* by examining the *Id* returned in the *PKC* (this prevents someone from masquerading as the *KDC*, and KDC_S must not be disclosed).

B unpacks PKC_A, checks the validity of the timestamp (according to some rule) and uses E_A to decrypt the signature and I. Now B can compute $X_1 := f(M, I)$ and check if $X_1 = X$. If so, the message originated from A.

This protocol does not provide message secrecy. To get that is fairly simple. The necessary changes are described in the article, but is of no interest in this context.

5.4 Peer entity authentication

We have found no good references on this subject. Most authors just mention the usage of a two- or three-way handshake protocol to establish a connection with the correct peer entity. Therefore, we will describe our interpretation of such a protocol (three-way handshake).

Three-way handshake

User (entity) A wishes to establish a connection with user B. Both A and B must be sure that the connection initiation takes place in real time, i.e., there is no playback of an earlier connection initiation. To do this they generate a random number which they send to each other and back, after some given manipulation is done on it (possibly a *null* operation). For this to be secure against insertion/deletion (of packets) the data must be encrypted. A generates R_1 and B generates R_2:

$$\begin{array}{lll} A & \longrightarrow & B: \ E_B(D_A(R_1)) \\ B & \longrightarrow & A: \ E_A(D_B(R_1, R_2)) \\ A & \longrightarrow & B: \ E_B(D_A(R_2)) \end{array}$$

In this scheme only the recipient can decipher the data (R_1 and R_2) since it was encrypted with the receiving partys public key. Equally certainly can the recipient be sure of the identity of the sender; when A receives R_1 from B encrypted under B_S and when B receives R_2 from A encrypted under D_A.

Apart from the described exchange, A and B must obtain each others public keys. This was a **three-way handshake**; it authenticates both communicating entities. If R_2 and the last message exchange is omitted, we have a **two-way handshake** that authenticates the receiver (B) but not the sender (A).

This is a way of doing mutual authentication with the use of a public key algorithm. Voydock and Kent [VK84] describes a protocol which involves the DES algorithm.

6 Non-repudiation

> Repudiation is the denial "by one of the entities involved in a communication of having participated in all or part of the communication". [ISO88]

6.1 Services

The OSI Security Architecture defines two non-repudiation services:

Non-repudiation with proof of origin — The recipient of data is provided with proof of the origin.

Non-repudiation with proof of delivery — The sender of data is provided with proof of delivery.

6.2 Mechanisms

Assurance of origin and/or delivery can be provided by a trusted third party notary, which holds the necessary information in a testifiable manner [ISO88], or it can be stored by the involved parties.

The basic mechanism used to provide non-repudiation is cryptography—preferably a public key system. It is usually implemented as digital signatures.

User (entity) A sends a message M to user B. A and B have a pair of encryption keys for use in a public key algorithm. E_A is the encryption of data with A's public key and D_A is the encryption of data with A's secret key. Furthermore, a hashing function h is also used. The function h takes a message as argument and gives a block X (of some size) as the result. The function h must be a one-way function and it must be impossible to find another (intelligible) message M' that gives the same result X.

Such a function can be devised with the DES algorithm. In short, the message M is broken up in 56-bit blocks and each block is used to encrypt a randomly chosen vector I. This is enough to ensure the second property of h. [Den83]

6.3 Proof of origin

Non-repudiation with proof of origin can be implemented in the following way. A computes $X := h(M, I)$. A then **signs** X by computing $S := D_A(X, I)$, and sends M and S to B. The origin of data can be proved by applying h to the message received with the I given in S. B can store S as long as a B wishes to be able to prove the origin of the message.

6.4 Proof of delivery

Proof of delivery is fundamentally the same as proof of origin, but in this case B computes X on the received message and sends $S := D_B(X)$ back

to A with the acknowledgement. A can then store S to prove a delivery if a dispute should arise.

6.5 Conclusions

This section relies on no specific article. Mechanisms (h and S) are provided by Denning, but the 'protocol' is not stated explicitly anywhere. It can be refined and extended, but the basic functionality is as stated.

A problem arises though, if the private key of a user (entity) is disclosed. Denning [Den83] elaborates on this. See also section 9 on Security Management.

7 Access Control

> "The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner." [ISO88]

7.1 Services

"Access control provides protection against unauthorized use of resources accessible..." [ISO88] For instance:

- Use of communication resources
- Read, write and/or deletion of an information resource
- Execution of a processing resource

7.2 Mechanisms

The access control mechanisms rely on the authenticated identity of an entity to enforce the access rights of the resource. Use by an entity of unauthorized resources or use of authorized resources in an unauthorized way will cause the access control mechanisms to deny access and also report the attempt (to some security supervisor) and/or generating an alarm.

Means with which to enforce an access control mechanism: [ISO88]

Access Control Information Base A list of resources available to authorized peer entities. This assumes that successful peer entity authentication has been done. The list can be an access control list, matrix or a hierarchical/distributed structure.

Authentication information The most common example of this is the good ol' password, in some form.

Capabilities The users of a computer system may have different capabilities, allowing them access to various types of resources.

Security labels A resource may have a label which determines the kind of access that is allowed on itself.

Timestamps A time, or duration, of an (attempted) access to a resource.

Time/route Duration and time of an attempted access, and perhaps by which route the attempt was made.

If a connectionless channel is used to access a resource, it must be known at the origin (of the request) which requirements the access control mechanism at the receiving end has, and supply enough information so that mechanism can grant access to the resource.

7.3 Other issues

In an example from 'real life', Blatchford [Bla87], one big stumbling block is the administration of the security mechanisms when they grow outside a well defined area (i.e., a house of complex of houses). It is not always obvious who, or what department, should do a particular task (e.g., key distribution). The extra work of surveillance of audit trails may be laid on the internal auditor, who will then find him/herself with a significant expansion of responsibilities. Note that these are administrative issues, not programmatic ones. Blatchford also mentions the problem with human beings and passwords. His conclusions are, that the mechanisms today involved in assigning, changing and protecting passwords (as an example of a mean by which privileges are bestowed) are far from satisfactory. One shortcoming is the lack of imagination used when a normal user chooses passwords...

7.4 Conclusions

Access control mechanisms are not easily generalized. Some exists today (e.g., passwords of some kind, or access control information to files), but they need to be expanded and improved upon in order to fit in a distributed environment. In addition to changes in communication protocols, the information of *how* the resources are accessed (and by whom) must be processed. This processing of information is an important part of enforcing the security policy and of being able to take precautions against attacks to the safety of the system.

8 Placement of Services

This section briefly describes which layers in the OSI model that are suitable for the placement of the various security services.

8.1 Possible placements

There are those who argue that the placement of security services should be within a sub- or super-layer, and those who argue that the services should coexist within the layer. Another disagreement concerns whether to offer a service at all possible layers or to minimize the number of places where to offer the services. [Tar85]

[ISO88] lists the following possible placements for security functions:

Physical layer

Connection confidentiality

Traffic flow confidentiality

Data link layer

Connection confidentiality

Connectionless confidentiality

Network layer

Peer entity authentication

Data origin authentication

Access control service

Connection confidentiality

Connectionless confidentiality

Traffic flow confidentiality

Connection integrity without recovery

Connectionless integrity

Transport layer

Peer entity authentication

Data origin authentication

Access control service

Connection confidentiality

Connectionless confidentiality

Connection integrity with recovery

Connection integrity without recovery

Connectionless integrity

Application layer

Peer entity authentication

Data origin authentication

Access control service

Connection confidentiality

Connectionless confidentiality

Selective field confidentiality

Traffic flow confidentiality

Connection integrity with recovery

Connection integrity without recovery

Selective field connection integrity

Connectionless integrity

Selective field connectionless integrity

Non-repudiation with proof of origin

Non-repudiation with proof of delivery

In the OSI Security Architecture, the provision of any security service is optional, depending upon requirements.

It would be too expensive to provide all security services at all possible layers. However, the goal of compatibility between peer layers would not be achieved if different implementors would chose different layers for the same services. [Bra87]

8.2 Traffic flow confidentiality

Full traffic flow confidentiality can be achieved only at layer 1 and only if the transmission path is two-way simultaneous and synchronous.

Above the physical layer, full traffic flow security is not possible. Some of its effects can be partly produced by the use of a complete SDU confidentiality service at one layer and the injection of spurious traffic at a higher layer. Such a mechanism is costly, and potentially consumes large amounts of carrier and switching capacity.

If traffic flow confidentiality is provided at layer 3, traffic padding and/or routing control can be used. Routing control may provide limited traffic flow confidentiality by routing messages around insecure links or subnetworks. The incorporation of traffic padding into layer 3 (instead of higher layers) enables better use of the network to be achieved, for example, by avoiding unnecessary padding and network congestion.

Limited traffic flow confidentiality can be provided at the application layer by the generation of spurious traffic, in conjunction with confidentiality to prevent identification of the spurious traffic. [ISO88]

8.3 Confidentiality

Electrical insertion of transparent pairs of transformation devices in layer 1 can give complete confidentiality upon a physical connection.

Confidentiality services can be placed in layer 3, for subnetwork access role over individual subnetworks and for relay and routing roles over the internetwork.

Since the individual transport connection gives an end-to-end transport service and can provide isolation of session connections, layer 4 could be chosen.

It is also possible to provide confidentiality in layer 7 in conjunction with mechanisms at lower layers. [ISO88]

8.4 Integrity

Layer 1 has no error detection or recovery mechanisms, and the layer 2 mechanism operates only on a point-to-point basis, not an end-to-end basis, and therefore, it is not considered appropriate to provide this service at these layers.

Only integrity without error recovery is available in layer 3, for subnetwork access role over individual subnetworks and for routing and relay roles over the internetwork.

Layer 4, on the other hand provides the true end-to-end transport connection and is therefore a suitable place for integrity mechanisms.

Integrity can also be provided by layer 7, in conjunction with mechanisms in the presentation layer.

The integrity of connectionless transfers should be provided only at the same layers as for integrity without recovery, in order to minimize the duplication of functions. [ISO88]

Providing detection measures in a communications protocol obviates the need for each application programmer to devise an application-specific means of detection. [VK85]

8.5 Authentication

Peer entity authentication services can be placed in layer 3 for authentication over individual subnetworks and for routing over the internetwork. Placement in layer 4 is appropriate for end-to-end system authentication prior to the commencement of a connection, and for the duration of that connection.

Data origin authentication can be provided end-to-end in the relay and routing role of layer 3 and/or in layer 4 by peer entity authentication in conjunction with continuous encipherment. Without peer entity authentication, data origin authentication can be provided with very little overhead to the data integrity mechanisms already present.

It is also possible to place authentication services in layer 7, possibly in conjunction with encipherment mechanisms provided in layer 6. [ISO88]

8.6 Selective field protection

Integrity and confidentiality in selected fields is only practical as part of the application layer, in conjunction with mechanisms in the presentation layer. [ISO88]

Encipherment at layer 6 provides security for selected fields of data specified by an application in such a way, that the data need not be unprotected even at the intended destination. [Bra87]

8.7 Non-repudiation

Origin and delivery non-repudiation services can be provided by a notarization mechanism which will involve a relay at layer 7.

Use of the signature mechanism for non-repudiation requires a close cooperation between layers 6 and 7. [ISO88]

8.8 Access control

In layer 3, access control mechanisms may be imposed on the subnetwork access role by the requirements of a particular subnetwork. When performed by the relay and routing role, access mechanisms in the network layer can be used both to control accesses to subnetworks by relay entities and to control access to end systems.

Access control for end-to-end connections can be placed in layer 4, and application-oriented access control in layer 7. [ISO88]

8.9 Position of encipherment

Most applications will not require encipherment to be used at more than one layer. The choice of layer depends on some major issues as described below:

- Full traffic flow confidentiality requires encipherment at the physical layer.
- If a high granularity of protection and non-repudiation or selective field protection is required then presentation layer encipherment will be chosen.
- If simple bulk protection of all end-to-end system communications is desired, then network layer encipherment will be chosen.
- If integrity with recovery is required together with a high granularity of protection, then transport layer encipherment will be chosen. This can provide confidentiality and integrity, with or without recovery. [ISO88]

8.10 The transport layer

Voydock and Kent [VK84, VK85] suggest the transport layer as the appropriate place for the security mechanisms. The layer is the lowest one that can offer end-to-end-security. Also, the error recovery mechanisms are easy to augment to support security. A higher placement would considerably complicate the protocol and, for the most part, duplicate already present mechanisms.

While present EFT (electronic funds transfer) security standards specify security services at layer 7, a wide variety of other applications could utilize similar security services if they are implemented at layer 4. The integrity service protocol utilizes the sequence number provided by layer 4, class 4 protocol. This is a 31-bit number defined as 4 octets in the header of each layer 4 PDU. The sequence number is provided by layer 4 for resequencing the PDUs if they arrive out of order and for flow control on a connection. The integrity service also utilizes the existing layer 4, class 4 mechanism for recovery from errors (that is, lost or modified data). Connectionless network layer services can then be used if a class 4 integrity service is provided and used at layer 4. [Bra87]

8.11 The presentation layer

The major advantage of presentation over transport is that presentation can selectively protect data. The major disadvantage over transport is the lack of potential recovery. The major advantage over the application layer is that other transformations offered in the presentation layer (such as data compression) may be considerably reduced in effectiveness if performed on enciphered data. [Tar85]

9 Security Management

Security management in the OSI Security Architecture [ISO88] consists of activities in the following categories:

System security management is the management of security aspects of the overall OSI environment.

Security service management is the management of particular security services.

Security mechanism management is the management of particular security mechanisms.

Security of OSI management is the security of all OSI management functions and of the communication of OSI management information.

In this section we focus on **key management** (an activity of security mechanism management) and, to a certain extent, **security audit trails and event handling** (activities of system security management).

Key management is a fairly large topic of which we only are interested in some parts. These parts are distribution of public keys in a public-key cryptosystem, and the distribution of session keys for individual communication sessions in the framework of the public-key cryptosystem.

9.1 Article overview

The most explicit article we have found on the subject of key management is written by Denning [Den83], who describes a complete and secure key distribution scheme for public keys. The scheme uses a key server (or several servers) which manages a public-key directory. The key server is assumed to be trustworthy and to distribute correct and current public keys. No assumptions are, however, made about the security of the network or the hosts on the network. Mechanisms for detecting and recovering from compromises are also included.

Voydock and Kent [VK84] write about distribution of session keys in the transport layer of the OSI model. They assume that each transport entity has a master key and does not address the problem of distributing it.

A very different method for distributing keys is presented by Quisquater [Qui87]. A "secret-key" cryptosystem and a trusted authority is used for distributing session keys.

Newman et al. [NOP87] discuss key distribution with two public-key cryptosystems: the RSA system and the SEEK system. In the RSA system a method similar to, but not as complete as, Denning is used. The SEEK system uses the Pohlig-Hellman encryption scheme which is based on computing exponentials over a finite field. A secret number is distributed between two users by exchanging exponentiated public numbers in the clear. The article discusses the need for certification of these public numbers and of the public keys of the RSA system.

9.2 Denning

Denning [Den83] identifies the following security threats to public keys:

- Fake public keys
- Compromise of a user's private key
- Compromise of the server's private key

9.2.1 Fake public keys

A penetrator can distribute fake public keys. A fake public key is a key claimed (by the penetrator) to be someones public key, but really is a public key of the penetrator. The countermeasure for fake public keys is for the key server to distribute the public keys inside signed certificates. A **public-key certificate** for the public key of user A has the form:

$$P := D_S(T, A, E_A).$$

It contains a timestamp T, the identity of A and A's public key E_A. The message is signed by encrypting with D_S, the private key of the server. The receiver of the certificate P can easily check the validity of A's public key by decrypting the certificate with the servers public key.

9.2.2 Compromise of a user's private key

If the private key of a user A becomes known to a penetrator, the security system must be able to handle and recover from the situation. Compromises must be reported to the key server, and new public keys must be registered. This protects messages encrypted for security, but not signatures.

To protect signatures also, a method for validating signatures is needed. This method must work even if the private key used for signing is compromised after the signature was made. Denning's method uses **signature certificates** created by the key server. These certificates are like public-key certificates, but with signatures inside. If $X = D_A(\overline{M}, I)$ is the signature of A for message M, where $\overline{M}$ is a cryptographic checksum for the message and I is a random initialization seed for the checksum, then the signature certificate (created by the key server) for X is:

$$G := D_S(T, A, E_A, X).$$

The timestamp T is essential for determining whether a message was signed before the private key of A was compromised. The key server appends T before making the signature by encrypting with its private key.

9.2.3 Compromise of the server's private key

The far most severe security threat is the compromise of the key server's private key. If this happens, the penetrator can forge public-key certificates,

and if he also knows the private key of some user, he can forge signature certificates for this user. The penetrator can continue to do this even after the server's key has been changed. Denning proposes a method which uses an **on-line certificate log** which handles compromises of all keys. The next section describes the method.

9.2.4 The certificate log

The key server keeps a log or audit trail of the following events:

- Registration of public keys, noted by entry of public-key certificates in the log.
- Registration of signatures, noted by entry of signature certificates in the log. (Signatures are not legally binding unless they are logged.)
- Notification of key compromises.

The log should be stored on a write-once device such as an optical disk, to prevent the log from being altered. The entries in the log are timestamped by the key server and only the server should be able to write new entries. No secret information is kept in the log, and anyone should be allowed to read it. The key server appends all new certificates to the log when they are created.

The correct public key for a user is given by the most recent public-key certificate in the log, if it's not followed by a "compromise" entry for the user.

When the user A detects that its private key is compromised, A signs and sends the message $M :=$ "compromise" to the key server, which appends the record $D_S(T, A, E_A, X)$, where X denotes the signature for M.

A record is also appended to the log when the server's key is compromised. This does not prevent false information to be appended to the log between the time of compromise and the time of detection, assuming that the compromiser *can* write to the log, which may not be possible. But it does limit the time stamps that can be placed in the false entries to that time period. It will, however, not be possible to forge a signature certificate for a previously compromised key D_A with a time stamp T when D_A was valid.

The log does not solve all problems arising from key compromises, but it does confine them substantially, and it allows for their detection.

9.3 Quisquater

The method for distributing session keys used by Quisquater [Qui87] has an unusual cryptosystem and a trusted authority. The cryptosystem is an "identity-based" system, which means that the identity of a user is used to compute the key for that user. The system assumes that each sender/receiver has a tamperfree device which handles all keys and all encryption/decryption using these keys.

The trusted authority has a number of "supersecret" keys (for instance 20) and distributes a few (for instance 3) of them to each user. Each user has a unique set of these keys, which means that there is a limit to the number of users. The distribution of the keys are held secret. The authority also sends each user his/hers identity encrypted by each of the 20 supersecret keys.

Let $K_1, K_2, K_3, \ldots, K_{20}$ be the supersecret keys of the authority, I_A be the identity of user A and K_{A_1}, K_{A_2} and K_{A_3} be the subset of the supersecret keys given to user A. Let also $k_1^A, k_2^A, k_3^A, \ldots, k_{20}^A$ be the identity I_A encrypted by the respective supersecret key. These secret values are computed by the authority and sent to A.

For A to send a secret session key r to B, A first encrypts I_B with his/hers subset of the supersecret keys to get $k_{A_1}^B$, $k_{A_2}^B$ and $k_{A_3}^B$. These values are then used as keys to repeatedly encrypt the session key r to obtain R, that is,

$$R := E_{k_{A_3}^B} \circ E_{k_{A_2}^B} \circ E_{k_{A_1}^B}(r).$$

Note that B knows all of $k_1^B, k_2^B, k_3^B, \ldots, k_{20}^B$ and therefore can decrypt R to get the session key r, but B needs a little help to know which keys to use. For this purpose A computes an *indicator* σ from I_B in the same way that R is computed from r. The indicator σ and the encrypted session key R is sent to B. After a maximum of $\binom{20}{3}$ tries, which equals 1140, B can find out which keys to use to decrypt σ, and thus also R, and eventually obtain r.

10 Protocols

This section summarizes four articles describing more or less "complete" communication protocols with security services. The authors of all four articles has chosen to extend a **transport protocol.**

Voydock and Kent [VK84] extend the transport layer of the OSI model to include a "secure" class, an enhancement of the existing class 4. Four versions of connection establishment is described and compared. They differ in the way the working key of the connection is obtained, or how the transport entity communicates with the DES hardware.

Diffie [Dif85] describes an extension to the U. S. Department of Defense Transmission Control Protocol (TCP). Three versions of the extended protocol are compared with respect to compatibility with non-secure TCP and the security service provided.

Tardo [Tar85] writes about the work done by ANSI to standardize cryptographic data protection services in the OSI transport layer protocol.

Branstad *et al.* [BDHR87] describe a security protocol at layer 4 of the OSI model called SP4. The protocol was developed by a working group of the Secure Data Network System (SDNS) project, which also includes security services in the application, network and physical layers of the OSI model.

10.1 Comparison of service

Service	Voydock and Kent	Diffie	Tardo	Branstad
Confidentiality	connection, limited traffic flow	connection	connection	connection or connectionless
Integrity	connection with recovery	connection with or without recovery	connection with recovery	connection with recovery or connectionless
Authentication	peer-entity	peer-entity	peer-entity	peer-entity
Access control	no	no	no	yes
Non-repudiation	no	no	no	no

Table 1: Comparison of service in the four articles.

Table 1 compares the security services provided by the protocols described in the four articles. The authors have chosen almost the same services, but there are differences when the protocols are examined more closely, especially concerning the peer-entity authentication service.

10.2 Voydock and Kent

The protocol by Voydock and Kent [VK84] is based on the OSI transport layer protocol, class 4. A new "secure" class is defined which makes use of the cipher block chaining (CBC) mode of the Data Encryption Standard

(DES). A separate encryption key is used for each transport connection—the "working key". A transport entity obtains the working key either from a key distribution center (KDC), or from the transport user.

10.2.1 TPDU encryption

A transport connection is identified by a pair of "reference numbers" assigned by the respective transport entity. The number assigned by the receiving entity is contained in the "destination reference number" (DST-REF) field in the header of the transport protocol data unit (TPDU). This number is used by the receiver to determine which transport connection the TPDU belongs to and is therefore sent in the clear. To simplify encryption and minimize the amount of unencrypted data, the DST-REF field is placed in the first two octets of the TPDU.

Since TPDUs can arrive out of order, each TPDU is independently encrypted. A pair of distinct initialization vectors (IVs) is used for each connection, one for the normal data stream and one for the expedited. The IVs must be protected from disclosure.

10.2.2 Integrity

To provide the integrity service the following measures are taken:

- A 16-bit cyclic error detection code is included in the encrypted portion of each TPDU.
- Each connection is assigned a unique working key.
- A field indicating direction is added in the TPDU header. This field is used to detect playback of TPDUs previously sent in the other direction.
- The working key of the connection is changed whenever the normal data or expedited data sequence number is about to cycle. The secure class always uses the extended (31-bits) sequence numbers to prevent rekeying from occurring often. Rekeying is needed to detect the replaying of, for example, the first TPDU as TPDU number 2^{31}, which both have sequence number 0.

10.2.3 Denial of service

The secure class includes a measure for detecting a more subtle form of denial of service attack in addition to the measures of the class 4 protocol. In this kind of attack, the intruder replays old AK TPDUs to prevent the inactivity timers of the transport entities from timing out and at the same time hinder the delivery of other TPDUs. To be able to distinguish replayed AK TPDUs from AK TPDUs generated when the window timer times out, a unique identifier (UID) parameter is added to every AK generated by the window timer.

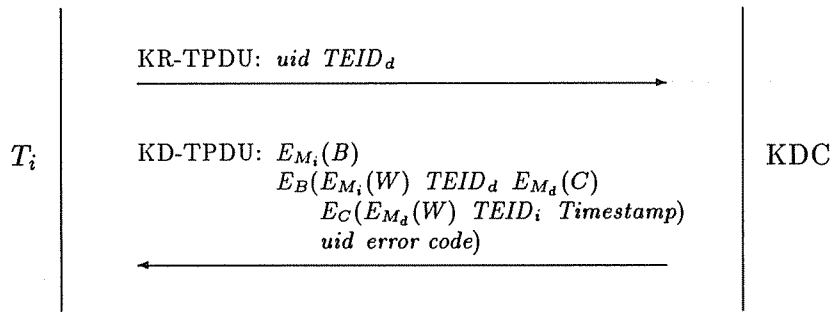

Figure 1: KDC/Transport Entity Protocol.

10.2.4 Connection initiation

The article presents four alternatives for the connection initiation. They differ in the way the working key of the connection is obtained (from a KDC, or from the transport user), and in the way the transport entity communicates with the DES hardware.

Here we concentrate on the first alternative in the article: with a KDC and a DES peripheral with an associated *master key*. An optional key parameter can be specified to the peripheral. If a key isn't specified, the master key of the peripheral is used. Key parameters are stored on the host encrypted under the master key. The DES peripheral decrypts the received keys with the master key to obtain the actual key to use. Each transport entity has a master key associated with it, corresponding to the master key of the DES peripheral. The master key is *not* stored on the host i any form. The KDC is a special transport entity preferably on a dedicated trusted host. The KDC maintains a table with the master keys of the transport entities in the network.

Let T_i be the initiating transport entity (that is, the entity which received a T-CONNECT.request from its user), T_d be the destination transport entity, $TEID_i$ and $TEID_d$ be their identifiers (network addresses) and let M_i and M_d be their master keys.

Figure 1 shows the protocol between the KDC and T_i. Two new TPDUs are used: the *Key Request* (KR) TPDU and the *Key Deliver* (KD) TPDU. The KR TPDU contains $TEID_d$ and *uid*, a 64-bit unique identifier that is used by T_i to verify the time integrity of the KD TPDU. When the KDC receives the KR TPDU, it looks up M_i and M_d and generates two *temporary keys* B and C. They are used to reduce the amount of information encrypted with M_i and M_d. It also generates W, the working key for the connection and then sends the KD TPDU to T_i. The portion of the KD TPDU encrypted with the key C works like a certificate for the working key and the identity of T_i.

The initiating transport entity is now ready to establish the connection. The protocol is shown in figure 2. The normal TPDUs from class 4 are

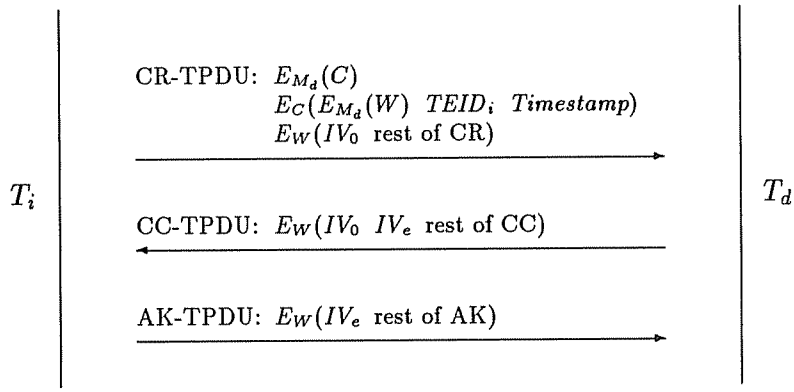

Figure 2: Connection Initiation Protocol.

used here with some modifications. In addition to its normal contents, the CR (Connect Request) TPDU contains the 'certificate' received from the KDC and an initialization vector, IV_0, for the encryption of the normal data stream and which also is used to authenticate the connection for the initiator. The destination transport entity, T_d, responds with a CC (Connect Confirm) TPDU containing IV_0, the same as received, and IV_e, the initialization vector for the expedited data stream, which also is used to authenticate the connection for the destination entity.

10.2.5 Changing the working key

The working key has to be changed when the sequence spaces of the normal or expedited data streams are exhausted, or when the AK unique identifier described above in section 10.2.3 is about to cycle. When rekeying is necessary, one of the transport entities requests and receives a new working key from the KDC in the same manner as above. Two new TPDUs are used to establish the new key: the *Initiate Key Change* (IK) TPDU, and the *Key Change Confirm* (IC) TPDU. The transport entity which received the new key from the KDC sends a IK TPDU to the other entity. The entities now waits until all old TPDUs have been received correctly in order to avoid problems with duplicate keys. Then the other entity responds with the IC TPDU to confirm the change of key.

10.3 Diffie

The protocol by Diffie [Dif85] is based on the well known TCP. Three versions of a secure extension of TCP is presented:

- Full compatibility with existing TCP.
- Upward compatible extension.
- Incompatible, but related, protocol.

Diffie doesn't choose any particular cryptosystem to use, but instead discusses the pros and cons of public-key and conventional systems. The protocol uses a cryptosystem in a "public-key style", which doesn't require a public-key system. In the protocol the sender uses a "sending key", and the receiver uses a "receiving key".

The cryptosystem is assumed to operate in one of the following modes:

- **Cipher block chaining** mode.
- **Cipher feedback** mode.
- Synchronous modes such as **counter driven** mode or **output feedback** mode.

In both the cipher block chaining and cipher feedback modes, each block of encrypted data depends on the preceding blocks as well as the encryption key. To be able to decrypt TCP segments[2] independently, an initialization vector is included in each segment. These initialization vectors must be sent in the clear to be of any use to the receiver. The article is a little unclear about this—the figures indicate that the vectors are encrypted.

All three versions of the protocol tries to encrypt as much as possible of each TCP segment. The version with full compatibility with the normal TCP cannot encrypt any part of the segment header in contrast to the most secure version which encrypts all of the header.

A key manager provides each TCP-entity with one or more session keys (depending on the cryptosystem) prior to connection establishment. The normal TCP connection establishment is complemented with a two way challenge-response procedure for authenticating the TCP-entities.

A cryptographic checksum computed over the whole TCP segment provides the integrity service.

10.4 Tardo

The article by Tardo [Tar85] presents the "highlights" of the draft security addendum to the OSI transport service proposed by ANSI. The services proposed are:

- Mutual connection verification
- Data confidentiality
- Data integrity (including origin authentication)

Tardo writes: "It is important to note that these services can be provided *orthogonally*, that is, independent of each other and selectable in any combination. Also, note that they are conceptualized as existing within the layer and not in an independent sublayer." The services *extends* the existing class 4—a new class is not proposed.

[2] A TCP segment corresponds to a protocol data unit (PDU) in the OSI-model.

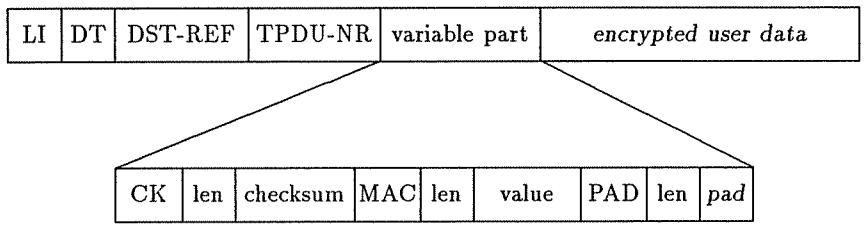

Figure 3: Cryptographically Protected Class 4 Data TPDU.

The cryptosystem used is a block cipher run in CBC mode. As can be seen in figure 3 the user data of a Data (DT) TPDU is encrypted, and the fixed part of the TPDU header is transmitted in the clear. The fields of the variable part of the header may be encrypted, but the type and length fields are always in the clear. An authentication code ("MAC") and encrypted padding bytes are added to the variable part of the TPDU header (see figure 3). All TPDUs have an authentication code for integrity protection.

The protocol uses the extended (31-bit) format of the TPDU sequence number. The sequence number is *not* allowed to cycle, since this would introduce a weakness in the confidentiality and integrity services. The connection must be broken and reestablished in this case.

A single initialization vector, IV, for the CBC mode of encryption is used for an entire connection. By XORing the fixed portion of the TPDU header and the connection IV, a unique IV is derived for each TPDU.

Two kinds of keys are used: key encryption keys (KEK) and data encryption keys (DEK). The transport entities has to share a secret KEK in order to establish a secure connection. The protocol doesn't specify how to generate and distribute KEKs.

The connection verification service is provided with a two-way challenge–response exchange which uses the shared KEK. The exchange is augmented the normal three way handshake performed at connection establishment. The DEK for the connection is a random value generated by the initiating transport entity and sent to the other entity as part of the challenge-response exchange.

10.5 Security Protocol 4

The article by Branstad *et al.* [BDHR87] provides an overview of SP4, a transport *encapsulation* protocol developed by the Security Data Network System (SDNS) project within the OSI-model of ISO. Encapsulation means in this case that the security operations are applied on each TPDU as the last step in the transport layer and that the format of the TPDUs are not modified.

SP4 is designed to be independent of encryption algorithm and key distribution method. In conjunction with SDNS, the key manager of SDNS is used to generate traffic keys for the communication. The traffic keys can have different "granularities", for example, one key per NSAP (network service access point) or one key per transport connection.

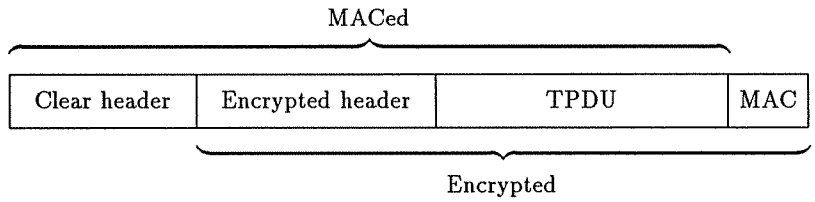

Figure 4: SE PDU Format.

The security services provided by SP4 are:

- Confidentiality
- Connection-oriented and connectionless integrity
- Access control
- Peer entity authentication

The TPDUs of the transport layer are packaged in a security (SE) PDU by the security protocol (see figure 4) before they are sent to the network layer.

The clear header contains four fields:

- A length indicator pointing to the beginning of the encrypted information.
- A PDU type field—always "SE" for SP4 PDUs.
- A key identifier field used by the receiver to select the decryption key.
- An initialization vector, IV, used to initialize the decryptor.

The encrypted header contains the following fields:

- A length indicator pointing to the beginning of the TPDU.
- A security label indicating the sensitivity of the data in the TPDU. (Optional)
- Final sequence numbers indicating the final sequence number sent and received. This field is included in the closing PDUs of the connection.
- A pad field used when the encryption algorithm requires.

The integrity service is mainly provided by the message authentication code (MAC) which uses the same key as the encryption/decryption. The MAC is computed before encryption and checked after decryption. The final sequence numbers are used to detect deletion of TPDUs at the end of the connection. The authors write that protection against replay is obtained if the TPDU sequence numbers do not wrap under the connection key, and that a new key must be obtained from the key manager in this case. Protection

against reflection is provided if different key identifiers are used for each direction.

The SDNS key manager provides part of the access control service. SP4 provides access control via security label checking.

Peer entity authentication is provided by the key manager.

A Definitions

The following selected definitions are mainly taken from the OSI Security Architecture [ISO88]. Additional definitions can be found in [ISO84], [ISO88], [NCSC87], and [AP87].

Access control: The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner.

Access control list: A list of entities, together with their access rights, which are authorized to have access to a resource.

Active threat: The threat of a deliberate unauthorized change to the state of the system.

Audit trail: Data collected and potentially used to facilitate a security audit.

Authentication: The corroboration, that a peer entity in an association is the one claimed, or in the context of connectionless service, that the source of data received is as claimed.

Confidentiality: The property that information is not made available or disclosed to individuals, entities, or processes.

Covert channel: A channel of information that is not intended to be a communication channel.

Digital signature: Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery.

Discretionary access control (DAC): A means of restricting access to objects based on the identity of subjects. A subject with a certain access permission is capable of passing this permission on to any other subject.

End-to-end encipherment: Encipherment of data within or at the source end system, with the corresponding decipherment occurring only within or at the destination end system.

Event-handling: Detection and reporting of security-relevant events.

Integrity: The property that data has not been altered or destroyed in an unauthorized manner.

Key management: The generation, storage, secure distribution and application of keys in accordance with a security policy.

Link-by-link encipherment: The individual application of encipherment to data on each link of a communications system.

Mandatory access control: A means of restricting access to objects based on the sensitivity of the information contained in the objects and the formal authorization of subjects to access information of such sensitivity.

Masquerade: A type of attack where an entity pretends to be another entity.

Message Authentication Code (MAC): An electronic data integrity seal usually in the form of a cryptographic checksum.

Notarization: The registration of data with a trusted third party that provides for future recourse to the data and assures accuracy concerning its characteristics such as content, origin, time and delivery of the data.

Object: (In the Red book) A passive entity that contains or receives information, such as files, directories, programs, processors, printers, etc.

Padding: The generation of spurious instances of communication, spurious data units and/or spurious data within data units.

Passive threat: The threat of unauthorized disclosure of information without changing the state of the system.

Peer entities: Entities within the same layer.

Protocol data unit (PDU): A unit of data specified in an N-protocol and consisting of N-protocol-information and possibly N-user-data.

Privacy: The right of individuals to control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

Repudiation: Denial by one of the entities involved in a communication of having participated in all or part of the communication.

Security audit: An independent review and examination of system records and activities in order to test for adequacy of system controls, to ensure compliance with established policy and operational procedures and to recommend any indicated changes in control, policy and procedures.

Security Management Information Base (SMIB): The storage place for security-relevant information, such as public keys and access rights.

Selective field protection: The protection of specific fields within a message which is to be transmitted.

Service data unit (SDU): An amount of N-interface-data whose identity is preserved from one end of an N-connection to the other.

Subject: (In the Red book) An active entity, such as a person, process or device, that causes information to flow among objects.

Trap door: A hidden software or hardware mechanism that permits system protection mechanisms to be circumvented. It is activated in some non-apparent manner (e.g. special "random" key sequence at a terminal).

Traffic analysis: The inference of information from observation of traffic flows (presence, absence, amount, direction and frequency).

Traffic flow confidentiality: A confidentiality service to protect against traffic analysis.

Trojan horse: This is when an entity has an unauthorized effect in addition to its authorized function, e.g., a relay which also copies messages to an unauthorized channel.

Trusted computing base (TCB): The totality of protection mechanisms within a computer system—including hardware, firmware and software—the combination of which is responsible of enforcing a security policy.

B Recommended Reading

We recommend the following literature on network security to be included in your bookshelf:

ISO 7498-2—OSI Security Architecture: [ISO88] The framework for security in the OSI reference model.

Trusted Network Interpretation, The Red book: [NCSC87] Interprets and extends the TCSEC [DoD83] for computer networks.

Tutorial—Computer and Network Security: [AP87] "This tutorial text is written for those who are concerned with security in automated systems, and those who should be concerned." Primarily a collection of recent articles on computer and network security which gives a wide view at the field.

Cryptography and Data Security: [Den82] The basics of cryptography and data security.

Computer Magazine, February 1983: [IEEE83] The whole issue is devoted to network security. Among others it includes Denning's article on protecting public keys and signature keys.

10th NCSC Proceedings: [NBS87] The proceedings from the 10th National Computer Security Conference in september 1987.

References

[AJ87] Marshall D. Abrams and Albert B. Jeng. Network security: Protocol reference model and the trusted computer system evaluation criteria. *IEEE Network Magazine*, 1:24–33, April 1987.

[AP87] Marshall D. Abrams and Harold J. Podell. *Tutorial—Computer and network security.* IEEE Computer Society Press, 1987. IEEE Computer Society order number 756.

[BDHR87] Dennis Branstad, Joy Dorman, Russell Housley, and James Randall. SP4: A transport encapsulation security protocol. In *10th National Computer Security Conference Proceedings*, pages 158–161. National Bureau of Standards/National Computer Security Center, U.S. Government Printing Office, September 1987.

[Bla87] Clive W. Blatchford. Security lapses under attack by worried network users. *Data Communications*, pages 191–200, September 1987.

[Bra87] Dennis K. Branstad. Considerations for security in the OSI architecture. *IEEE Network Magazine*, 1:34–39, April 1987.

[Den82] Dorothy E. Denning. *Cryptography and Data Security.* Addison-Wesley Publishing Company, Inc, 1982.

[Den83] Dorothy E. Denning. Protecting public keys and signature keys. *Computer*, 16(2):27–35, February 1983.

[Dif85] Whitfield Diffie. Security for the DoD Transmission Control Protocol. In *Advances in cryptology, Proceedings of CRYPTO '85*, pages 108–127. Springer-Verlag, Berlin, 1985. Lecture Notes on Computer Science no. 218.

[DoD83] U.S. Departement of Defence, Computer Security Center. *Trusted Computer System Evaluation Criteria,* the "Orange book", August 1983. CSC-STD-001-83.

[IEEE83] *Computer* magazine, volume 16, number 2. IEEE Computer Society, February 1983.

[ISO84] ISO, International Organization for Standardization. International Standard 7498-1, *Information processing systems – Open Systems Interconnection – Basic Reference Model*, 1984. Ref. No. ISO 7498-1984.

[ISO88] ISO, International Organization for Standardization. International Standard 7498-2, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security architecture*, 1988. Ref. No. ISO 7498-2-1988 (E).

[Kar85] Paul A. Karger. Authentication and discretionary access control in computer networks. *Computer Networks and ISDN Systems*, 10:27–37, 1985.

[MPH88] Francois Law Min, Ahmed Patel, and Jan Heijnsdijk. Provision of a data origin authentication service within the OSI framework. In *EUTECO '88*, pages 507–522. Elsevier Science Publishers B.V. (North-Holland), 1988.

[NBS87] National Bureau of Standards/National Computer Security Center. *10th National Computer Security Conference Proceedings.* U.S. Government Printing Office, September 1987.

[NCSC87] National Computer Security Center. *Trusted Network Interpretation of the TCSEC,* the "Red book", July 1987. NCSC-TG-005.

[NOP87] David B. Newman, Jr., Jim K. Omura, and Raymond L. Pickholtz. Public key management for network security. *IEEE Network Magazine*, 1:11–16, 1987.

[Pri87] Wyn L. Price. Standards for data security – A change of direction. In *Advances in cryptology, Proceedings of CRYPTO '87*, pages 3–8. Springer-Verlag, Berlin, 1987. Lecture Notes on Computer Science no. 293.

[Qui87] Jean-Jacques Quisquater. Secret distribution of keys for public-key systems (Extended abstract). In *Advances in cryptology, Proceedings of CRYPTO '87*, pages 203–208. Springer-Verlag, Berlin, 1987. Lecture Notes on Computer Science no. 293.

[Sch88] Ingrid Schaumüller-Bichl. The EUREKA-project "OASIS" and its technological challenge. In *EUTECO '88*, pages 467–474. Elsevier Science Publishers B.V. (North-Holland), 1988.

[Tar85] Joseph J. Tardo. Standardizing cryptographic services at OSI higher layers. *IEEE Communications Magazine*, 23:25–29, July 1985.

[VK84] Victor L. Voydock and Stephen T. Kent. Security mechanisms in a transport layer protocol. *Computer Networks*, 8:433–449, 1984.

[VK85] Victor L. Voydock and Stephen T. Kent. Security in high-level network protocols. *IEEE Communications Magazine*, 23:12–24, July 1985.