A Distributed System Security Architecture: Applying The Transport Layer Security Protocol

Mohammad Mirhakkak
mmirhakk@mitre.org

The MITRE Corporation
Washington C3 Center
7525 Colshire Drive
McLean, VA 22102

Abstract

A great deal of attention has been given to the development of Open Systems Interconnection (OSI) security protocols in recent years. However, limited work has been dedicated to using these protocols to develop security architectures for securing distributed systems consisting of trusted computer systems communicating via untrusted networks. This paper presents an overview of the Transport Layer Security Protocol (TLSP) and discusses its application to the development of a security architecture for a generic distributed system that uses the OSI transport layer protocol. In addition, the paper discusses the security services that can be achieved as a result of employing the TLSP in a distributed system. It also examines the security functions that are expected from the end-systems, and addresses the security management functions that are required for the operation of the TLSP.

1. Introduction

Open Systems Interconnection (OSI) protocols permit the interconnection of computer systems or hosts by allowing distributed applications or processes to communicate with each other. A major requirement of such communications is to devise security controls or services to protect the information exchanged between applications or processes.

Because of this, much attention has been given to the development of OSI security protocols within the International Organization for Standardization (ISO) in conjunction with the International Electrotechnical Commission (IEC). The ISO/IEC work is performed in the Joint Technical Committee 1 (JTC1), which consists of several Subcommittees (SC) that are divided into Working Groups (WG). The first document to address OSI security was the security architecture document [1] that was developed by WG1 of SC21 and was published in 1989. This early document defines the security services and mechanisms that may be provided for secure communications, and defines positions within the 7-layer reference model where these services and mechanisms may be provided. However, this standard does not address details of the protocols that are required for the implementation of these services and mechanisms.

To address this need, several other documents are being developed by different SCs and their WGs to address such issues [2-8]. One of these standards is the Transport Layer Security Protocol (TLSP) [2] that was developed by the WG4 of SC21. The intent of this paper is to describe the application of TLSP to a distributed environment and to discuss the security services that can be achieved as a result of using TLSP. The paper will also discuss the security functions expected from the end-systems, and will address the security management functions that are required to support TLSP operation.

2. Distributed System Model

A distributed system is required to include certain security controls to protect the system from unauthorized accesses and to ensure that system interactions will not violate the system's secure operation as defined by the security policy of the distributed system. The distributed system security policy is mapped onto individual security policies of components which define the roles of individual components in providing protection against unauthorized interactions. Such protection requires a security infrastructure for the distributed system, which is supported by security controls provided by individual components (hosts, routers, gateways, or other). These controls are conceived by the hardware and software of individual components and the secure communications that allow secure exchange of information among the distributed system's components.

Such a distributed system security infrastructure assumes the security controls are distributed throughout the distributed system to support distributed security functions, with each component trusting others to perform their specified security functions. These components use OSI services to communicate with each other and rely on communication security services such as authentication or data integrity, supported by cooperating components, to provide secure communications with each other.

2.1 Distributed System Reference Monitor

The collection of security controls in a stand-alone computer system, which are responsible for enforcing the security policy of the computer system constitute its reference monitor [9]. A stand-alone component's reference monitor provides the functions to control interactions among three classes of entities: users, applications or processes, and application's data [10]. Active entities such as users and processes that can manipulate data are referred to as subjects, while entities that hold information or data are referred to as objects. Examples of objects include files, devices, or memory. Therefore, the function of a reference monitor is to control and authorize accesses made by subjects to objects.

These reference monitors, belonging to different components, cooperate with each other to mediate interactions that occur across a distributed system and form a reference monitor for the entire distributed system whose function is to mediate and control such accesses for the entire distributed system.

A distributed system's reference monitor is responsible for the same functions that are provided by a single component's reference monitor with the exception that the three types of entities may reside any where in the distributed system. The possible interactions in a secure component and a secure distributed system are shown in Figure 1. These interactions consist of the user logon which creates a process and attaches a user device to the process, process-to-process association in which two processes establish a relationship and cooperate in performing tasks, and finally the accesses made by a process to data. The interactions between components (distributed processing) is limited to only one type of interaction; this interaction is the process-to-process association. Process-to-process

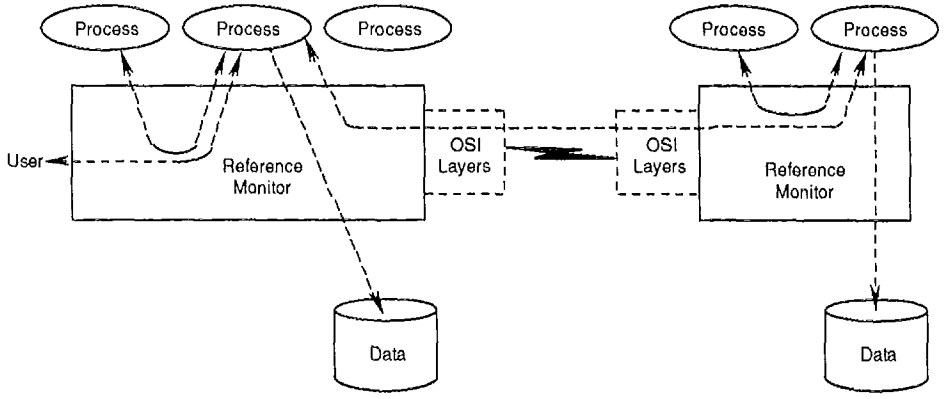

Figure 1. Distributed System Model

association begins by two processes that identify and authenticate themselves to each other and send messages between each other to request their peer to perform some specific operation and to transmit the results of the operation back to the requesting process. At the termination of the operation when there is no need for further process-to-process communications, the two processes dissolve the association.

It is the responsibility of a component's reference monitor to mediate these three possible interactions on a single component while the distributed system's reference monitor mediates these accesses on individual components as well as process-to-process associations that occur across a network. It only allows those which have proper authorization to succeed.

Each component's reference monitor contains an authorization service which maintains security information or attributes associated with local and remote users, processes, files, and other entities to determine if any of the above three accesses should be allowed. The authorization services residing at various components are combined with each other to form an authorization service for the entire distributed system. This distributed system authorization service maintains security attributes of all the entities in the entire distributed system and the rules by which a decision to allow or deny an access should be made. The cooperating and distributed reference monitors enforce the security policy of the entire distributed system and use the information in the distributed system authorization service to determine whether to allow a network-wide access request to succeed or fail.

In a single component, the authorization service of the component determines whether to allow or to deny an access by examining the security attributes of the user or the process on whose behalf the access is being requested, the security attributes of the data (object) that is being accessed, and the access policy supported by the component. Examples of access policies that may be supported by a component and the entire distributed system are the Mandatory Access Control (MAC) and Discretionary Access Control (DAC) policies [9]. The security attributes employed for MAC are usually the sensitivity levels and categories associated with the process (subject) and the object to which the process is attempting to gain access. For the DAC policy, the security attributes consist of the user identity of the process, the identities of the users that can access a given object and the type of access that they are authorized to perform.

The security attributes do not have to be statically stored at a component. Each component can request and obtain the security attributes and decision rules that are required for verifying a requested access from other authorization service partitions.

2.2 System Interactions

As Figure 1 depicts, the user interactions begin with a logon procedure on a component that requires the user to enter a userid for identification followed by a password for authentication of that identity. After this action, some processes will be initiated that execute user requests and communicate with the user. The security attributes of these processes will be the same as those of the user on whose behalf they are executing. Other interactions are process-to-process communications or associations, and processes accesses to data. In each case, the security attributes of processes and data, that are maintained in appropriate authorization services, will be used to determine whether the interaction should be allowed.

A service that is required of a reference monitor, whether it is associated with a single component or a distributed system, is a secure association service which controls process-to-process associations. Processes interact with other processes through this secure association service of the reference monitor which protects against unauthorized associations and allows those associations that are authorized, and do not violate the security policy of the component or the distributed system, to succeed. This service mediates and controls interactions on a single component and between components of a distributed system. The initiating process does not distinguish between the two cases and the target process may reside on the same component or on a different one. If an association is to be established between two processes on the same component, a single component's secure association service will be invoked. For a situation in which the target process resides on a separate component, the reference monitors of the two components must cooperate and use secure communications channels to communicate the necessary security information between distributed reference monitors to provide a security association service for the distributed system.

The secure association service has to consult with the authorization service to determine whether or not an association should be allowed. The authorization service is also consulted when processes attempt to access local data. For distributed operations, the security attributes associated with a process or data may not be available at a component to be used by its reference monitor to verify the legitimacy of a requested access. For such cases, the required information can be obtained through communications by requesting the information from another component's authorization service where the information is maintained.

Distributed operations requiring access to some resource on a remote component require an association to be established between a local process (client) that directly supports the user and another process (server) running on the remote component to perform the requested access on behalf of the user (the client process). For this action, the secure association service must be invoked to establish a secure process-to-process association between the two processes. This involves consultation with the initiating component's authorization service to verify the authorization for process-to-process association, and consultation with the remote component's authorization service to verify the authorization for the association and local accesses to the information. After the establishment of the association, the target process that performs the actual access will have to use the user security attributes to consult with the authorization service of the target system or component when requesting access to the target data. Since the authorization information may not be available at the target component, some security exchange protocol or service will be required to communicate user security attributes from the requesting process to the target process. Examples of user security attributes include the user identification information used by the DAC policy, and the user or data classification information used by the MAC policy. Explicit exchanges may be required to transfer a user identification while classification information may be transferred in the sensitivity label field of communication protocols. In addition, because of the communication that causes the data to pass beyond the boundaries of secure and trusted components, it will be required to provide communication-specific services to protect the data during transmission which may pass through hostile and unprotected environments. Examples of this includes protection against disclosure or modification of data that is passing through such environments. Additional services that may be required are those supporting management of encryption keys used to support provision of various security services, management of secure associations, and network management.

The purpose of the above discussion is to present an architectural concept. No attempt has been made to generalize or determine the type, location, or communication of the security attributes. It is possible to visualize several alternatives for implementation of such an architecture whose details are beyond the scope of this paper.

3. Transport Layer Operation

It is appropriate to consider the operation of the transport layer protocol before discussing the services provided by the TLSP. There are two types of transport layer protocols: the connection-oriented transport protocol that can operate in five classes (Classes 0-4) [11]; and the connectionless transport protocol [12]. Each type can use either a connection-oriented or connectionless network service for its operation. A connectionless service is a datagram service. This service transfers data in discrete packets from a source to destination without any guarantee for delivery, any indication to the sender when the packets are delivered to the destination, any notion of order among the packets, or any notion of connection or association between the source and destination entities. The only function supported is the optional guarantee that each packet that arrives at the destination arrives intact - it is neither truncated nor corrupted.

The connectionless transport layer protocol supports two service primitives referred to as the T-Unitdata request (a send operation) and the T-Unitdata indication (a receive operation). Each primitive has four parameters: the source transport address, the destination transport address, the Quality of Service (QoS) parameters, and the user data belonging to the transport service user. The QoS parameters indicate a potential level of service that a transport service user is requesting to receive. This protocol entity receives the T-Unitdata request primitive and builds a Unit Data Transport Protocol Data Unit (UD TPDU) for transmission to its peer using the network layer service primitives that provides connectionless or connection-oriented services. The target transport layer entity (peer) receives the UD TPDU,

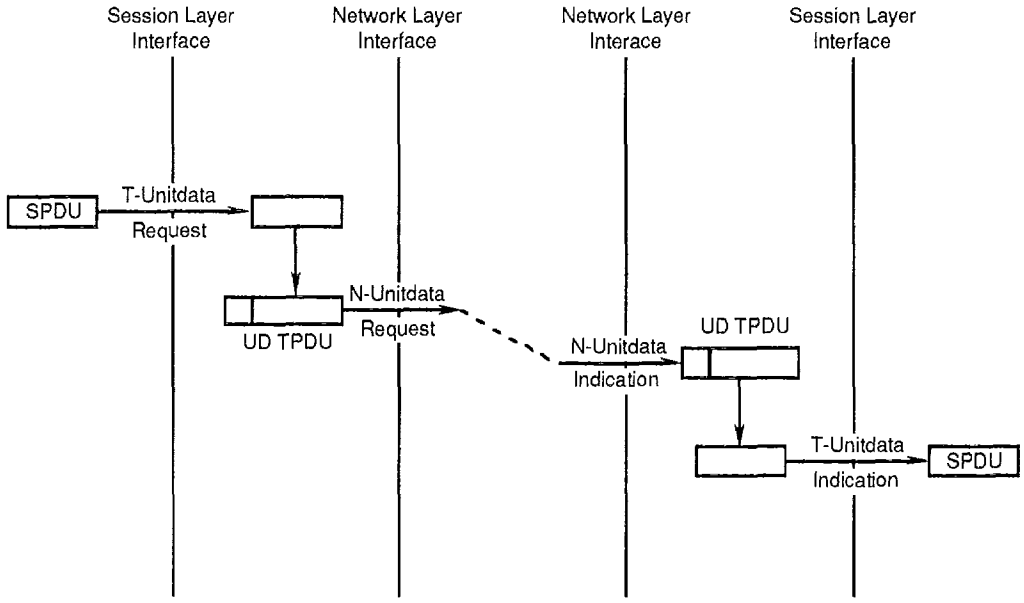

Figure 2. Connectionless Transport Layer Protocol

reconstructs the user data and other parameters and delivers them to the transport layer user at the destination in a T-Unitdata indication. Figure 2 depicts this operation and assumes the network layer service used is connectionless. Transmission of every Session Protocol Data Unit (SPDU), or the Protocol Data Unit (PDU) belonging to any other protocol positioned above the transport layer protocol, results in transmission of UD TPDUs which include the addresses of sender and receiver, the user data (SPDU, or other), and the QoS parameters. Depending on the size of SPDUs, it would be possible to split an SPDU into multiple pieces and send them in multiple UD TPDUs or concatenate multiple SPDUs and send the result in a single UD TPDU.

Connection-oriented service offers many more guarantees than the connectionless service. Connection-oriented transport protocol operates in five different classes (Class 0 through 4) and provides different classes of services that may or may not be used. Class 4 offers the most error detection and recovery guarantees. It guarantees data is delivered at the destination intact, uncorrupted, and in the same order in which it was sent. To maintain an order among the TPDUs, which is supported by all classes, TPDUs must be labeled in some way. To do this, transport protocol establishes a communication path between the source and destination transport users (e. g., session layer entities), called a transport connection (a connection for transmission of data between two particular session entities). There is a phase of connection mode service in which the connection is established, a phase during which all data is transferred, and a phase in which the connection is dissolved. During the data transfer phase, the data is associated with the connection or the path rather than with a pair of addresses, that is used in the connectionless transport service. There may exist any number of transport connections associated with a given pair of transport users.

Figure 3 shows the actions that occur when the session layer attempts to send its SPDU to its peer on behalf of the layers above it. It would have to set up a connection with its peer by placing its request in a T-Connect service primitive for transmission to its peer. For communication of the session data to a session entity at the destination, a transport connection between the two session entities is required. After this transport connection is established, it can be used to transfer SPDUs between session entities.

The transport connection is established by the transport entity by constructing a Connect Request (CR) TPDU and sending it to its peer using the services of the network layer. The target transport entity accepts this request for connection establishment by constructing and returning a Connection Confirm (CC) TPDU. At this step, a connection emerges and is used to transfer the session layer's requests and data to its peer. The

Data (DT) TPDU is used to transfer this higher layer's data and requests between the two higher layer entities. The connection will be released when requested by the higher layer entities.

The bottom part of the transport layer is responsible for transmission of TPDUs using the network layer services. It assigns each transport connection (connection between a pair of session

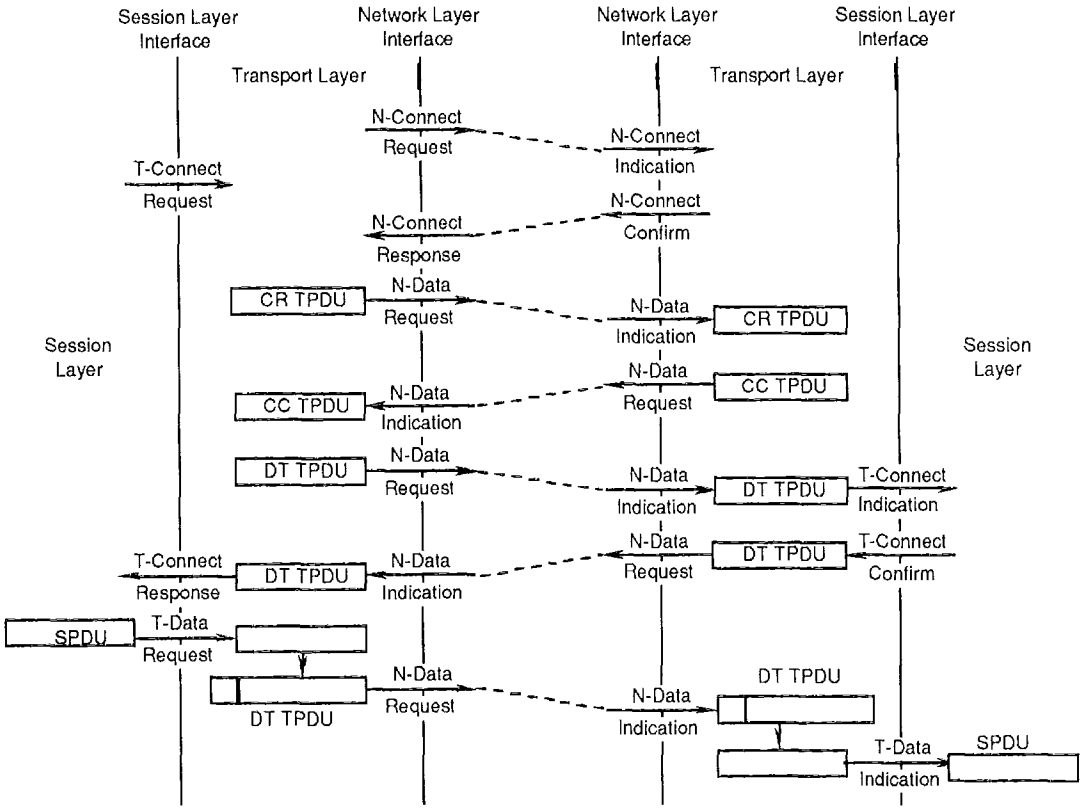

Figure 3. Connection-Oriented Transport Layer Protocol

During the transport connection establishment, a pair of local and remote identifiers are established for the transport connection. These identifiers are source-reference (SRC-REF) and destination-reference (DST-REF). After the connection establishment, they are used to associate transmitted and received TPDUs with the transport connection.

The transport layer can be visualized as having two sublayers or partitions regardless of the type of the protocol used (connection-oriented or connectionless). It is the responsibility of the top portion to construct all TPDUs. Some of these TPDUs carry data or requests on behalf of the higher layer (session, or others) and others are used to carry protocol control information which is required to support normal protocol functions. DT TPDU is an example of the first, while CR TPDU and CC TPDU are examples of the latter.

entities) to one or more network connections (a connection or path between two transport layer entities) if the network service is connection-oriented for transmission and reception TPDUs. It may send two of them together, or send them on multiple network connections. If the network service is connectionless, it will use the datagram service for transmission and reception of TPDUs.

4. Transport Layer Security Protocol

The TLSP is positioned between the higher part of the transport layer protocol that is responsible for constructing the TPDUs for protocol operations and the lower part whose task is communication of TPDUs by using the interface to the network layer and its service primitives. This indicates that the operation of transport layer protocol is independent of TLSP, and if the transport layer protocol is implemented in a modular manner, it

would be possible to add TLSP to the protocol stack with minimal impact on the transport layer protocol.

TLSP is the same for both the connection-oriented and the connectionless transport protocols, and its operation is independent from the type of network service. The services provided by the TLSP, however, depend on the type of the transport service protocol.

The primary function of TLSP is to add a protected and a clear header to each TPDU (See Figure 4). The combination of the protected header and the TPDU form the portion called the protected part that needs to be protected during the transmission. Based on the security services requested, TLSP applies appropriate security transformations to the protected part before transmission. As a result of this action, an encapsulated TPDU is generated that is used to protect the contents of TPDUs. If multiple TPDUs require similar protection and are communicated between the same source and destination TLSP entities, they can be concatenated and encapsulated together.

The protected header contains information such as a security label that contains the security classification and protection authority associated with each TPDU. If confidentiality service is required, the protected part consisting of TPDUs and the protected header will be subjected to encipherment to encrypt the data that is to be protected. If only integrity service is requested, TLSP will generate a checksum called the Integrity Check Value (ICV) and add it to the encapsulated TPDU. The ICV field is cryptographically generated and its value depends on the value of the encryption key and the protected data. Any change to the data during transmission can be detected because the ICV value generated at the destination will be different from the one transmitted in the encapsulated TPDU; therefore, the receiving TLSP entity will be able to detect and reject those encapsulated TPDUs that have been tampered with during the transmission. If both confidentiality and integrity services are requested, an ICV will be added to the protected part before being encrypted. In such cases, when both confidentiality and integrity are to be provided, the ICV may be generated by non-cryptographic approaches because the ICV is protected by the confidentiality service and potential intruders will not be able to modify the data and create a corresponding ICV that will not be detected by the recipient.

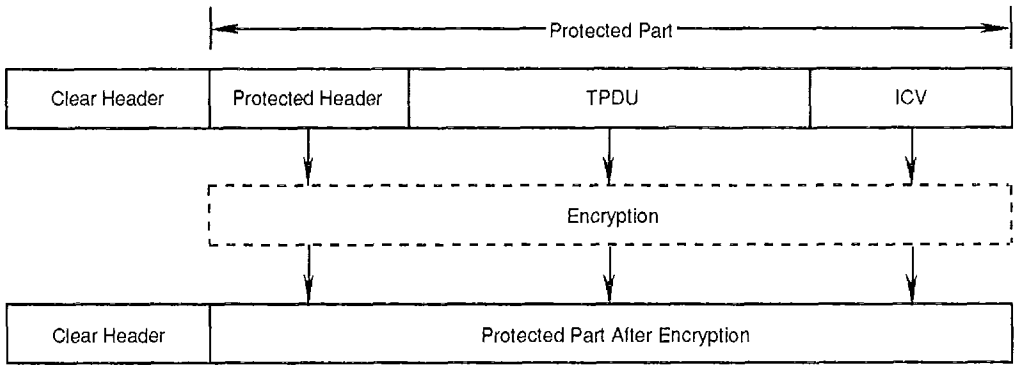

Figure 4. Security Encapsulation of TPDUs

The clear header portion of the encapsulated TPDU is used to allow a TLSP entity to communicate some protocol control information to its TLSP peer in clear text. This includes the protocol identifier that refers to the TLSP, and a security association identifier (SA-ID), which is used to point to the security association between the two TLSP peers to specify the values of parameters used by the association. This includes, among other things, the encipherment keys, algorithms for encipherment, the security services supported by the security association, and the strength of each service.

Before such an encapsulation operation can be initiated, the two TLSP entities should agree on the security association (SA) attributes and the SA attribute values that will be used by the TLSP to perform the encapsulation function. The selection of an SA is based on the services requested by the TLSP service user (higher layer). If a proper SA does not exist, then one needs to be created. TLSP PDUs are generated either to communicate encapsulated TPDUs or for setting up and updat-

ing of SA attributes. More information on this issue is presented in the Security Management Function section.

5. Security Management Functions

The protection achieved by this security protocol depends on the proper operation of security management functions, which include key management and establishment of SAs used by the TLSP.

The security management functions are not part of TLSP and will be defined by other standard protocols. Different techniques can be used for establishment of SAs. One approach would be to establish SAs and set the value of their parameters manually. For automated establishment of SAs, one may consider two alternatives. In one, the TLSP will be augmented with other procedures to establish SAs. For this, new PDUs, other than security encapsulation, will be required to exchange security parameters and to negotiate SA attributes and set their values.

In the second alternative, the security management function will be independent from user communication services. When TLSP requires a security association with the characteristics that do not exist, it makes a request to some local security management process to establish an SA for its use with the specified characteristics. This causes the security management process to establish an association with another remote security management process that serves the remote TLSP peer. After the authentication and access control requests are validated, a cryptographic association that specifies a pair-wise keys will be established. The parameters set by the security management process should include the cryptographic keys used for confidentiality and integrity services, access control privileges, and the security protocol identification. Other parameters required by the SA may be set by the security management process interactions or left to be set by TLSP through the exchange of other PDUs as described in the previous alternative. After the completion of the required SA establishment, TLSP will use it to provide the services that were requested by its user through its interface QoS parameters.

6. Resulting Security Services

In the OSI environment, the security exchanges may be specified in a common way following the ISO guidelines [13] which partitions various functions into Application-Service-Elements (ASE). The security services may be provided by the Security Exchange ASE (SE ASE). This ASE will be responsible to support the security services within the application layer of OSI. Each application or process requiring security services will invoke the SE ASE to communicate the appropriate security information required by the application-to-application (or process-to-process) association.

A process that is engaging in a process-to-process association can request to use a number of security services for the association through the use of security parameters of SE-ASE. Each of these services may either be directly provided by the encipherment mechanisms of the presentation layer or indirectly provided by requesting the use of appropriate mechanisms in the lower layers. The services at a lower layer such as the TLSP can be invoked through the QoS parameters that are provided at its interface for use by upper layers.

Not all the security services required for OSI communications can be provided by the TLSP. Examples of services that are not addressed by TLSP include non-repudiation and selective field confidentiality [1]. If security services beyond those provided by the TLSP are required, the security architecture must augment TLSP with other layer security services to fulfill the security needs of the architecture.

The following sections describe the security services that can be supported by invoking the TLSP. These security services are different for connection-oriented or connectionless transport services.

6.1 Confidentiality

Confidentiality service provides protection against disclosure of user data. This service can be provided either by limiting an intruder's access to data by using some access control mechanism or by use of encipherment. TLSP is an example of protocols that use encipherment to provide confidentiality service for data in transit. To accomplish this, one TLSP entity enciphers data before transmission and its peer deciphers it after reception. The services provided by TLSP depend

on the type of transport layer protocol that is being considered and could be those associated with either connectionless or connection-oriented protocols. The service provided will be connectionless confidentiality if the transport layer protocol is connectionless; otherwise, it will provide connection confidentiality. Connectionless confidentiality refers to the protection of all UD TPDUs going from a transport entity to another transport entity. Connection confidentiality refers to the confidentiality of all TPDUs exchanged over a transport connection during all of its operational phases. TLSP provides these services by enciphering all TPDUs.

As discussed earlier in the paper, this service requires the existence of pre-established SAs between pairs of TLSP entities before an instance of protected communication may begin.

6.2 Integrity Service

Integrity service protects against manipulation of data by an intruder. Communication protocols guard against accidental change of data due to transmission errors. However, they are not capable of guarding against intentional modifications.

Like confidentiality service, integrity service can be provided by either mechanisms controlling access to data or by mechanisms that make any change to data detectable. It would also be possible to introduce mechanisms to help recover the data portion that has been corrupted.

The integrity service provided by the TLSP is a function of the transport layer protocol. In the connectionless environment the service is connectionless integrity, while in the connection-oriented case it is connection integrity.

The mechanism used for the integrity service is the ICV field described earlier. Any modification to an encapsulated TPDU, as it crosses beyond the boundaries of trusted components into unprotected and potentially hostile environment, can be detected by recalculating the ICV and comparing it against the ICV field of the encapsulated TPDU. If the two ICVs do not mach, the encapsulated TPDU will be discarded by the receiving TLSP entity. The ICV mechanism provides limited integrity protection, if any, against replaying of an encapsulated TPDU.

Connectionless integrity relies on the ICV mechanism only and ensures that no UD TPDU has been modified during transmission. It does not convey any information about UD TPDUs that may have been lost, duplicated, or replayed. Furthermore, it does not have the capability to recover any of the lost UD TPDUs. If required, such recovery has to be supported by higher-layer protocols.

Using the connection-oriented transport layer protocol, TLSP is capable of providing connection integrity with or without recovery depending on the class of the transport layer protocol. In this case, TLSP uses the ICV plus the source and destination reference-numbers for its protocol operation. This allows it not only to detect changes to encapsulated TPDUs, but also to detect loss or duplication of encapsulated TPDUs.

TLSP by itself can not provide recovery service; however, when it is associated with a transport layer protocol that supports a recovery function, the services provided will be extended to include the recovery function. The only transport layer protocol providing recovery function is the Transport Protocol Class 4 (TP4). When TP4 and TLSP are integrated, the retransmission capability of TP4 can be used to recover lost TPDUs.

6.3 Authentication Service

Like the previous two services, the authentication service provided by the TLSP depends on the type of transport layer protocol. In the connectionless environment the service is the data-origin authentication, while the service associated with the connection-oriented protocol is the peer-entity authentication.

Data-origin authentication provides a proof of the origin of UD TPDUs. Peer-entity authentication, on the other hand, provides the assurance that the peer-entities communicating with each other are mutually authenticated and each is assured that the other is the one it is claiming to be. Peer-entity authentication ensures authentication of peer-entities at connection time and at all the other times during the connection.

Data-origin authentication requires exchange of some information that identifies the sending entity. The peer-address is used for this verification and it should be protected during transmission. Therefore, either confidentiality or integrity service is required to provide this protection and to ensure that the peer-addresses have not been tampered with during the communication.

Peer-entity authentication consists of mutual authentication of transport users during all phases of an association including connection establishment, data transfer, and finally connection termination. TLSP uses source and destination addresses during connection establishment and source and destination reference-numbers that point to the connection at all other times to associate the encapsulated TPDU with a connection and identify the source and destination addresses of each TPDU.

This service relies on the integrity service to protect the source and destination addresses at the connect establishment time and the source and destination reference-numbers at all times during the life of a connection.

6.4 Access Control Service

The goal of access control is to provide protection against unauthorized use of resources. Access control can be enforced between systems or within a system. It is the responsibility of a system's reference monitor to control access to local resources. For access to remote resources, an association with a remote process must be established to perform the requested access and communicate the results. OSI security can control accesses only when a remote access is required. The access control service of the TLSP can control only pair-wise associations between remote processes. It can not control, however, the activities of the local or remote processes.

For example, when a user invokes the File Transfer, Access and Management (FTAM) application to access some remote data, a number of activities will occur. First, the user invokes a copy of (an instance of) the FTAM client which attempts to establish an association with the FTAM server at the destination. The access control service provided by the TLSP ensures such an association is not violating the network's security policy. TLSP can not control the actions performed by the FTAM client or server when they access resources that are local to them. Resource access control will be the purview of the particular FTAM application that provides the protocol for exchanging manipulation requests and responses for a particular resource. As part of the manipulation requests, the client or the requesting process sends the appropriate user security attributes including access control information (e.g., the user identity and classification) to the remote process or the server. These security attributes will be used by the server at the remote host to access the resource on behalf of the requesting client. The reference monitor of the remote host mediates the access to the resource attempted by the server and uses the communicated security attributes to determine whether or not the access should be allowed.

TLSP controls associations between processes by using different approaches. It may use the confidentiality or integrity service and the transport addresses to control such associations. That is, if the data is not deciphered correctly as indicated by the transport addresses or transport sequence numbers, which are known values, no communication will be allowed. If integrity is used to support access control, the ICV is used to verify address and reference-number parameters. Based on the value of the addresses and connection reference-numbers, TLSP will determine whether or not a process-to-process communication should be allowed.

Another mechanism available to TLSP is the security label associated with each encapsulated TPDU that can be used to enhance TLSP's access control capability. Security labels can provide additional information for controlling process-to-process communications. The information in the security label consists of a defining authority identifier and the information that is defined by the defining authority. Security labels allow control of communication between two processes to be enhanced by limiting the access or TPDU exchanges to some pre-defined set of labels.

Security labels serve a very useful purpose in multilevel secure (MLS) environments. The trusted portions (referred to as the trusted computer base or TCB in reference 9) of the distributed system which includes TLSP entities would require to differentiate among different classifications. If no label mechanism is use, it would be required to use a separate key for each access level (access level consists of a security classification or level and some categories). If labels are employed, they can be used to communicate such information and one pair-wise encryption key(s) would be sufficient for supporting communications at all access levels between each pair of end-system components.

7. Summary And Conclusions

The TLSP can be used to implement a secure distributed system providing several security services. The paper presents a model of a distributed system and shows how TLSP can be applied to this model to develop a security architecture that not only protects communicated data and messages, but also creates a secure distributed system controlled by a network-wide reference monitor that guards against unauthorized accesses to any resource irrespective of its location in the distributed system. The discussions presented include a summary of the OSI transport layer protocols, details of the TLSP operation, and the security service than can realized as a result of using TLSP. In addition, the paper discusses the security functions expected from the end-systems, and the security management functions that need to be addressed.

References

[1] ISO/IEC 7498-2, February 1989, Information Processing System - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture.

[2] ISO/IEC 10736, December 1992, Transport Layer Security Protocol.

[3] ISO TC97/SC21/N6765, November 1991, Guide to Open Systems Security.

[4] ISO/IEC DIS 10181-2, May 1991, Information Technology - Open Systems Interconnection - Security Frameworks in Open Systems - Part 2: Authentication Framework.

[5] ISO/IEC CD 10181-3, June 1991, Information Technology - Open Systems Interconnection - Security Frameworks in Open Systems - Part 3: Access Control.

[6] ISO/IEC CD 10181-5, 1992, Information Technology - Open Systems Interconnection - Security Frameworks in Open Systems - Part 5: Confidentiality.

[7] ISO/IEC CD 10181-6, June 1991, Information Technology - Open Systems Interconnection - Security Frameworks in Open Systems - Part 6: Integrity.

[8] ISO/IEC DIS 10745, 1992, Information Technology - Open Systems Interconnection - Upper Layers Security Model.

[9] DOD 5200.28-STD, December 1985, Department of Defense Trusted Computer System Evaluation Criteria.

[10] ECMA TR/42, July 1987, Framework for Distributed Office Application.

[11] ISO/IEC 8073, 1988, Information Processing System - Open Systems Interconnection -Connection Oriented Transport Protocol Specification.

[12] ISO 8602, 1987, Information Processing System - Open Systems Interconnection - Protocol for Providing the Connectionless-Mode Transport Service.

[13] CCITT X.219, November 1988, Remote Operations: Model, Notation and Service Definition.